



SAPIENZA
UNIVERSITÀ DI ROMA

Sapienza University of Rome

Department of Computer, Control and Management Engineering
PhD in Data Science

THESIS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

Architectural Components of Trustworthy Artificial Intelligence

Thesis Advisor
Prof. Fabrizio Silvestri

Candidate
Federico Siciliano
1604124

Academic Year 2022-2023 (XXXVI cycle)

Laugh Hard
Run Fast
Be Kind

Summary

Trustworthy Artificial Intelligence (AI) is a cornerstone of the digital era, encompassing the need for AI systems to be not only powerful but also transparent, resilient, and accountable. This thesis, titled *Architectural Components of Trustworthy Artificial Intelligence*, aims to explore the essential elements that underpin the development of AI systems that are inherently trustworthy. This work unfolds the foundations, methodologies, and innovations crucial for fostering trust in AI systems. The following summary provides an overview of the key contributions and insights of this thesis.

The introduction provides a backdrop to the research, elucidating the motivations and objectives driving the study. It outlines the structure of the thesis, setting the stage for a systematic exploration.

Starting our exploration, we delve into the fundamentals of *Explainability-by-design*. We introduce innovative concepts, including a novel generalization of artificial neurons, that redefine the foundations of model transparency. Furthermore, we investigate concept-based explainability, shedding light on how these networks provide insight into the decision-making processes of AI models.

Turning our attention to the critical aspect of training trustworthy AI, we explore the development of loss functions tailored to address the challenges posed by noisy labels and missing data, particularly in recommender systems. We also show how integrating item relevance into the loss functions makes the model more resilient and dependable in the face of adversities.

We then broaden our investigation introducing the concept of *Trustworthy Auxiliary Frameworks*: it extends beyond model-centric trustworthiness by incorporating elements such as counterfactual personalized recourse, active learning for misinformation detection, and retrieval augmentation. These auxiliary components address crucial aspects like data governance, monitoring, and interpretability, strengthening the AI system's trustworthiness throughout its lifecycle.

The final part of this thesis summarizes key findings and contributions to the field of Trustworthy AI. It shows how we achieved the objectives outlined in the introduction, advancing the understanding and practical implementation of architectural components that enhance trustworthiness in AI systems across diverse domains. It also offers insights into future research directions, emphasizing the need for ongoing innovation and development in this critical domain.

In conclusion, this thesis represents a significant step in the ongoing pursuit of Trustworthy AI. It stands as a valuable resource for researchers and practitioners striving to create AI systems that inspire trust and confidence. With the principles of trust, accountability, and transparency at its core, this research contributes to the collective effort of ensuring that AI serves humanity with the highest standards of ethics and responsibility.

Keywords: Explainability, Artificial Intelligence, Trustworthy, Neural Networks

Contents

1	Introduction	1
2	Explainable-by-Design Neural Networks	8
2.1	NEWRON: a New Generalization of the Artificial Neuron to Enhance the Interpretability of Neural Networks	9
2.2	Explaining Neural Networks Using a Ruleset Based on Interpretable Concepts	23
2.3	Concept Distillation in Graph Neural Networks	34
2.4	Explainable-by-design Machine Learning Model for Overlapping Fluorophores Separation Based on Fluorescence Lifetime	52
3	Robust Losses for AI Systems	58
3.1	Robust Training of Sequential Recommender Systems with Missing Input Data	59
3.2	Integrating Item Relevance in Training Loss for Sequential Recommender Systems	72
3.3	Leveraging Inter-rater Agreement for Classification in the Presence of Noisy Labels	80
4	Auxiliary Frameworks for Trustworthy AI Systems	94
4.1	Human-in-the-loop Personalized Counterfactual Recourse	95
4.2	Deep Active Learning for Misinformation Detection Using Geometric Deep Learning	107
4.3	RRAML: Reinforced Retrieval Augmented Machine Learning	124
5	Conclusions	130
	Bibliography	133
A	NEWRON- Supplementary Materials	160
B	Explaining Neural Networks Using a Ruleset Based on Interpretable Concepts	176
C	Noisy Labels - Supplementary Materials	179
D	Personalized Recourse - Supplementary Materials	195

Chapter 1

Introduction

In recent years, artificial intelligence (AI) has witnessed remarkable advancements [102, 253, 318], transforming various aspects of our lives, from recommendation systems powering e-commerce platforms [133] to the deployment of AI in critical decision-making processes [203].

With AI demonstrating human-level performance in numerous tasks, including image recognition [120], natural language processing [17, 163, 227], and autonomous decision-making [110, 307], it has found its way into critical applications, where reliability and safety are paramount. However, this rapid integration of AI has raised significant concerns about interpretability, robustness and trustworthiness, as its deployment into critical applications necessitates not only high performance but also transparency and reliability [241]. Trustworthiness and accountability in AI systems are no longer optional but imperative [46]. The consequences of AI errors, biases, or misinterpretations can have far-reaching implications, impacting individuals and society as a whole. In fact, it is important to note that Trustworthy AI is not merely an aspiration but also a legal requirement in compliance with regulations such as the General Data Protection Regulation (GDPR) [233] and the AI Act [298]. These legal frameworks underscore the urgency of addressing trustworthiness in AI and ensure that AI development aligns with evolving legal standards and societal expectations [213, 276].

Therefore, it becomes essential to address the *black-box* nature of deep learning models, enhancing their explainability, and fortifying their robustness in the face of evolving challenges, making them more reliable and resilient tools for a wide range of applications. Yet, with the increasing complexity of AI models, understanding their inner workings and making informed decisions based on their outputs become challenging [228, 242].

To address these challenges, the concept of *eXplainable AI* (XAI) [9] has emerged as a pivotal research area. XAI aims to create AI systems that can provide clear and interpretable explanations for their decisions and predictions. This transparency not only enhances user trust [28, 76, 261] but also enables users to understand [255, 340] and potentially rectify erroneous outcomes [143].

Furthermore, achieving trustworthiness in AI extends beyond explainability. It encompasses robustness against adversarial attacks [5, 16, 39, 130, 277, 332], fairness in decision-making [185], and resilience in the face of uncertain or noisy data [112, 198, 214, 317].

Motivated by these challenges, this thesis delves into the architectural components of trustworthy AI. We recognize that the path to trustworthy AI fundamentally lies in the design of AI systems. By imbuing these systems with inherently trustworthy features and capabilities, we can mitigate risks and promote their responsible deployment.

In the following chapters, we will explore various facets of trustworthy AI, from explainable-by-design neural networks to robust loss functions and advanced auxiliary frameworks. Through a systematic investigation of these architectural components, this research aims to provide practical insights and solutions for the development of AI systems that are not only high-performing but also dependable.

AI Legal Framework

The European Union (EU) has responded to the rapid advancement of artificial intelligence (AI) with a set of regulations aimed at ensuring the ethical development of AI, and strengthening data protection. These initiatives include the central Artificial Intelligence Act (AI Act) [298], the Artificial Intelligence Liability Directive (AILD) [299] and the current General Data Protection Regulation (GDPR) [233].

Central to the AI Act is a concerted effort to prevent harm through a risk-based approach. This classification system stratifies AI systems according to the level of risk they pose to the security and fundamental rights of individuals. Recognising the surge in the development and use of generative AI, the European Commission has introduced strict transparency obligations, including explicit disclosure when content is generated by AI, prevention of the generation of illegal content, and disclosure of copyrighted material in the datasets used. However, while the AI Act mandates certain aspects of transparency, it lacks specificity in certain areas. In particular, it requires providers to supply users with instructions for AI systems and inform human decision-makers in order to facilitate informed decisions and mitigate the potential bias introduced by automation.

While the AI Act will reduce the risks to security and fundamental rights, it will not completely eliminate the risk of potential direct or indirect damage caused by AI systems. The AILD, in turn, outlines rules for compensation of damage caused intentionally or negligently by AI systems within the EU market.

The distinction between the AI Act and the AILD underlines their different functions. The AI Act primarily emphasises safety measures to prevent AI-induced harm, while the AILD provides avenues for seeking compensation following AI-related damage. When comparing the scopes of the AI Act and GDPR, a noticeable difference emerges. The AI Act applies to providers, users, and other entities that operate AI systems within the EU market. In contrast, the GDPR, which regulates the processing of personal data within or relating to EU data subjects. As a result, while an AI system may fall within the scope of the AI Act, it may not meet the specific criteria delineated by the GDPR.

Once approved, both the AI Act and the AILD are set to become the world's first regulations specifically focused on AI. The AI Act is expected to come into force in late 2023 or 2024, accompanied by a two-year grace period to allow organisations to comply. Conversely, the exact timeline for the adoption of the AILD remains uncertain, but member states will be required to incorporate it into their legal frameworks within two years of its adoption.

Research Objectives

The research objectives that guide this thesis are formulated to address the fundamental challenges in enhancing the trustworthiness and explainability of artificial intelligence systems.

Objective 1: Develop Explainable AI Methods and Components

The primary objective of this research is to investigate and propose architectural components as well as methodological approaches that enhance the explainability of AI systems. We aim to develop novel techniques and models that make AI decisions interpretable, transparent, and accessible to both technical and non-technical stakeholders.

Objective 2: Establish Trustworthiness Through Robust Loss Functions

Trustworthiness is a multifaceted concept, encompassing robustness, fairness, and resilience in AI systems. Our second objective is to design and evaluate robust loss functions that mitigate vulnerabilities to adversarial attacks and improve the fairness of AI decision-making.

Objective 3: Architect Trustworthy AI Auxiliary Frameworks

To build AI systems that are inherently trustworthy, we aim to design auxiliary architectural frameworks that bolsters the reliability and accountability of AI systems. These auxiliary frameworks will facilitate decision corrections, misinformation detection, and reinforcement learning augmentation with memory, contributing to the overall trustworthiness of AI systems and thereby instilling confidence in the technology’s deployment across various applications and domains.

Objective 4: Contribute to Trustworthy AI Research

Beyond the immediate objectives of this research, we seek to contribute valuable insights and solutions to the broader field of trustworthy AI. By conducting empirical studies, evaluations, and experiments, we aim to provide practical guidelines and best practices for researchers and practitioners working in AI, ensuring that the principles of trustworthiness are integrated into future AI technologies.

These research objectives serve as the compass that guides our exploration of architectural components for trustworthy AI. By pursuing these objectives, we aspire to contribute to the advancement of AI technologies that are not only powerful but also accountable, transparent, and ethically grounded [174].

Thesis Outline

The organization of this thesis is meticulously designed to address the research objectives and contribute to the understanding and development of architectural components for trustworthy AI.

Chapter 2: Explainable-by-Design Neural Networks

In this chapter, we introduce the concept of Explainable-by-Design and explore how it aligns with the broader objective of trustworthy AI. This chapter introduces NEWRON, a novel generalization of artificial neurons, and investigates how to enhance interpretability through concept-based explanations.

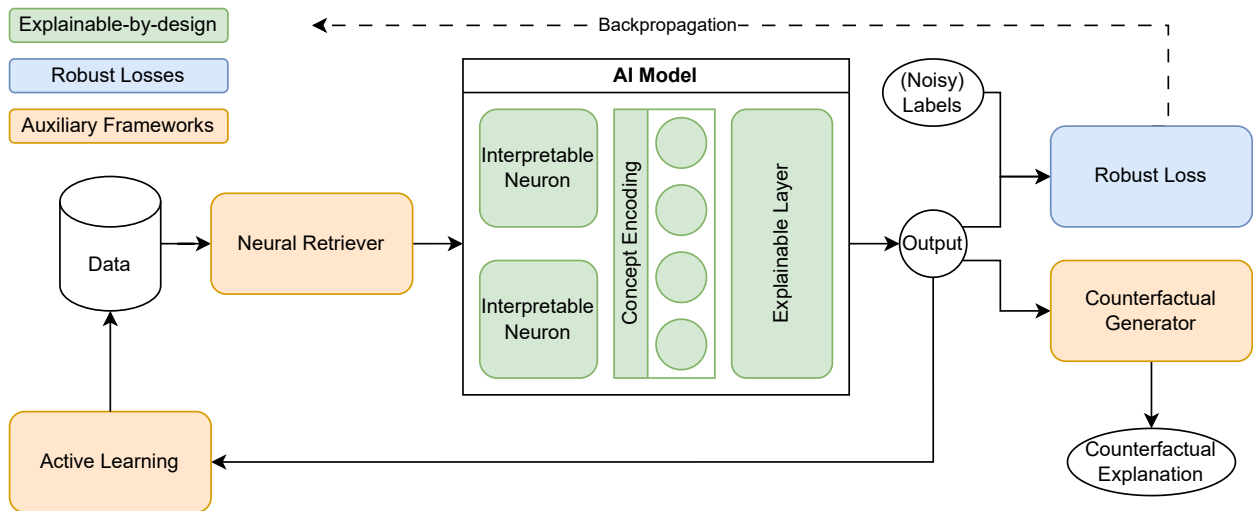


Figure 1.1: Abstract Depiction of Architectural Components in Trustworthy Artificial Intelligence

Chapter 3: Robust Losses for AI Systems

Chapter 3 focuses on the development of robust loss functions for robust AI systems. It starts with the design and evaluation of robust recommender loss functions. The chapter further explores the integration of item relevance in training loss for sequential recommender systems, and addresses challenges related to noisy labels in machine learning, leveraging inter-rater agreement to enhance classification accuracy.

Chapter 4: Auxiliary Frameworks for Trustworthy AI Systems

Chapter 4 introduces the concept of Auxiliary Frameworks and their significance in building inherently trustworthy AI systems. We explore counterfactual personalized recourse for AI systems and active learning mechanisms for misinformation detection. Additionally, we introduce Reinforced Retrieval Augmented Machine Learning (RRAML) as an architectural auxiliary framework for trustworthiness.

Chapter 5: Conclusions

In the final chapter, we summarize the key findings and contributions of this research. We reflect on the architectural components developed and their implications for trustworthy AI. Additionally, we outline future research directions in the field of trustworthy AI, highlighting areas where further exploration is warranted. The chapter concludes with closing remarks and a reaffirmation of the ethical and responsible AI development principles advocated throughout this thesis.

Figure 1.1 illustrates an abstract representation of how various architectural components can be integrated into a neural network model. Although these components may not be applied simultaneously in practice, the figure conveys the idea that these individually developed elements can be flexibly combined to continually enhance the trustworthiness of a model. This concept underscores the significance of composing these architectural components to progressively improve the model's trustworthiness.

Research Focus

The central theme of this doctoral research revolves around three primary domains: Explainable-by-design AI, Robust Losses, and Auxiliary Frameworks for Trustworthy AI. Throughout Federico Siciliano’s doctoral journey, significant contributions have been made to these areas, resulting in novel insights and approaches. This section provides an overview of Federico Siciliano’s key research endeavors and contributions, together with the publications that compose each chapter.

Explainable-by-design AI

A substantial portion of FS’ efforts has been dedicated to the concept of Explainable-by-Design AI.

- F. Siciliano, M. S. Bucarelli, G. Tolomei, and F. Silvestri. Newron: a new generalization of the artificial neuron to enhance the interpretability of neural networks. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 01–17. IEEE, 2022

In particular, he conceptualized and developed NEWRON [269], a generalized version of the McCulloch-Pitts neuron. NEWRON enables the definition of new artificial neurons, and through the universal approximation theorem, it was demonstrated that this new model does not compromise its representation power. Additionally, the Inverted Artificial Neuron was introduced, offering a straightforward translation into logical rules, paving the way for interpretable, white-box neural networks.

- F. Siciliano, L. C. Magister, M. S. Bucarelli, P. Barbiero, F. Silvestri, and P. Lio. Explaining neural networks using a ruleset based on interpretable concepts. In *Submitted to EPJ Data Science*, 2023

Expanding on the NEWRON framework, a collaboration with researchers from the University of Cambridge resulted in the development of NEWRON+LEN [273]. This integrated approach combines NEWRON with Logic Explained Networks (LEN), allowing the distillation of interpretable concepts in the form of rules during training for classification. These concepts are harnessed by LEN to produce global explanations in the form of First Order Logic rulesets, thereby linking concepts to classes.

- L. C. Magister, P. Barbiero, D. Kazhdan, F. Siciliano, G. Ciravegna, F. Silvestri, M. Jamnik, and P. Liò. Concept distillation in graph neural networks. In *World Conference on Explainable Artificial Intelligence*, pages 233–255. Springer, 2023

Continuing the work in Concept-based Explainability, FS collaborated with Cambridge researchers on [183]. In this work, a Concept Distillation Module was introduced, a pioneering differentiable concept-distillation approach for graph networks. This module can be seamlessly integrated into any graph network to make it explainable by design, distilling graph concepts from the latent space to solve the task.

- L. Cuneo, F. Siciliano, M. Castello, S. Piazza, F. Silvestri, and A. Diaspro. Explainable-by-design machine learning model for overlapping fluorophores separation based on fluorescence lifetime. In *Computational Intelligence Methods for Bioinformatics and Biostatistics: 18th International Meeting, CIBB 2023, Padova, Italy, September 6–8, 2023*, 2023

In the design of an Explainable-by-Design neural network to separate contributions from different fluorophores in fluorescence microscopy imaging [61], FS' innovative method leverages a CNN-based network to analyze both temporal information and 2D spatial features, marking a groundbreaking development in this field.

Robust Losses for AI

In parallel to the primary focus, FS studied the development and applications of Robust Losses for AI.

- F. Siciliano, S. Lagziel, and G. Gamzu, Iftah Tolomei. Robust training of sequential recommender systems with missing input data. In *Submitted to Information Processing and Management*, 2023

During his internship at Amazon, FS devised a robust loss function [272] to handle missing elements in Sequential Recommender Systems.

- F. Siciliano, A. Bacciu, N. Tonellotto, and F. Silvestri. Integrating item relevance in training loss for sequential recommender systems. In *Proceedings of the 17th ACM Conference on Recommender Systems*, RecSys '23, page 1114–1119, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400702419. doi: 10.1145/3604915.3610643. URL <https://doi.org/10.1145/3604915.3610643>

Continuing and expanding this work, FS concieved the idea of integrating item Relevance into recommender systems [271], both during training, making them more robust and performant, and in evaluation, allowing a more refined assessment.

- M. S. Bucarelli, L. Cassano, F. Siciliano, A. Mantrach, and F. Silvestri. Leveraging inter-rater agreement for classification in the presence of noisy labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3439–3448, 2023

In [38], FS took a leading role in designing and implementing the experimental setup, developing code for running and evaluating experiments, and significantly contributing to the paper's writing.

Trustworthy AI Auxiliary Frameworks

In a secondary capacity, FS has also explored the concept of Trustworthy AI Auxiliary Frameworks.

- F. Siciliano, C. Abrate, F. Bonchi, and F. Silvestri. Human-in-the-loop personalized counterfactual recourse. In *Submitted to International Conference on Artificial Intelligence and Statistics (AISTATS) 2024*, 2023

Within the realm of Explainability, FS formalized and introduced the concept of Personalized Counterfactual Recourse [270], an innovative approach to Counterfactual Explainability that accounts for user preferences.

- G. Barnabò, F. Siciliano, C. Castillo, S. Leonardi, P. Nakov, G. Da San Martino, and F. Silvestri. Deep active learning for misinformation detection using geometric deep learning. *Online Social Networks and Media*, 33:100244, 2023

FS collaborated on the implementation and writing of [23], which introduces an active learning method that empowers a network to select samples for annotation and training.

- A. Bacciu, F. Cuconasu, F. Siciliano, F. Silvestri, N. Tonello, and G. Trappolini. Rraml: Reinforced retrieval augmented machine learning. In *AIxIA 2023—Advances in Artificial Intelligence: XXIIInd International Conference of the Italian Association for Artificial Intelligence, AIxIA 2023, Rome, Italy, November 6 – 9, 2023, Discussion Track*, 2023

Lastly, FS contributed to the conception of Reinforced Retrieval Augmented Machine Learning (RRAML) [14], a visionary framework that integrates the reasoning capabilities of Large Language Models (LLMs) with information retrieved from a user-provided database.

The writing of this thesis also benefited from work on the following publications:

- F. Betello, F. Siciliano, P. Mishra, and F. Silvestri. Investigating the robustness of sequential recommender systems against training data perturbations: an empirical study. *46th European Conference on Information Retrieval (ECIR) 2024*, 2024
- G. Barnabò, F. Siciliano, C. Castillo, S. Leonardi, P. Nakov, G. Da San Martino, and F. Silvestri. Fbmultilingmisinfo: Challenging large-scale multilingual benchmark for misinformation detection. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2022
- F. Siciliano, L. Maiano, L. Papa, F. Baccini, I. Amerini, and F. Silvestri. Adversarial data poisoning for fake news detection: How to make a model misclassify a target news without modifying it. In *ECML-PKDD Deep Learning and Multimedia Forensics Workshop*, 2023
- G. Grani, M. Gentili, F. Siciliano, D. Albano, V. Zilioli, S. Morelli, E. Puxeddu, M. C. Zatelli, I. Gagliardi, A. Piovesan, et al. A data-driven approach to refine predictions of differentiated thyroid cancer outcomes: a prospective multicenter study. *The Journal of Clinical Endocrinology & Metabolism*, page dgad075, 2023
- F. Greco, A. Polli, F. Siciliano, et al. Leveraging deep learning models to assess the temporal validity of emotional text mining procedures. In *JADT 2022 Proceedings: 16th International Conference on Statistical Analysis of Textual Data*, volume 2, pages 475–481, 2022
- F. Siciliano, G. Consolini, R. Tozzi, M. Gentili, F. Giannattasio, and P. De Michelis. Forecasting sym-h index: A comparison between long short-term memory and convolutional neural networks. *Space Weather*, 19(2):e2020SW002589, 2021

In conclusion, Federico Siciliano’s doctoral journey has been characterized by a diverse array of research contributions spanning Explainable-by-design Neural Networks, Robust Losses and AI Auxiliary Frameworks, and more. These endeavors collectively exemplify a dedication to advancing the frontiers of Explainable AI research and its practical applications.

Chapter 2

Explainable-by-Design Neural Networks

Traditional black-box neural networks often hinder our ability to trust AI systems in critical applications. *Explainable-by-design* neural networks aim to rectify this limitation by embedding interpretability into the very fabric of the model. By designing networks with transparency as a core principle, we can achieve a more profound understanding of model behaviour, decision boundaries, and feature importance. This proactive approach to explainability fosters trust in AI systems and paves the way for their adoption in domains where accountability and trustworthiness are paramount.

At the heart of *Explainable-by-design* is the recognition that interpretability should not be an afterthought but a guiding principle in the design and development of neural networks. It involves the deliberate consideration of how the model will provide explanations for its decisions during the architecture and training phases. Unlike post hoc explainability methods that attempt to interpret a model's predictions after training, *Explainable-by-design* takes a proactive approach. It integrates interpretability from the outset, making it an integral part of the neural network's architecture, objectives, and evaluation metrics. Explainable-by-design neural networks encompass several crucial concepts and techniques that set them apart from conventional models. These may include architectural choices, regularization methods, and training strategies specifically tailored to promote interpretability.

Throughout this chapter, we will explore these concepts in detail, offering a comprehensive understanding of how they contribute to the creation of AI systems that are not only accurate but also explainable by design.

2.1 NEWRON: a New Generalization of the Artificial Neuron to Enhance the Interpretability of Neural Networks

In this work, we formulate NEWRON: a generalization of the McCulloch-Pitts neuron structure. This new framework aims to explore additional desirable properties of artificial neurons. We show that some specializations of NEWRON allow the network to be interpretable without affecting their expressiveness. We can understand the rules governing the task by just inspecting the models produced by our NEWRON-based networks. Extensive experiments show that the quality of the generated models is better than traditional interpretable models and in line or better than standard neural networks.

2.1.1 Introduction

Neural Networks (NNs) have now become the *de facto* standard in most Artificial Intelligence (AI) applications. The world of Machine Learning has moved towards Deep Learning, i.e., a class of NN models that exploit the use of multiple layers in the network to obtain the highest performance.

Research in this field has focused on methods to increase the performance of NNs, in particular on which activation functions [7] or optimization method [284] would be best. Higher performances come at a price: [9] show that there is a trade-off between interpretability and accuracy of models. Explainable Artificial Intelligence (XAI) is a rapidly growing research area producing methods to interpret the output of AI models in order to improve their robustness and safety (see e.g. [97] and [28]). Deep Neural Networks (DNNs) offer the highest performance at the price of the lowest possible interpretability. It is an open challenge to attain such high performance without giving up on model interpretability.

The simplest solution would be to use a less complex model that is natively interpretable, e.g., decision trees or linear models, but those models are usually less effective than NNs. We ask the following question: can we design a novel neural network structure that makes the whole model interpretable without sacrificing effectiveness?

NNs are black-box models: we can only observe their input and output values with no clear understanding of how those two values are correlated according to the model's parameters. Although a single neuron in the NN performs a relatively simple linear combination of the inputs, there is no clear and straightforward link between the parameters estimated during the training and the functioning of the network, mainly because of the stacking of multiple layers and non-linearities.

In this work, we propose a generalization of the standard neuron used in neural networks that can also represent new configurations of the artificial neuron. Thus, we discuss a specific example that allows us to interpret the functioning of the network itself. Like the standard neuron, ours can also be used to stack multiple layers in sequence, i.e. to generate DNNs.

We focus our efforts on tabular data since we investigate how NEWRON works only in the case of fully connected NNs. It is more straightforward to produce human-readable rules for this kind of data. We also remark that our goal is not to improve the performance of NNs, but rather to create interpretable versions of NNs that perform as well as other interpretable models (e.g., linear/logistic regression, decision trees, etc.) and similarly to standard NNs, when trained on the same data.

Motivating Example

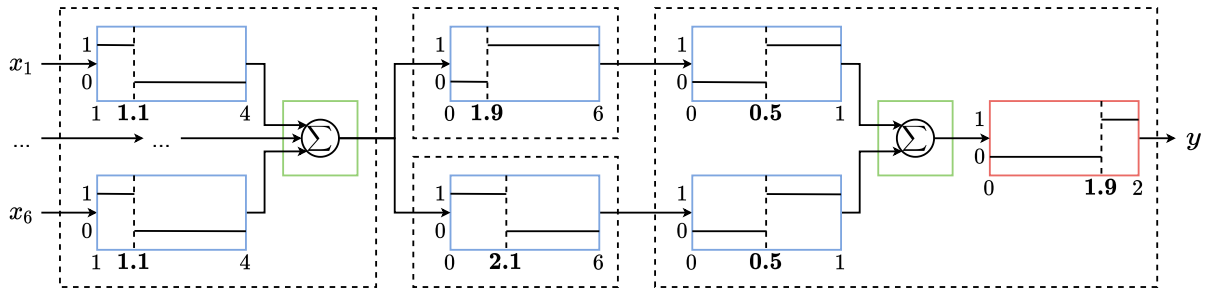


Figure 2.1: An example of a network for the MONK-2 dataset. x_i are the inputs, y is the output. The red and blue rectangles represent the plot of functions, with input range on the x -axis and output on the y -axis. The green rectangles contain the aggregation function. The numbers in bold represent the thresholds for the step functions.

Consider a simple dataset: MONK’s¹. Each sample consists of 6 attributes, which take integer values between 1 and 4 and a class label determined by a decision rule based on the 6 attributes. For example, in MONK-2, the rule that defines the class for each sample is the following: “*exactly two*” out of the six attributes are equal to 1.

It is impossible to intuitively recover rules from the parameter setting from a traditional, fully connected NN.

We shall see in the following that our main idea is that of inverting the activation and aggregation. In NEWRON the nonlinearity directly operates on the input of the neuron. The nonlinearity acts as a thresholding function to the input, making it directly interpretable as a (fuzzy) logical rule by inspecting its parameters. Consider the following network, represented in Figure 2.1: 2 hidden layers, the first with 1 neuron, the second with 2 neurons, and 1 output neuron. The x_i ’s are the inputs of the model, y is the output.

We present the form of a typical architecture composed by NEWRON in Figure 2.1. We show how we can interpret the parameters obtained from a trained network. The rectangles represent the plot of a function that divides the input domain into two intervals, separated by the number below the rectangle, taking values 1 and 0.

The functions that process the input give output 1 only if the input is less than 1.1, given that inputs are integers and assume values only in $\{1, 2, 3, 4\}$, this means “if $x_i = 1$ ”. The sum of the output of all these functions, depicted in the green rectangle, then represents the degree of soundness of those rules are.

The second layer has two neurons: the first outputs 1 if it receives an input greater than 1.9, i.e. if at least 2 of the rules $x_i = 1$ are valid, while the second outputs 1 if it receives an input less

¹<https://archive.ics.uci.edu/ml/datasets/MONK%27s+Problems>

than 2.1, i.e. if 2 or less of the rules $x_i = 1$ are valid. Notice that the two neurons are activated simultaneously only if $x_i = 1$ is true for exactly two attributes.

In the last layer, functions in the blue rectangles receive values in $\{0, 1\}$ and do not operate any transformation, keeping the activation rules unchanged. The sum of the outputs of these functions is then passed to the function in the red rectangle. This function outputs 1 only if the input is greater than 1.9. Since the sum is limited in $0, 1, 2$, this happens only when it receives 2 as input, which occurs only if the two central neurons are activated. As we have seen, this only applies if exactly 2 of the rules $x_i = 1$ are valid.

So we can conclude that the network gives output 1 just if “*exactly two*” of $\{x_1 = 1, x_2 = 1, x_3 = 1, x_4 = 1, x_5 = 1, x_6 = 1\}$ are true.

Contributions

The main contributions of this work are the following:

- We propose NEWRON, a generalization of the McCulloch-Pitts neuron allowing the definition of new artificial neurons. We show how special cases of NEWRON may pave the way towards interpretable, white-box neural networks.
- We prove the universal approximation theorem for three specializations of NEWRON, demonstrating that the new model does not lose any representation power in those cases.
- We experiment on several tabular datasets showing that NEWRON allows learning accurate Deep Neural models, beating interpretable by design models such as Decision Trees and Logistic Regression.

2.1.2 Related Work

[240] introduced the single artificial neuron: the Perceptron. The Perceptron resembles the functioning of the human/biological neuron, where the signal passing through the neuron depends on the intensity of the received signal, the strength of the synapses, and the receiving neuron’s threshold. In the same way, the Perceptron makes a linear combination of the inputs received and is only activated if the result exceeds a certain threshold. Over the years, various improvements to neural networks have been proposed: Recurrent Units, Convolutional Layers, and Graph Neural Networks, but for Fully Connected NNs, research efforts have mainly focused on finding more efficient activation functions [7]. Two works that have focused on modifying the internal structure of the neuron are those of [156], and [81]. In the former, a neuron is introduced that performs both a sum and a product of the inputs in parallel, applies a possibly different activation function for the two results, and then sums the two outcomes. Despite promising results, given the use of fewer parameters, better performance, and reduced training time compared to standard MLPs and RNNs, the proposed neuron, rather than being a generalization, is a kind of union between two standard neurons, one of which uses the product, instead of sum, as aggregation function. In the second paper, starting from the notion that the traditional neuron performs a first-order Taylor approximation, the authors propose a neuron using a second-order Taylor approximation. Although this improves the capacity of a single neuron, the authors do not demonstrate any gains in terms

of training time or convergence. Indeed, this can be considered a particular case of the higher-order neural units (HONUs) (see, e.g., [111]), i.e., a type of neurons that, by increasing the degree of the polynomial computed within them, try to capture the higher-order correlation between the input patterns. Recent works that focus on interpretation at neuron level ([63], [64], [122], [197]) often concentrate on extracting the most relevant neurons for a given task, but mostly deal with Recurrent or Convolutional neural networks. Although not designing an alternative version of the neuron, [331] proposes an alternative neural network structure, based on a Binning Layer, which divides the single input features into several bins, and a Kronecker Product Layer, which takes into account all the possible combinations between bins. The parameters estimated during training can be interpreted to translate the network into a decision tree through a clever design of the equations defining the network. Although interpretable, the main issue in this work is its scalability. The Kronecker Product Layer has an exponential complexity that makes training time unfeasible when the number of features grows.

2.1.3 The NEWRON Structure

A neuron, in the classical and more general case, is represented by the equation $y = f(b + \sum_{i=1}^n w_i x_i)$.

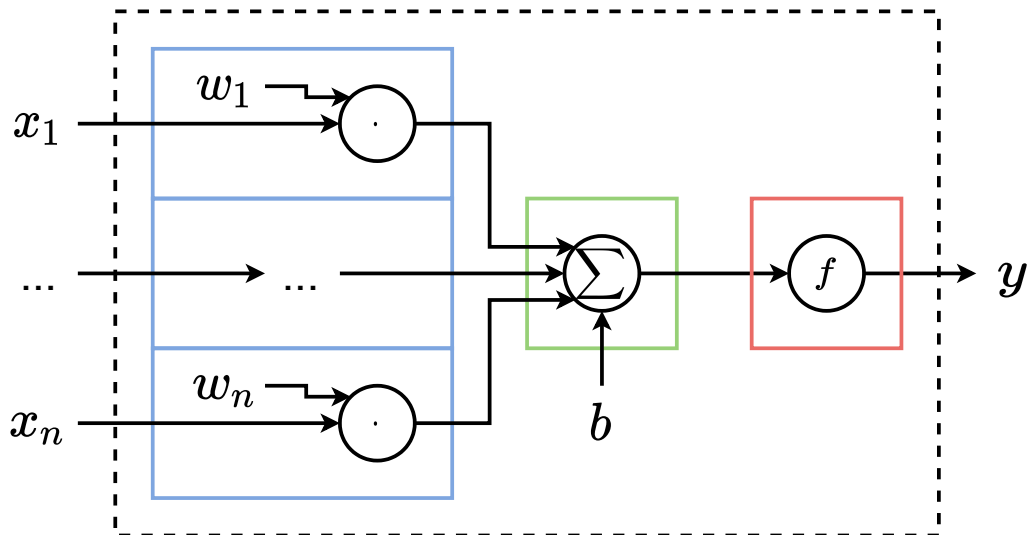


Figure 2.2: Structure of the standard artificial neuron. w_i and b are respectively weights and bias. f is the activation function. x_i 's are the inputs and y is the output.

b is called the bias, w_i are the weights, and x_i s are the inputs. f represents the activation function of the neuron. Usually, we use the sigmoid, hyperbolic tangent, or ReLU functions.

We first generalize the above equation, introducing NEWRON as follows:

$$y = f(G_{i=1}^n(h_i(x_i))) \quad (2.1)$$

Each input is first passed through a function h_i , which we will call *processing function*, where the dependence on i indicates different parameters for each input. G , instead, represents a generic aggregation function.

Using NEWRON notation, the standard artificial neuron would consist of the following: $h_i(x_i) = w_i x_i$, $G = \sum$, and $f(z) = f^*(z + b)$.

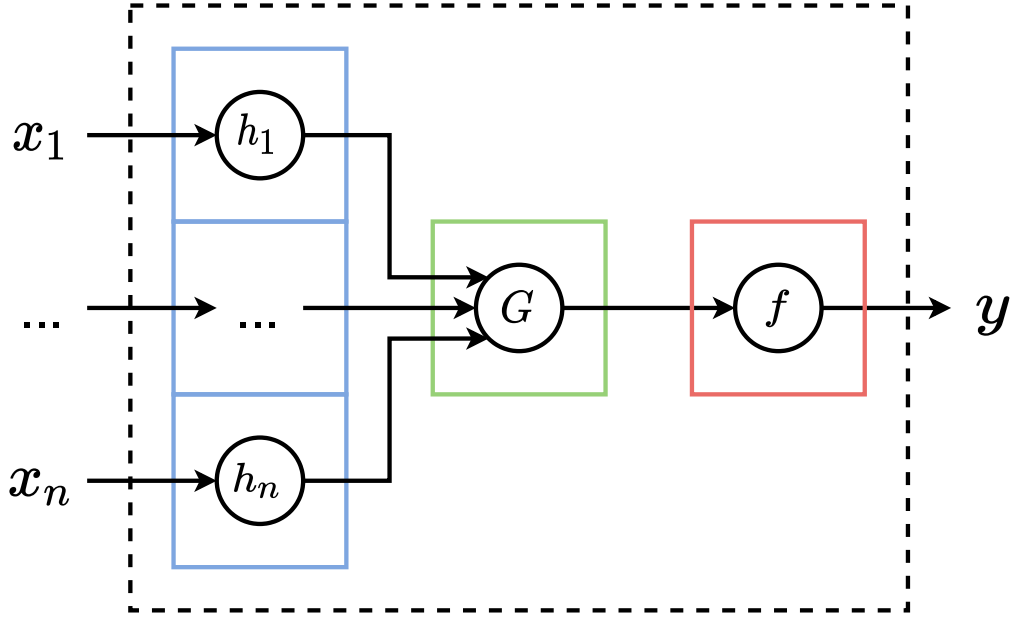


Figure 2.3: Structure of NEWRON, the generalized artificial neuron. The blue rectangles represent the processing function sections, the green rectangles contain the aggregation function, and the red rectangles represent the activation part. Same colors are also used in Figure 2.2

G does not have any parameters, while b parametrizes the activation function.

Inverted Artificial Neuron (IAN)

We present 3 novel structures characterized by an inversion of the aggregation and activation functions. We name this architectural pattern: Inverted Artificial Neuron (IAN). In all the cases we consider the sum as the aggregation function and do not use any activation function: $G = \sum$, and $f(z) = z$.

Heaviside IAN

The first case we consider uses a unit step function as activation. This function, also called the Heaviside function, is expressed by the following equation:

$$H(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases} \quad (2.2)$$

According to (2.1) we can define the processing function as follows:

$$h_i(x_i) = H(w_i(x_i - b_i)) = \begin{cases} H(w_i) & x_i \geq b_i \\ 1 - H(w_i) & x_i < b_i \end{cases} \quad (2.3)$$

where w_i and b_i are trainable parameters.

Sigmoid IAN

We cannot train the Heaviside function using gradient descent, and it represents a decision rule that in some cases is too restrictive and not “fuzzy” enough to deal with constraints that are not

clear-cut.

A natural evolution of the unit step function is therefore the sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}}$. This function ranges in the interval $(0, 1)$, is constrained by a pair of horizontal asymptotes, is monotonic and has exactly one inflection point.

The sigmoid function can be used as a processing function with the following parameters:
 $h_i(x_i) = \sigma(w_i(x_i - b_i))$.

Product of tanh IAN

Another option we consider as a processing function is the multiplication of hyperbolic tangent (tanh). For simplicity, we will use the term “tanh-prod”.

The tanh function $\tanh(x) = \frac{e^{2x}-1}{e^{2x}+1}$ is on its own very similar to the sigmoid. An interesting architecture is that using M tanh simultaneously. Each tanh applies its own weights, on each individual input.

While the sigmoid is monotonic with only one inflection point, roughly dividing the input space into two sections, the multiplication of tanh, by being not monotonic, allows us to divide the input space into several intervals. The multiplication would remain in $(-1, 1)$, but can be easily rescaled to $(0, 1)$.

We can therefore write the processing function in the case of the tanh multiplication as follows:

$$h_i(x_i) = \frac{\left(\prod_{m=1}^M \tanh(w_{im}(x_i - b_{im}))\right) + 1}{2} \quad (2.4)$$

Note how, in this case, the weights depend on both the input i and the m -th function. Such a neuron will therefore have M times more parameters than the Heaviside and sigmoid cases.

Output layer

The output layer would produce values ranging in the interval $(0, N)$ ($\{0, 1, \dots, N\}$ for the Heaviside case), where N represents the number of neurons in the penultimate layer. This is because the last neuron makes the sum of N processing functions restricted in the interval $(0, 1)$ ($\{0, 1\}$ for the Heaviside case). To allow the last layer to have a wider output range and thus make our network able to reproduce a wider range of functions, we modify the last layer processing function h_i^* as follows: $h_i^*(x_i) = \alpha_i h_i(x_i)$,

where α_i are trainable parameters.

In the same way, as for a traditional neural network, it is important, in the output layer, to choose an adequate activation function. We need, indeed, to match the range of the output of the network and the range of the target variable. In particular, in the case of output in $(0, 1)$, we use a sigmoid centered in b^* : $f^*(z) = \sigma(z - b^*)$

In the case of a classification problem with more than 2 classes, a softmax function ($s(z_j) = \frac{e^{z_j}}{\sum_l e^{z_l}}$) is used to output probabilities.

Note(s)

The writing $w(x - b)$ is theoretically identical to that $w^*x + b^*$, where simply $w^* = w$ and $b^* = -bw$. This notation allows us to interpret the weights directly. From b , we already know the inflection

point of the sigmoid; while looking at w , we immediately understand its direction.

2.1.4 Interpretability

[9] presented a well-structured overview of concepts and definitions in the context of Explainable Artificial Intelligence (XAI).

They make a distinction among the various terms that are mistakenly used as synonyms for interpretability. According to them:

- **Interpretability:** is seen as a passive feature of the model and represents the ability of a human to understand the underlying functioning of a decision model, focusing more on the cause-effect relationship between input and output.
- **Transparency:** very similar to interpretability, as it represents the ability of a model to have a certain degree of interpretability. There are three categories of transparency, representing the domains in which a model is interpretable. Simulatable models can be emulated even by a human. Decomposable models must be explainable in their individual parts. For algorithmically transparent models, the user can understand the entire process followed by an algorithm to generate the model parameters and how the model produces an output from the input.
- **Explainability:** can be seen as an active feature of a model, encompassing all actions that can detail the inner workings of a model. The explanation represents a kind of interface between a human and the model and must at the same time represent well the functioning of the model and be understandable by humans.

In this paper, we show decomposable models that, in some cases, are also algorithmically transparent.

Heaviside

The interpretability of an architecture composed of Heaviside IANs has to be analyzed by discussing its four main sections separately.

First layer - Processing function

A single processing function $h(x) = H(w(x-b))$ divides the space of each variable x in two half-lines starting from b , one of which has a value of 1 and one of which has a value of 0, depending on the sign of w .

Aggregation

Using sum as the aggregation function, the output takes values in $\{0, 1, \dots, n\}$; where 0 corresponds to a deactivation for each input, and n represents an activation for all inputs, and the intermediate integer values $\{1, 2, \dots, k, \dots, n-1\}$ represent activation for k of inputs.

$$y = \sum_{i=1}^n h_i^* = \begin{cases} n & h_i^* = 1 \forall i \in \{1, \dots, n\} \\ k & h_i^* = 1 \ i \in S \subseteq \{1, \dots, n\}, |S| = k \\ 0 & h_i^* = 0 \forall i \in \{1, \dots, n\} \end{cases} \quad (2.5)$$

where we simplified the notation using $h_i^* = h_i(x_i)$.

2+ Layer - Processing function

Let us define an M -of- N rule as true if at least M of the N rules of a given set are true.

The Heavisides of the layers after the first one receive values in $\{0, 1, \dots, n\}$, where n represents the number of inputs of the previous layer. In the case where $0 \leq b \leq n$ and $w > 0$, the Heaviside will output 1 only if the input received is greater than or equal to b , therefore only if at least $\lceil b \rceil$ of the rules R_i of the previous layer are true, which corresponds to a rule of the type $\lceil b \rceil$ -of- n $\{R_1, R_2, \dots, R_n\}$. In the opposite case, where $0 \leq b \leq n$ and $w < 0$, Heaviside will output 1 only if the input received is less than or equal to b , so only if no more than $\lfloor b \rfloor$ of the rules of the previous layer are true. This too can be translated to an M -of- N rule, inverting all rules R_j and setting M as $\lceil n - b \rceil$: $\lceil n - b \rceil$ -of- n $\{\neg R_1, \neg R_2, \dots, \neg R_n\}$.

Last layer - Aggregation

In the last layer we have to account for the α factors used to weigh the contribution of each input:

$$y = \sum_{i=1}^n \alpha_i h_i(x_i) = \sum_{i=1}^n \alpha_i H(w_i(x_i - b_i)) \quad (2.6)$$

We have an activation rule for each of the n Heavisides forcing us to calculate all the 2^n possible cases. The contribution of each input is exactly α_i . So, the output corresponds to the sum of the α_i 's for each subset of inputs considered.

Sigmoid

In the case of sigmoid IAN, b_i represents the inflection point of the function, while the sign of w_i tells us in which direction the sigmoid is oriented; if positive, it is monotonically increasing from 0 to 1, while if negative, it is monotonically decreasing from 1 to 0. The value of w_i indicates how fast it transitions from 0 to 1, and if it tends to infinity, the sigmoid tends to the unit step function.

Sigmoid Interpretation

The sigmoid can be interpreted as a fuzzy rule of the type $x_i > b_i$ if $w_i > 0$ or $x_i < b_i$ if $w_i < 0$, where the absolute value of w_i indicates how sharp the rule is. The case $w_i = 0$ will always give value 0.5, so that the input does not have any influence on the output.

If w_i is very large, the sigmoid tends to the unit step function. If, on the other hand, w_i takes values for which the sigmoid in the domain of x_i resembles a linear function, what we can say is that there is a direct linear relationship (or inverse if $w_i < 0$) with the input.

The fuzzy rule can be approximated by its stricter version $x_i > b_i$, interpreting fall under the methodology seen for Heaviside. However, this would result in an approximation of the operation of the network.

It is more challenging to devise clear decision rules when we add more layers. Imagine, as an example, a second layer with this processing function:

$$h(y) = \sigma(w^*(y - b^*)) \quad (2.7)$$

where y is the aggregation performed in the previous layer of the outputs of its processing functions, its value roughly indicates how many of the inputs are active. In the second layer, consider as an example a value of $w^* > 0$. To have an activation, this means that we might need k inputs greater than or equal to b^*/k . Although this does not deterministically indicate how many inputs we need to be true, we know how the output changes when one of the inputs changes.

The last case to consider takes into account the maximum and minimum values that the sigmoid assumes in the domain of x . If they are close to each other, that happens when w is very small, the function is close to a constant bearing no connection with the input.

Product of tanh

The multiplication of tanh has more expressive power, being able to represent both what is represented with the sigmoid, as well as intervals and quadratic relations.

tanh-prod Interpretation

In this case, it is not possible to devise as quickly as in the previous case decision rules. Indeed, it is still possible to observe the trend of the function and draw some conclusions. When the product of the two tanh resembles a sigmoid, we can follow the interpretation of the sigmoid case. In other cases, areas with quadratic relations can occur, i.e., bells whose peak indicates a more robust activation or deactivation for specific values.

Summary of Interpretation

The advantage of this method lies in the fact that it is possible to analyze each input separately in each neuron, thus easily graph each processing function. Then, based on the shape taken by the processing function, we can understand how the input affects the output of a neuron.

The Heaviside is the most interpretable of our models, allowing a direct generation of decision rules.

Sigmoid and tanh-prod cases depend on the parameter w . When it is close to 0, the activation is constant regardless of the input. When w is large enough, the processing function is approximately a piecewise constant function taking only values 0 and 1.

In all the other cases, the processing function approximates a linear or bell-shaped function. Even if we can not derive exact decision rules directly from the model, in these cases, we can infer a linear or quadratic relation between input and output.

Each layer aggregates the interpretations of the previous layers. For example, the processing function of a second layer neuron gives a precise activation when its input is greater than a certain

threshold, i.e., the bias b of the processing function. The output of the neuron of the first layer must exceed this threshold, and this happens if its processing functions give in output values whose sum exceeds this threshold.

A separate case is the last layer, where the α parameters weigh each of the interpretations generated up to the last layer.

We can interpret a traditional individual neuron as a linear regressor. However, when we add more layers, they cannot be interpreted. Our structure, instead, remains interpretable even as the number of layers increases.

Dataset	IAN models			Interpretable models		Non-interpretable models	
	Heaviside	sigmoid	tanh-prod	LR	DT	GBDT	NN
adult	80.2 (± 0.06)	82.6 (± 0.05)	82.3 (± 0.06)	76.2 (± 0.07)	81.5 (± 0.06)	<u>87.5 (± 0.05)</u>	83.1 (± 0.06)
australian	86.5 (± 0.51)	87.0 (± 0.5)	88.7 (± 0.4)	88.7 (± 0.4)	87.0 (± 0.41)	<u>90.2 (± 0.47)</u>	88.0 (± 0.4)
b-c-w	98.9 (± 0.16)	98.9 (± 0.16)	98.9 (± 0.16)	97.8 (± 0.23)	97.7 (± 0.23)	<u>98.3 (± 0.21)</u>	<u>98.9 (± 0.17)</u>
car	95.1 (± 0.2)	95.9 (± 0.21)	100.0 (± 0.0)	51.4 (± 0.45)	98.5 (± 0.11)	<u>100.0 (± 0.0)</u>	<u>99.8 (± 0.04)</u>
cleveland	65.6 (± 1.02)	60.1 (± 1.1)	62.9 (± 1.13)	60.8 (± 1.13)	53.6 (± 1.19)	<u>61.5 (± 1.01)</u>	<u>65.6 (± 1.01)</u>
crx	86.2 (± 0.51)	85.4 (± 0.58)	86.5 (± 0.5)	84.6 (± 0.45)	88.0 (± 0.42)	82.9 (± 0.58)	<u>87.7 (± 0.44)</u>
diabetes	73.3 (± 0.56)	72.7 (± 0.68)	76.1 (± 0.61)	75.6 (± 0.6)	74.1 (± 0.63)	<u>75.1 (± 0.64)</u>	<u>74.2 (± 0.65)</u>
german	78.2 (± 0.53)	77.0 (± 0.53)	75.5 (± 0.52)	75.1 (± 0.52)	68.3 (± 0.57)	<u>76.6 (± 0.55)</u>	<u>76.7 (± 0.54)</u>
glass	77.0 (± 1.17)	81.6 (± 1.04)	85.6 (± 1.02)	72.1 (± 1.08)	72.7 (± 1.19)	<u>87.3 (± 0.9)</u>	<u>82.5 (± 0.91)</u>
haberman	76.9 (± 0.94)	76.1 (± 0.92)	77.2 (± 0.88)	73.0 (± 1.05)	64.4 (± 1.08)	<u>72.5 (± 1.09)</u>	<u>76.1 (± 0.92)</u>
heart	88.7 (± 0.67)	86.3 (± 0.85)	82.7 (± 0.8)	82.4 (± 0.95)	81.4 (± 1.02)	<u>81.7 (± 0.98)</u>	<u>82.9 (± 0.95)</u>
hepatitis	84.7 (± 1.26)	85.1 (± 1.23)	82.5 (± 1.16)	79.1 (± 1.45)	79.1 (± 1.33)	<u>81.7 (± 1.32)</u>	<u>82.4 (± 1.13)</u>
image	93.0 (± 0.11)	94.0 (± 0.1)	94.4 (± 0.09)	90.4 (± 0.12)	90.6 (± 0.12)	<u>95.8 (± 0.08)</u>	<u>92.6 (± 0.11)</u>
ionosphere	94.4 (± 0.48)	96.7 (± 0.34)	96.5 (± 0.37)	92.0 (± 0.51)	94.5 (± 0.45)	<u>95.4 (± 0.37)</u>	<u>96.7 (± 0.34)</u>
iris	100.0 (± 0.0)	100.0 (± 0.0)	100.0 (± 0.0)	100.0 (± 0.0)	97.3 (± 0.52)	<u>97.3 (± 0.52)</u>	<u>100.0 (± 0.0)</u>
monks-1	94.4 (± 0.21)	100.0 (± 0.0)	100.0 (± 0.0)	66.0 (± 0.46)	90.6 (± 0.27)	<u>100.0 (± 0.0)</u>	<u>100.0 (± 0.0)</u>
monks-2	100.0 (± 0.0)	100.0 (± 0.0)	100.0 (± 0.0)	54.5 (± 0.45)	82.7 (± 0.33)	<u>94.2 (± 0.21)</u>	<u>87.6 (± 0.27)</u>
monks-3	97.1 (± 0.15)	97.1 (± 0.15)	97.1 (± 0.15)	81.2 (± 0.31)	97.2 (± 0.16)	<u>96.2 (± 0.16)</u>	<u>90.3 (± 0.25)</u>
sonar	93.3 (± 0.74)	96.8 (± 0.48)	95.2 (± 0.53)	89.5 (± 0.75)	83.4 (± 0.98)	<u>88.1 (± 0.9)</u>	<u>89.4 (± 0.87)</u>
bisector	98.9 (± 0.13)	99.3 (± 0.09)	99.3 (± 0.09)	100.0 (± 0.0)	97.7 (± 0.18)	<u>98.3 (± 0.16)</u>	<u>100.0 (± 0.0)</u>
xor	100.0 (± 0.0)	100.0 (± 0.0)	99.2 (± 0.11)	53.2 (± 0.65)	99.2 (± 0.12)	<u>100.0 (± 0.0)</u>	<u>100.0 (± 0.0)</u>
parabola	98.8 (± 0.15)	100.0 (± 0.0)	99.6 (± 0.07)	77.8 (± 0.52)	97.6 (± 0.18)	<u>97.7 (± 0.17)</u>	<u>100.0 (± 0.0)</u>
circle	96.8 (± 0.22)	99.3 (± 0.1)	99.6 (± 0.07)	52.4 (± 0.67)	98.8 (± 0.13)	<u>97.6 (± 0.2)</u>	<u>99.2 (± 0.11)</u>

Table 2.1: Datasets accuracy ($\pm 95^{th}$ percentile standard error) results of the best performing model. In **bold** we indicate the best performing model amongst the interpretable ones. If GBDT or NN exceeds this accuracy, the corresponding result is underlined.

2.1.5 Universality

A fundamental property of neural networks is that of universal approximation. Under certain conditions, multilayer feed-forward neural networks can approximate any function in a given function space. In [62] it is proved that a neural network with a hidden layer and using a continuous sigmoidal activation function is dense in $C(I_n)$, i.e., the space of continuous functions in the unit hypercube in \mathbb{R}^n . [128] generalized to the larger class of all sigmoidal functions.

To make the statement of theorems clearer we recall that the structure of a two-layer network with IAN neurons and a generic processing function h is

$$\psi(x) = \sum_{j=1}^N \alpha_j h_j \left(\sum_{i=1}^n h_{ij}(x_i) \right) \quad (2.8)$$

where $\alpha_j \in \mathbb{R} \forall j \in \{1, \dots, N\}$.

Dataset	IAN models			Interpretable models		Non-interpretable models	
	Heaviside	sigmoid	tanh-prod	LR	DT	GBDT	NN
adult	80.2	82.6	82.3	76.2	81.5	<u>87.5</u>	83.1
australian	86.5	87.0	88.7	88.7	87.0	<u>90.2</u>	88.0
b-c-w	98.9	98.9	98.9	97.8	97.7	98.3	<u>98.9</u>
car	95.1	95.9	100.0	51.4	98.5	<u>100.0</u>	99.8
cleveland	65.6	60.1	62.9	60.8	53.6	61.5	<u>65.6</u>
crx	86.2	85.4	86.5	84.6	88.0	82.9	87.7
diabetes	73.3	72.7	76.1	75.6	74.1	75.1	74.2
german	78.2	77.0	75.5	75.1	68.3	76.6	76.7
glass	77.0	81.6	85.6	72.1	72.7	<u>87.3</u>	82.5
haberman	76.9	76.1	77.2	73.0	64.4	72.5	76.1
heart	88.7	86.3	82.7	82.4	81.4	81.7	82.9
hepatitis	84.7	85.1	82.5	79.1	79.1	81.7	82.4
image	93.0	94.0	94.4	90.4	90.6	<u>95.8</u>	92.6
ionosphere	94.4	96.7	96.5	92.0	94.5	95.4	<u>96.7</u>
iris	100.0	100.0	100.0	100.0	97.3	97.3	<u>100.0</u>
monks-1	94.4	100.0	100.0	66.0	90.6	<u>100.0</u>	<u>100.0</u>
monks-2	100.0	100.0	100.0	54.5	82.7	94.2	87.6
monks-3	97.1	97.1	97.1	81.2	97.2	96.2	90.3
sonar	93.3	96.8	95.2	89.5	83.4	88.1	89.4
bisector	98.9	99.3	99.3	100.0	97.7	98.3	<u>100.0</u>
xor	100.0	100.0	99.2	53.2	99.2	<u>100.0</u>	<u>100.0</u>
parabola	98.8	100.0	99.6	77.8	97.6	97.7	<u>100.0</u>
circle	96.8	99.3	99.6	52.4	98.8	97.6	99.2

Table 2.2: Datasets accuracy results of the best performing model. In **bold** we indicate the best performing model amongst the interpretable ones. If GBDT or NN exceeds this accuracy, the corresponding result is underlined.

When the processing function is the Heaviside function we proved that the network can approximate any continuous function on unit hypercube, I_n , Lebesgue measurable functions on I_n and functions in $L^p(A, \mu)$ for $1 \leq p < \infty$, with μ being a Radon measure and $A \in \mathcal{B}(\mathbb{R}^n)$ a Borel set. More precisely, the following theorems hold; we detail the proofs of the theorems in the appendix.

Theorem 5.1. *When the processing function is the Heaviside function the finite sums of the form (2.8) are dense in $L^p(A, \mu)$ for $1 \leq p < \infty$, for any $A \in \mathcal{B}(\mathbb{R}^n)$ - \mathcal{B} denote the Borel σ -algebra - and μ Radon measure on $(A, \mathcal{B}(A))$.*

Theorem 5.2. *When the processing function is the Heaviside function, the finite sums of the form (2.8) are dense in the space of Lebesgue measurable functions on I_n w.r.t the convergence in measure.*

Theorem 5.3. *Given $g \in C(I_n)$ and given $\epsilon > 0$ there is a sum $\psi(x)$ of the form (2.8) with Heaviside as processing function such that*

$$|\psi(x) - g(x)| < \epsilon \quad \forall x \in I_n.$$

When the processing function is the sigmoid function or tanh-prod, we proved that the finite sums of the form (2.8) are dense in $C(I_n)$.

Theorem 5.4. *When the processing function is a continuous sigmoidal function the finite sums of the form (2.8) are dense in $C(I_n)$.*

Theorem 5.5. *Let $\psi(x)$ be the family of networks defined by the equation (2.8) when the processing function is given by (2.4). This family of functions is dense in $C(I_n)$.*

2.1.6 Experiments

Datasets

We selected a collection of datasets from the UCI Machine Learning Repository. We only consider classification models in our experiments. However, it is straightforward to apply NEWRON architectures to regression problems. The description of the datasets is available at the UCI Machine Learning Repository website or the Kaggle website.

We also used 4 synthetic datasets of our creation, composed of 1000 samples with 2 variables generated as random uniforms between -1 and 1 and an equation dividing the space into 2 classes. The 4 equations used are bisector, xor, parabola, and circle.

We give more details about the datasets in the appendix.

Methods

We run a hyperparameter search to optimize the IAN neural network structure, i.e., depth and number of neurons per layer, for each dataset. We started our search by trying a single neuron, followed by a shallow network and then continued through various DNN configurations. We tested IAN with all three different processing functions. In the tanh-prod case, we set $M = 2$.

Concerning the training of traditional neural networks, we tested the same structures used for NEWRON, i.e., the same number of layers and neurons. Finally, we also ran a hyperparameter search to find the best combinations in the case of Logistic Regression (LR), Decision Trees (DT), and Gradient Boosting Decision Trees (GBDT). We include all the technical details on the methods in the appendix.

Results

Table 2.1 presents on each row the datasets used while on the columns the various models. Each cell contains the 95% confidence interval for the accuracy of the model that obtains the best performance.

Results obtained with the new IAN neurons are better than those obtained by DTs and LRs (interpretable) models. Moreover, IAN's results are on par, sometimes better than, results of traditional NNs and GBDT classifiers. These last two methods, though, are not transparent.

Amongst the Heaviside, sigmoid, and tanh-prod cases, we can see that the first one obtains the worst results. The reason may be that it is more challenging to train, despite being the most interpretable among the three cases. tanh-prod instead performs slightly better than sigmoid, being more flexible. Sigmoid, being more straightforward to interpret than tanh-prod, could be a good choice at the expense of a slight decrease in accuracy that remains, however, similar to that of a traditional neural network.

Circle dataset example

In order to first validate our ideas, we show what we obtained by applying a single neuron using multiplication of 2 tanh in the case of our custom dataset circle.

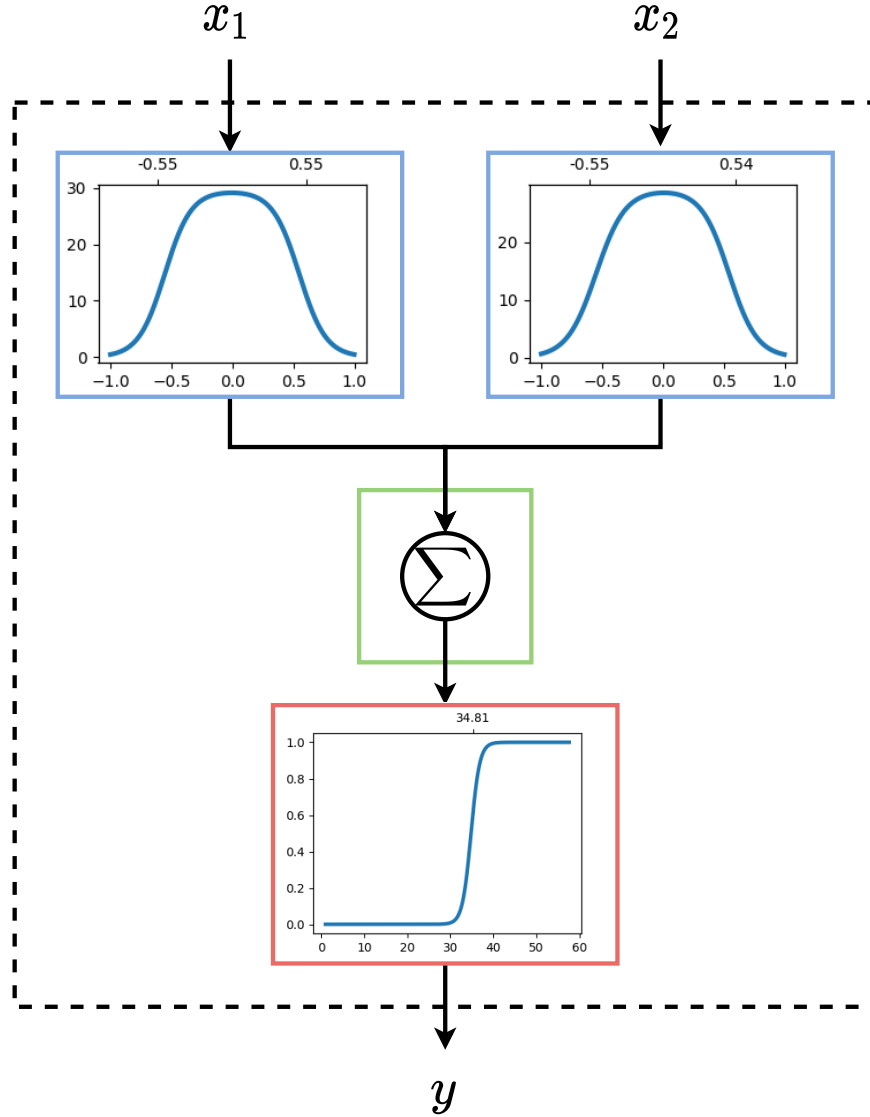


Figure 2.4: tanh-prod Neural Network trained on the circle dataset. The figure follows the color convention used for NEWRON in Figure 2.3. x_1 and x_2 are the inputs of the network and y is the output. The processing and activation functions are plotted with input on the x -axis and output on the y -axis. Coordinates of the inflection points are indicated above the plots.

In Figure 2.4 we can see how the multiplication of tanh has converged to two bells centred in 0, while α_1 and α_2 have gone to 30. According to the IANinterpretation method, values below 30 correspond to an activation function output of 0, while it is 1 for values above 38. In the middle range, the prediction is more uncertain. Combining this data with the previous prediction, we can conclude that we need the sum of the two values output by the two processing functions to be greater than 38 to have a prediction of class 1. Therefore, if one of the two inputs is 0 (output 30), it is enough for the other to be between -0.65 and 0.65 (output greater than 8). Otherwise, we may need an output of at least 19 from both outputs, corresponding to input values between -0.5

and 0.5, i.e., the area covered by the circle. We show more examples in the appendix.

Current limitations

The extraction of proper rules from the network can be harrowing; in the Heaviside case, they might be too long in the sigmoid and tanh-prod cases because their simplicity depends on the final value parameters. Nevertheless, methods of regularization during training or additional Rule Extraction methods may help to simplify interpretability. We defer the study of regularization to future works.

Also, we have not compared NEWRON against state-of-the-art Deep Learning models for tabular data, as our main goal was to show that our formulation was more suitable than traditional neurons compared to “traditional” interpretable models. Comparisons with more advanced solutions for tabular data will be the subject of future work.

2.1.7 Conclusions and Future Work

We have introduced the concept of a generalized neuron and proposed three different specializations, along with the corresponding method to interpret the behavior of the network. Also, in cases where from the network we cannot devise exact rules (e.g., in the sigmoid and tanh-prod cases), the structure of the neuron and the parameters allow the visualization of its behavior. Indeed, for every input, we apply the nonlinearity operation before the aggregation reducing it to a one-dimensional space allowing the analysis of each input separately. Through universal approximation theorems, we have proved that the new structure retains the same expressive power as a standard neural network. In future studies we will investigate more in detail the expressiveness of IAN based models with respect to the number of layers or neurons in arbitrarily deep but width-limited networks and arbitrarily wide but depth-limited networks. Experiments conducted on both real and synthetic datasets illustrate how our framework can outperform traditional interpretable models, Decision Trees, and Logistic Regression, and achieve similar or superior performance to standard neural networks. In the future, we will investigate the influence of hyper-parameters (network depth, number of neurons, processing functions) and initialization on the model quality. Also, we will refine the analysis of the tanh-prod case as the number of tanh increases. In addition, we will investigate IAN with additional processing functions, such as ReLU and SeLU. Finally, we will extend this method to other neural models, such as Recurrent, Convolutional and Graph Neural Networks. Although we have not yet defined exactly how to extend to the other cases, the general idea remains the same: avoid linear combinations, instead apply a function to each input and then aggregate the results. Since CNNs are in fact a special case of Fully-connected NNs with certain weights fixed and/or shared, our neuron would already be applicable to images, but the interpretation for this case will require more investigation.

2.2 Explaining Neural Networks Using a Rule-set Based on Interpretable Concepts

We propose a new category of interpretable machine learning models for tabular data, called self-explainable, which are able to distill interpretable concepts in the form of rules during training for classification. The concepts are then used to produce global explanations as First Order Logic rulesets, linking concepts to classes. The architecture we show in this paper integrates two recent variants of traditional neural networks, namely Inverted Artificial Neuron and Logic Explained Network. We also solve problems that afflicted both variants, such as discovery of concepts and oversized rulesets. Our experiments show that self-explainable neural networks can directly produce rules explaining their predictions (with an average fidelity of 89% across 18 datasets), performing similar to classical interpretable classifiers in terms of classification accuracy (given the same optimization time), with a better performance on half of the datasets tested. Our models also obtain a performance similar to traditional multilayer feedforward artificial neural networks (2.0% more accuracy on average).

2.2.1 Introduction

Artificial Neural Networks (ANNs) have become predominant in many domains thanks to their remarkable performance [102, 253, 318], but they conceal the undesired pitfall of being black-box models. Precisely for this reason, the last few years have seen the emergence of an increasing number of works focused on explaining how Neural Networks (NNs) operate [9].

Within Explainable Artificial Intelligence (XAI) however, much work focuses on providing post-hoc explainability methods applicable to traditional networks [9]. While achieving impressive results, these methods struggle against the black-box nature of neural networks [42, 91, 207].

In contrast, a network that was built to be explainable by design could greatly facilitate the extraction of explanations. To this end, in this paper we propose a neural network that can extract interpretable concepts from the features of a tabular dataset. Current approaches [149, 153, 334] to concept-based interpretations assume the training dataset already contains associations between concepts and samples. In other words, instead of having a data sample made up of pairs (x, y) , the dataset contains triples (x, c, y) , where c is the concept vector associated with x . The network can then be trained to predict them. This is different from what we have done in this work, because the concepts are distilled from the dataset itself. Moreover, distilled concepts are interpretable and not sample-based, because they take the form of feature-based rules. Using these concepts, the network is able to (i) solve a classification task and (ii) provide explanations for its predictions.

To build this network, we combine two models: a specialization of NEWRON [269], specifically the Inverted Artificial Neuron (IAN), and the Entropy-based version [21] of Logic Explained Networks (LENs) [55]. The IAN is a variant of the traditional artificial neuron which is able to extract rules (in other words, *concepts*) from raw features. IAN is then followed by a Logic Explained

Network, a model that can provide explanations in the form of concise concept-based rulesets. In this paper we start by firstly addressing two limitations of IAN and LEN architectures: (i) we solve the Inverted Artificial Neuron problem of producing overly large rulesets, which are difficult to interpret; and (ii) we can generate interpretable concepts in the form of a feature-based rule. The use of this type of neuron makes it possible to overcome LEN’s inability to produce concepts.

The major contributions we are making in this work can be summarized as follows:

- By combining IAN and LEN we are able to overcome their limitations when used on their own: we allow extracting interpretable concepts used for generating concise rulesets.
- Because the network is optimizable end-to-end, concepts are distilled directly during network training and do not require a ground truth or a specific loss.
- Our self-explaining neural network provides a good trade-off between accuracy and explainability, being able to directly explain its predictions through First Order Logic.
- Conducted experiments show that this type of network can achieve a performance, in terms of accuracy, similar to traditional black-box neural networks and classical interpretable classifiers on tabular data.

2.2.2 Related Work

Rule-extraction from Neural Network Despite the great number of recent works in the field of Explainable AI, a small amount of them focus on the extraction of rules from neural networks [9]. Some early work on understanding neural networks using rules dates back to the 1990s [91, 255, 256]. Rule extraction algorithms [6, 12, 65, 113, 137] are mainly divided into two groups: decompositional, which analyze the activation of individual neurons to extract rules, and pedagogical, which define rules that most closely replicate the output of the network given the input. A third type of rule extraction, called eclectic, incorporates elements of both the other types decompositional and pedagogical.

Of decompositional approaches, we can mention NeuroLinear [257], GRG [204] and NeuroRule [175] that work by clustering the hidden unit activation values. [296] algorithm operates by approximating each neuron with a boolean function, while [295] build an algorithm that is able to extract both “if-then” as well as “MofN” rules. [119, 259] tried to add discretized inputs to train the network to increase both performance and facilitate rule extraction, while DeepRED [358] extracts rules for each NN layer and then merge them.

Concerning pedagogical approaches, both Re-RX [258], REANN [140] and RxREN [11] focus on pruning insignificant neurons, generating and then eliminating insignificant rules, while HYPINV [245] tries to find hyper-planes that approximate the neural network decision boundary.

The eclectic approach of ERENN_MHL [43] performs a rule extraction from each layer, different for the first layer, and then combines and refines them to increase performance and reduce complexity. In contrast, CGA [129] makes use of a clustering genetic algorithm to group hidden unit activation values. While Inverted Artificial Neuron [269] falls into the decompositional type of rule extraction, LEN [55] is a pedagogical type of rule extraction algorithm. Our complete rule extraction method, can therefore be considered an eclectic type approach.

Tabular data

Neural networks achieved outstanding results in many areas, showing that they are capable of effectively process a variety of data types: tabular [33, 132], time series [169, 184], images [78, 109], graphs [323, 353], etc. Nevertheless, for tabular data, it is still unclear whether neural networks are indeed the best model. Models based on Gradient Boosting, such as Gradient Boosted Decision Trees, still often emerge as better [33, 107].

Recent research papers try to outperform Gradient Boosting techniques [2, 8, 103] without success, while others [85, 139] propose improvements to neural networks, stating that they had finally managed to outperform models based on Boosting. Papers such as [267] instead disprove some of these claims by expanding the results to more datasets and rigorously addressing hyperparameters tuning, achieving once again superior results with Gradient Boosting.

While the issue of performance on tabular data is still debated, our work does not aim to beat Boosting models in terms of performance, but rather to achieve performance similar to a standard neural network with the added quality of being interpretable.

It should also be added that the restriction to tabular data is, in part, also forced by the fact that a major component of the network, i.e. IAN, has for now been analyzed in depth only for this type of data and not yet extended to others.

Concept-based Explainability A growing branch in the field of XAI is that of Concept-based Explainability [98, 149, 334]. Concepts are a method of getting closer to the human type of reasoning. Humans often aggregate information into high-level concepts that best describe the situation they are in, and then use a set of rules to determine an action [55]. Concepts are indeed more intuitive to understand as opposed to raw features.

Concept-based explainability methods are used for Recurrent Neural Networks [144], Convolutional Neural Networks [49, 153], and Graph Neural Networks [94, 182]. On tabular data, on the other hand, the concept-based literature is sparse [148, 216].

To the best of our knowledge, in other Concept-based Explainability work, concepts are already present in the dataset or defined in advance [153, 351]. *In contrast, our network is able to distill them jointly with its training.*

Moreover, in other papers [98, 182] concepts are often interpreted by selecting a set of samples and analyzing the similarities among them. In our case, the extracted concepts are directly interpretable since they are in fact logical rules applicable to any given samples.

2.2.3 Methodology

NEWRON & Inverted Artificial Neuron NEWRON [269] is a generalization of traditional artificial neurons. Equation (2.9) shows NEWRON's structure: input features x_i 's are passed through the processing function h_i , are aggregated through function G (iteratively from $i = 1$ to N). An activation function f is then applied to give the output y . The processing functions can take the form of any complex operations with corresponding trainable parameters. The standard neuron is obtained by choosing multiplication by weights as the processing function and the sum as the aggregation function.

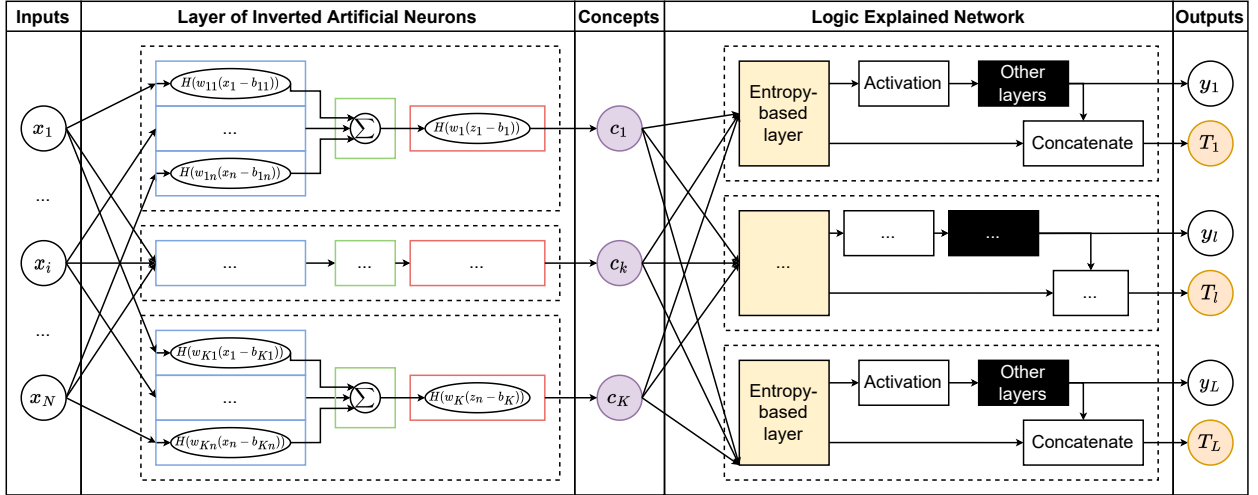


Figure 2.5: Structure of the self-explainable neural network. x_i represents the i -th input. The first layer of the network consists of Inverted Artificial Neurons (IANs), where the blue squares contain the processing functions, the green squares contain the aggregation function, and the red squares contain the activation function. The output of the k -th IAN is the k -th concept. The concepts are passed to the Logic Explained Network (LEN), consisting of an Entropy-based layer, an activation, then a series of layers, and a concatenation. The output provided by LEN is both the prediction y_i and the truth table T_i for each of the L classes. The figure follows the color convention of both original papers [269] and [21].

$$y = f(G_{i=1}^N(h_i(x_i))) \quad (2.9)$$

The Inverted Artificial Neuron (IAN) [269] is a specialization of NEWRON in which first a non-linear function is applied to each individual input (with appropriate parameters) then an aggregation is performed by summation. We use the Heaviside function as activation function since we want the extracted concepts to fall in the set $\{0, 1\}$. Also, we use the Heaviside function as processing function because it is the only one of the three processing functions proposed in the original paper that provides crisp rules; the others can provide fuzzy rules. The equation (2.10) shows the structure of Heaviside-IAN with Heaviside activation function (that we denote with H):

$$y = H \left(w \left(\sum_{i=1}^N H(w_i(x_i - b_i)) - b \right) \right) \quad (2.10)$$

where w , b , w_i and b_i are trainable parameters.

This type of neuron has the peculiarity of being able to be translated into a rule of the form M -of- N , resulting from the inversion of the order of operations. This kind of rule is true if **at least** M of the N rules it is composed of are true. In the IAN case, each of the N rules refers to one of the N input features. The main limitation with this type of neuron arises when moving to multi-layer networks. In the paper it is shown how a single neuron is interpretable and how it is possible to propagate the rules through each layer to obtain a complete ruleset representing the entire network. However this ruleset is likely to be very large as no method has been proposed to regularize or aggregate the final ruleset.

Concept Representation Since the network is optimizable end-to-end, concepts are distilled in conjunction with network training: they do not need ground truth in the dataset since they are

learned directly from the data. Likewise, there is no need to have an additional loss for them.

In our case, the concepts are an aggregate of various features. In particular, they take the form of an M -of- N rule which, as mentioned above, is valid if **at least** M of the N rules it is composed of are true. Since the concepts are logical rules, they are easily human-interpretable. Moreover, the rules depend solely on the weights of the Inverted Artificial Neuron. Since, once trained, the weights of each Inverted Artificial Neuron are fixed, so are the rules that represent them and, consequently, so are the concepts they symbolize. For this very reason, the extracted concepts are interpretable and not explainable through individual samples, and they remain valid for whatever sample is given as input to the network, since the concept of the neuron is always the same and may only be active for one sample and not for another.

To give an example for tabular data, a concept, that is, the actual rule that IAN follows to give its output, might be represented by 2-of- $\{X1 > 3, X2 < 10, X3 \geq -0.4\}$. Since this writing means “at least 2 of the rules $X1 > 3, X2 < 10, X3 \geq -0.4$ are valid”, it can be unrolled and translated into the complete rule $(X1 > 3 \wedge X2 < 10) \vee (X1 > 3 \wedge X3 \geq -0.4) \vee (X2 < 10 \wedge X3 \geq -0.4)$. However, writing it as M -of- N is more convenient and can cover both the case of complete logical conjunction and disjunction, respectively when $M = N$ and $M = 1$.

The maximum number of concepts that can be found is given by the number K of IANs, i.e. the neurons in the first layer of the network. There is, however, the possibility that a concept is active or inactive for all samples and thus that neuron may be pruned later from the network, since it does not make sense to use it for ruleset generation. In addition, concepts that are discarded during ruleset construction can similarly be considered to be less important, but not completely irrelevant in the regular network’s prediction. The concept space is $C = \{0, 1\}^K$, where K is the number of concepts we are considering. For each sample the i -th entry of the vector $c \in C$, c_i , takes value 1 if the i -th concept is active/valid for that particular sample and 0 otherwise.

Logic Explained Network Logic Explained Networks (LEN) [55] are neural models that can generate simple concept-based logical explanations for network predictions. In particular, in this work we make use of the entropy-based version of LEN (E-LEN) [21] that makes use of an entropy-based criterion whose purpose is to identify the most relevant concepts. This layer encourages the neural model to choose a limited subset of input concepts, which thus leads to the creation of concise explanations of its predictions. Starting from the trained network and training instances, E-LEN is able to generate, for each class m , a truth table T_m representing the functioning of the network. This can be summarized, through various logical rule aggregation techniques, into sets of explanations in the form of a ruleset. The approach tries to make the created rule sets as faithful as possible to the network’s predictions, while maintaining low complexity.

Heaviside-IAN + LEN The combination of the two IAN and LEN models is able to create a network suitable for making predictions for tabular data and explaining them. An illustrative diagram is shown in Figure 2.5.

A first layer composed of K Heaviside IANs is able to aggregate the N input features into interpretable concepts. It should be noted that the benefit of IANs is precisely that the concepts are *interpretable*, not explainable in terms of single samples. In fact, other methods [98, 182] explain concepts as aspects (mostly visual) that synthesise the set of samples for which those concept are

valid. Once our network is trained, a rule can be automatically extracted from the neuron’s weight. This also favors the possibility to intervene on the concepts [153], perhaps to refine some splits on certain features or prune some concepts that are considered faulty.

After the input has been transformed into concepts, a LEN module predicts, for each class m , the class membership y_m as well as a truth table T_m . After training, the module can be used to provide an explanation at the level of each individual class based on the concepts extracted from the IAN. The truth tables $\{T_1, \dots, T_M\}$ are simplified using logical techniques to get concise FOL expressions for each class. These explanations can then be evaluated in terms of accuracy on the test dataset and fidelity. Fidelity measures how well the ruleset’s predictions represent the networks’ predictions.

It should also be noted that the two frameworks are optimized end-to-end simultaneously, thus not requiring a separate step to identify the concepts.

2.2.4 Experiments

Dataset	GB	NN	IAN-LEN (ours)	IAN-LEN Rules (ours)	Rules fidelity
adult	<i>87.29±0.03</i>	85.03±0.12	84.50±0.59	82.51±0.78	77.91±4.41
australian	<i>92.03±0.00</i>	86.52±0.58	86.81±1.16	83.48±1.25	91.45±3.76
breast-cancer-wisconsin	98.43±0.29	98.57±0.00	98.43±0.29	98.14±0.57	99.43±0.29
cleveland	57.38±0.00	59.02±3.28	54.43±2.41	55.74±1.80	56.72±6.85
diabetes	<i>78.57±0.00</i>	72.21±0.95	73.90±1.85	71.30±1.04	84.94±5.58
eye	<i>94.64±0.23</i>	53.97±0.00	56.38±0.87	57.18±1.62	71.72±18.49
german	<i>81.00±0.00</i>	76.10±0.58	75.60±1.46	74.20±0.40	93.30±9.25
haberman	<i>78.06±1.29</i>	75.81±0.00	74.52±1.21	74.19±1.44	94.19±6.66
heart	<i>87.04±0.00</i>	81.85±1.39	82.22±1.89	81.48±2.03	91.85±3.43
hepatitis	<i>87.10±0.00</i>	83.87±0.00	80.65±3.53	82.58±1.58	95.48±6.32
ionosphere	<i>95.21±0.69</i>	92.96±2.52	94.65±1.05	90.14±0.89	100.00±0.00
iris	<i>100.00±0.00</i>	96.67±0.00	98.67±1.63	99.33±1.33	98.00±2.67
monks-1	<i>100.00±0.00</i>	77.60±13.05	82.40±4.08	82.40±7.42	84.00±10.43
monks-2	95.88±1.44	88.24±7.44	100.00±0.00	100.00±0.00	98.82±1.44
monks-3	<i>100.00±0.00</i>	91.20±4.66	96.00±0.00	96.00±0.00	100.00±0.00
poker	<i>66.09±0.00</i>	49.84±0.00	54.31±0.70	51.34±1.24	80.71±8.70
sonar	<i>89.52±1.17</i>	83.33±4.52	82.86±3.16	80.95±3.98	90.00±3.50
thyroid	<i>99.87±0.00</i>	96.87±0.52	94.83±0.19	95.02±0.72	98.25±2.14
Average	<i>88.23±0.56</i>	80.54±4.00	81.73±1.86	80.89±2.28	89.27±6.85

Table 2.3: Accuracy results on test set of all datasets for non-interpretable classifiers, i.e. Gradient Boosting (GB), Neural Network (NN) and for our model (IAN-LEN). The accuracy of the rules extracted from IAN-LEN (IAN-LEN Rules) and the fidelity (Rules fidelity) of these to the original network’s prediction are also shown. Gradient Boosting result is put in *italics* if it is the best model for that dataset according to the average. In **bold**, the best model (except GB) according to the mean is highlighted.

Datasets For our experiments, we selected 18 tabular datasets for classification task from the UCI Machine Learning Repository [75].

We divided each dataset into train, validation, and test set. For datasets that are already provided separated into sets, we have kept the same separations, dividing the validation where missing, selected uniformly at random as 20% of the train. For the other cases, 20% of the samples were selected uniformly at random as the test set, and an additional 20% was separated to form the validation set.

Models For each of the datasets, we trained our model and 4 other classifiers. Two of these, Logistic Regression (LR) and Decision Tree (DT), are interpretable by design, and are therefore competitors to our model in the domain of explainability. The other two, traditional multilayer Neural Network (NN) and Gradient Boosted (GB) Decision Trees, are instead selected for their high performance.

For each classifier, we performed five training repetitions with different initialization seeds. For neural networks, this translates into a different initialization of the network’s initial weights and therefore presumably obtaining a different local minimum during gradient descent. For Decision Trees and Gradient Boosted Decision Trees the seed controls the random permutations of the features at each split. For Logistic Regression, on the other hand, it does not make sense to perform several repetitions.

Optimization For each classifier, we performed a random search to optimize its hyper-parameters. For a fairer comparison between the various models, the random search for each was performed at equal execution time (30 minutes per dataset). We note that this may put our model in disadvantage, since all the other traditional classifiers must probably have a greatly optimized code.

For further information on the experimental setup, please refer to the appendix.

2.2.5 Results

The results obtained show that our network performs equally, if not slightly better ($> 1\%$ increase) in accuracy than a traditional neural network. Still considering accuracy, we do much better than Logistic Regression ($> 6\%$ increase), while similarly to Decision Tree ($< 1\%$ difference). However, various factors such as optimization time, the number of combinations tested and the efficiency of the code itself are to be taken into account. Finally, we can claim to have obtained results in line with Rule Extraction methods in the literature. The comparison is a bit unfair (to our disadvantage) since some of the results for the other RE methods are missing information, such as number of hyperparameters combinations tested and sometimes even standard deviation.

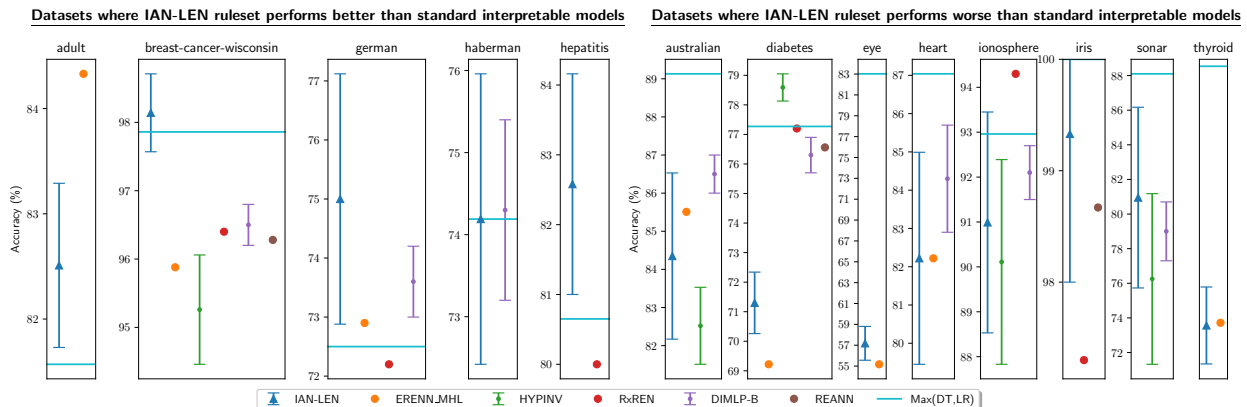


Figure 2.6: Accuracy mean and standard deviation for rules generated by our method (IAN-LEN) and by other rule extraction algorithms on datasets where we are able to outperform both Decision Tree and Logistic Regression. The maximum accuracy obtained by Decision Tree and Logistic Regression is also shows as a horizontal line.

Dataset	IAN-LEN Rules (ours)	DT	LR
adult	82.51±0.78	81.31±0.00	77.86±0.00
australian*	83.48±1.25	88.41±0.00	89.13±0.00
breast-cancer-wisconsin	98.14±0.57	97.14±0.00	97.86±0.00
cleveland	55.74±1.80	49.18±0.00	52.46±0.00
diabetes*	71.30±1.04	77.27±0.00	73.38±0.00
eye	57.18±1.62	83.02±0.09	56.86±0.03
german	74.20±0.40	72.50±0.84	70.50±0.00
haberman	74.19±1.44	72.58±3.38	74.19±0.00
heart*	81.48±2.03	87.04±0.00	85.19±0.00
hepatitis	82.58±1.58	80.65±5.77	74.19±0.00
ionosphere*	90.14±0.89	92.96±0.00	92.96±0.00
iris	99.33±1.33	96.67±0.00	100.00±0.00
monks-1	82.40±7.42	80.00±0.00	60.00±0.00
monks-2	100.00±0.00	91.76±2.20	55.88±0.00
monks-3	96.00±0.00	96.00±0.00	92.00±0.00
poker	51.34±1.24	47.50±0.70	13.43±5.03
sonar	80.95±3.98	76.67±1.78	88.10±0.00
thyroid	95.02±0.72	99.87±0.00	91.89±2.23
Average	80.89±2.28	81.7±1.73	74.77±1.3

Table 2.4: Accuracy results on test set of all datasets for interpretable classifiers, i.e. Decision Tree (DT) and Logistic Regression (LR), and for rules extracted from IAN-LEN (IAN-LEN Rules). In **bold**, the best model according to the average is highlighted. The asterisks mark the only datasets where our model performs worse than both LR and DT.

Comparison with non-interpretable models For each dataset, the classifier with the highest accuracy value in terms of mean minus standard deviation of accuracy was selected for each model type. This method to some extent ensures that more robust models are selected, compared to looking only at the mean.

Table 2.3 shows the accuracy obtained by Gradient Boosting, Neural Network, IAN-LEN and rules extracted from it for all the tested datasets. As expected, Gradient Boosting turns out to be the best model on almost all but two datasets. For this reason, to improve visualization, we decided to put it in italics instead of bold. In bold, we can instead see the comparison between a standard neural network and our model. We can see that the results obtained by the two models are comparable, with ours outperforming the other on 10 of the 18 datasets considered. In the last column we show the rules extracted from the respective IAN-LEN networks. We can see that in some cases the ruleset manages to perform even better than the network from which it is drawn, but the difference is not significant.

As for fidelity, on average the extracted rulesets are faithful to the network from which they are extracted by almost 90%. We should note, however, that the two datasets (cleveland and eye) where we perform worst (along with NN) suffer from very low fidelity. This certainly hampers the overall average. In fact, only on 5 dataset out of 18 dataset the fidelity is lower than 90%.

Comparison with interpretable models Table 2.4 shows the accuracy obtained by Decision Tree, Logistic Regression and the rules extracted from our IAN-LEN network for all the tested

datasets. The best result according to the accuracy mean is highlighted in bold. We can see that on 10 datasets over 18 the accuracy of our ruleset is equal or better than the accuracy obtained by both Decision Tree and Logistic Regression. Instead, only on 4 datasets we are outperformed by both LR and DT.

However, our network has greater potential than DT and LR. In fact, the experiments presented were carried out at equal execution time, for the fairest possible comparison. This means that for some datasets, our model manages to test only 5 combinations of hyper-parameters, while Decision Trees are capable of testing 100 and LR 1000. This possibly means that with more optimization time, our method can achieve even higher performance. Although it can be speculated that DT and LR could also benefit from more optimization time, we believe that given the high amount of combinations of hyper-parameters tested for them, it is unrealistically to increase greatly the performance of these two. While not featured in the present study, this analysis will certainly be pursued in the future. Another point to be made is that ours is a differentiable neural structure, which can also be combined with other differentiable models in a multi-model setting. Since IANLEN is a method for building architectures, we can imagine it being applicable to other types of networks as well, such as Convolutional or Transformers. Thereby, having been explored here for the first time, we believe it may have greater room for improvement and more versatile applicability than models such as DT and LR that have already been explored in detail.

Comparison with rule-extraction methods We can also analyze our performance in comparison with other rule extraction methods to see if they can beat interpretable models on datasets on which we cannot.

You might notice how the results for Figure 2.6 vary slightly from those shown in the tables 2.4 and 2.3. The reason for this is that for the figures we selected our rules that perform best on average, without considering standard deviation. We decided to do this since it is the same strategy adopted by the other papers on Rule Extraction with which we compare ourselves. We applied the same method to obtain the maximum performance of Logistic Regression and Decision Trees. We selected only those Rule Extraction models that on at least one dataset prove to be better than all others. In particular, we compare ourselves with ERENN_MHL [43], HYPINV [245], RxREN [11], DIMLP-B [31], REANN [140]. In Figure 2.6 we first see the performance in terms of mean and standard deviation of accuracy for five of the datasets where we are able to outperform interpretable models; we show our performance and that of five other Rule Extraction algorithms. We clearly outperform on two of five datasets (*breast-cancer-wisconsin* and *hepatitis*). On the *german* dataset, we can say that only DIMLP-B is comparable, although it scores below our average accuracy. On *haberman* we are virtually equal to DIMLP-B, but with little more standard deviation. The only dataset in which we are outperformed is *adult*. We have only the comparison with *RxREN*, of which we want to emphasize, however, that this value is provided without standard deviation, so it is difficult to assess its real superiority.

Looking in more detail at the datasets where we are unable to outperform Decision Tree or Logistic Regression, we can see that we are not the only ones having difficulty outperforming the classical interpretable models. The dataset *diabetes* is the only one where we are clearly inferior of one other method, HYPINV, which is also able to also outperform DT and LR. The same thing is done by RxREN on the *ionosphere* dataset, remembering, however, how without its standard

Dataset	Ruleset	Accuracy
Australian Credit Approval	7-of- $\{V_1 < 50.362, V_3 < 2.389, V_4 < 13.097, V_5 < 8.017,$ $V_7 < 0.514, V_9 < 53.475, V_{13} < 46658.008\} \Rightarrow \text{class 0}$ Default: class 1	85.51%
Breast Cancer Wisconsin	4-of- $\{V_0 \geq 5.048, V_1 \geq 2.1, V_2 \geq 2.045, V_3 \geq 6.918, V_4 \geq 1.96,$ $V_5 \geq 3.197, V_6 \geq 2.96, V_7 \geq 8.583, V_8 \geq 3.026\} \Rightarrow \text{class 1}$ Default: class 0	98.57%
Iris	2-of- $\{V_0 \geq 4.361, V_2 \geq 4.997, V_3 \geq 1.791\} \vee$ 2-of- $\{V_0 \geq 6.14, V_1 \leq 3.073, V_2 \geq 4.952, V_3 \geq 1.68\} \Rightarrow \text{class 2}$ 2-of- $\{V_0 \geq 5.68, V_1 \leq 2.999, V_2 \geq 3.055, V_3 \geq 0.857\} \Rightarrow \text{class 1}$ Default: class 0	100.00%

Table 2.5: Rulesets generated by IAN-LEN on some of the tested datasets and the corresponding accuracy on each test set. M -of- $\{N$ rules $\}$ indicates an active rule if *at least* M of the N rules are active. V_i denotes feature i for the corresponding dataset, with V_0 being the first feature.

deviation this result is hardly reliable. On this and 2 other datasets (*australian* and *heart*) we are comparable to the other models, while on *eye* we are superior to ERENN_MHL. On the remaining 3 datasets (*iris*, *sonar* and *thyroid*), we can consider ourselves almost superior as the other results rank below or similar to our average.

We can summarize that even where we lose the comparison to classical interpretable models, the same happens to the state of the art for Rule Extraction. Moreover, we still come out similar or better than the state of the art for rule extraction by neural network. In addition, it is important to emphasize that for the other RE methods, we have neither the execution times nor the type of machine they ran on, which makes it impossible for us to make a completely fair comparison with them.

Rulesets examples From the five replications that generated the results in Table 2.4, we extracted the fold that obtained the best result to show the extracted rulesets. Table 2.5 shows the aforementioned rulesets extracted by our method for some of the tested datasets, along with their performance obtained on the test set. As already specified, by the term M -of- N rules we mean an active rule only if **at least** M of the N rules are active.

2.2.6 Conclusions

In this paper, we propose neural network models for classification that is able to generate explanations for the network’s predictions in terms of simple first-order logic rulesets. These rulesets involve concepts that are automatically distilled by our network. To the best of our knowledge, this is a novelty in the field of Concept-based Explainability for tabular data. Moreover, the concepts are interpretable being composed as M -of- N rules on the features of the dataset.

Our model is obtained by concatenating a layer of Inverted Artificial Neurons and a Logic Explained Network. We prove that we have overcome the limitations of the previous two works. In fact, we are able to generate interpretable concepts, overcoming LEN’s inability to produce concepts. They can also produce easy-to-read rule sets, unlike IAN, which can produce excessively large rule

sets that are therefore difficult to interpret.

We have shown, through experiments on tabular datasets, that our model achieves on average slightly better results (in terms of accuracy) than neural network and results similar to Decision Trees with equal optimization time for hyperparameter search. DT can rely on a larger hyperparameter optimization, taking advantage of a reduced execution time per combination, while our network, for the same amount of time, is able to explore fewer combinations. This suggests that there are still many opportunities for improvement for our neural structure. We showed that our results are in line with those of other Rule Extraction methods and we proved experimentally that our model is able to produce rules explaining its predictions with a fidelity of more than 89%.

A current limitation is being able to apply this type of network only to tabular data, due both to the difficulty of generating concepts that can be easily interpreted for the other kinds of data and to the restriction given by IAN. Future work may therefore focus on adapting IAN to the case of time series, images, graphs or text, so that interpretable concepts can be generated for these other types of data as well.

2.3 Concept Distillation in Graph Neural Networks

The opaque reasoning of Graph Neural Networks induces a lack of human trust. Existing graph network explainers attempt to address this issue by providing post-hoc explanations, however, they fail to make the model itself more interpretable. To fill this gap, we introduce the Concept Distillation Module, the first differentiable concept-distillation approach for graph networks. The proposed approach is a layer that can be plugged into any graph network to make it explainable by design, by first distilling graph concepts from the latent space and then using these to solve the task. Our results demonstrate that this approach allows graph networks to: (i) attain model accuracy comparable with their equivalent vanilla versions, (ii) distill meaningful concepts achieving 4.8% higher concept completeness and 36.5% lower purity scores on average, (iii) provide high-quality concept-based logic explanations for their prediction, and (iv) support effective interventions at test time: these can increase human trust as well as improve model performance.

2.3.1 Introduction

Human trust in machine learning requires high task performance alongside interpretable decision making [261]. For this reason, the opaqueness of Graph Neural Networks (GNNs, [249])—despite their state-of-the-art performance [25, 66, 212, 281]—raises ethical [76, 174] and legal [80, 307] concerns. As their practical deployment is now under question, interpreting GNN reasoning has become a major concern in the field [241, 336].

Early explainability methods for GNNs produce local, low-level post-hoc explanations [179, 306, 336], which exhibit the same unreliability as analogous methods for convolutional networks [3, 97, 151]. In contrast, concept-based explainability overcomes the brittleness of low-level explanations by providing robust global explanations in form of human-understandable concepts [150], i.e., interpretable high-level units of information [98, 154]. In relational learning, the Graph Concept Explainer (GCExplainer, [182]) pioneered concept-based explainability for GNNs by extracting global subgraph-based concepts, such as a “house-shaped” structure, from the latent space of a trained model. This way users can check whether the extracted concepts are meaningful [98], whether they are coherent across samples [182], and whether they contain sufficient information to solve a target task [334]. However, as any post-hoc approach, GCExplainer does not encourage the GNN to make interpretable predictions using the extracted concepts [241]. At best, post-hoc techniques can correctly describe what models learn [241], but they cannot make the GNN itself more interpretable. Therefore, the opaque reasoning of GNNs remains an open problem.

To fill this knowledge gap, we propose the Concept Distillation Module (CDM, Figure 2.7), the first concept-based end-to-end differentiable approach which makes graph networks **explainable by design**. It achieves this by first distilling a set of concepts present in the GNN’s latent space and then using these to solve the task at hand. Our module can be introduced in any GNN

architecture. We will refer to the resulting family of architectures as ‘‘Concept Graph Networks’’, or CGNs. We experimentally show that CGNs: (i) attain better or competitive task accuracy w.r.t. their equivalent vanilla GNN, (ii) distill coherent human-understandable concepts from the latent space and obtain high scores in all the key concept-based explainability metrics, i.e., purity and completeness, (iii) can provide simple and accurate logic explanations based on discovered concepts, (iv) allow effective interventions at concept level: these can increase human trust and significantly improve model performance.

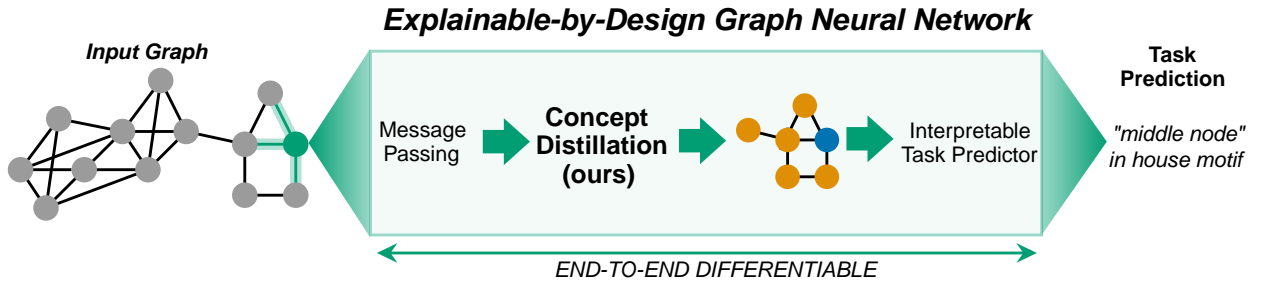


Figure 2.7: The proposed Concept Distillation Module makes graph networks explainable-by-design by discovering a set of concepts and using these to solve the task with an interpretable classifier.

2.3.2 Background and Related Work

Graph Neural Networks

Graph Neural Networks (GNNs, [249]) are differentiable models designed to process relational data in the form of graphs. A graph can be defined as a tuple $G = (V, E)$ which comprises nodes $V = \{1, \dots, n\}$, the entities of a domain, and edges $E \subseteq \{1, \dots, n\} \times \{1, \dots, n\}$, the relations between pairs of nodes. Nodes (or edges) can be endowed with features $\mathbf{x}_i \in \mathbb{R}^d$, representing d characteristics of each entity (or relation), and with l ground truth task labels $y_i \in Y \subseteq \{0, 1\}^l$. A typical GNN g learns a set of node embeddings \mathbf{h}_i with a scheme known as message passing [101]. Specifically, message passing aggregates for each node $i \in V$ local information shared by its neighboring nodes $N_i = \{k : (k, i) \in E\}$:

$$\mathbf{h}_i = \sum_{k \in N_i} g(\mathbf{m}_{ik}, \mathbf{x}_i) \quad \mathbf{m}_{ik} = \phi(\mathbf{x}_i, \mathbf{x}_k) \quad (2.11)$$

where \mathbf{m}_{ik} is the aggregate of the feature vectors \mathbf{x}_i and \mathbf{x}_k of nodes i and k , respectively, computed using a permutation invariant function ϕ . A readout function $f : H \rightarrow Y$ then processes the node embeddings to predict node labels \hat{y}_i . GNNs are trained via stochastic gradient descent minimizing the cross entropy loss between \hat{y}_i and ground-truth y_i .

Graph Concept Explainer

The Graph Concept Explainer (GCExplainer, [182]) is the first concept-based approach for interpreting GNNs. Following methods successfully applied in vision [98], GCExplainer is an unsupervised approach for post-hoc discovery of global concepts. It achieves this by applying k-Means clustering [88] on the node embeddings \mathbf{h}_i of a trained GNN. [182] argue that each of the k clusters possibly represents a learnt concept according to human perception, as already suggested by [343]

and [334]. Using this clustering, GCExplainer then assigns a concept label $\hat{c}_i \in \hat{C} \subseteq \{0, 1\}^k$ to each sample. Finally, it represents each concept using the five samples closest to each cluster centroid, where each sample is visualized as a subgraph with the corresponding node and its p -hop neighbors. This visualization technique is aligned with the reasoning of GNNs, as it takes into account how the information flows via message passing. For example, after three layers of message passing, each node can receive at most information from its 3-hop neighbors.

Trust through Concepts and Interventions

Predicting tasks as a function of learnt concepts makes the decision process of deep learning models more interpretable [154, 261]. In fact, learning intermediate concepts allows models to provide concept-based explanations for their predictions [97] which can take the form of simple logic statements [55]. In addition, [154] show how learning intermediate concepts allows human experts to rectify mispredicted concepts through test-time interventions, improving model performance and engendering human trust [261].

Related Work

State of the Art Graph Explainers

GNNEExplainer [336] represents the first seminal work on GNN explainability. It maximizes the mutual information between GNN predictions and the distribution of possible subgraphs for explanations. By focusing on individual predictions, the method is limited to explaining one instance at a time corresponding to a localized view of the data distribution. To get a full picture, [336] suggest to perform subgraph matching on a substantial number of instances. However, this is not scalable as subgraph matching is NP-Hard [336]. In an attempt to alleviate this issue, the Parameterised Graph Explainer (PGExplainer, [179]) and the Probabilistic Graphical Model Explainer (PGM-Explainer, [306]) parametrize the process of generating explanations using deep neural networks to provide multi-instance explanations. However, all these methods remain fundamentally limited in their locality as they cannot explain a class of samples in its entirety. In contrast, GCExplainer [182] fills the gap of global explainability for GNNs using concept-based explanations. Similarly, the concurrent work of [13] propose a global and differentiable explainer for GNNs, however, it is post-hoc as it is applied to a trained GNN. While these existing techniques begin to address the lack of insight into the computations of GNNs, they are all post-hoc methods whose goal is to explain a trained GNN, not to make it more interpretable. The proposed method instead aims at filling this knowledge gap by making GNNs explainable by design. The Prototype Graph Neural Network (ProtGNN, [349]) learns prototypical graph patterns that can be used for classification. While ProtGNN has a similar aim of producing an interpretable model as opposed to post-hoc explanations, it is not concept-based. In contrast our method is concept-based and increases the interpretability of the model.

Concept-based Explainability

From a broader perspective, our work borrows ideas from supervised and unsupervised concept-based methods. These methods have been explored in various ways for other neural networks, such as

convolutional neural networks [5, 49, 98, 145, 154] and recurrent architectures [144]. From supervised concept-based methods, our approach inherits the ability to perform effective human interventions at concept level extending Concept Bottleneck Models [154] to graphs. As for unsupervised methods, our approach mainly draws from the Automatic Concept Extraction algorithm (ACE, [97]). The algorithm extracts visual concepts by performing k-Means clustering [88] on image segments in the activation space of a convolutional network. The ACE approach is based on the observation that the learned activation space is similar to human perceptual judgement [343]. This was the main motivation behind GCEExplainer, as well as our approach. However, in contrast to ACE and GCEExplainer, we embed the clustering step within the network architecture, making GNNs explainable by design. While our work proposes clustering within the GNN, similar to [146], we focus on the interpretability aspect achieved via this clustering.

2.3.3 Concept Distillation in Graph Neural Networks

We propose the *Concept Distillation Module* (CDM), a differentiable approach which makes GNNs explainable by design. In fact, CDM empowers GNNs through a more interpretable decision making process. It achieves this by first distilling a set of concepts present in the latent space and then using these to solve a classification task. Note that these concepts are inherent to the GNN, and CDM only filters them from the latent space. Thus, they do not improve classification accuracy but make classification more interpretable. Our approach can be integrated in any GNN architecture. We will refer to the resulting family of architectures as Concept Graph Networks (CGNs). As in GCEExplainer, humans can visualize CGN concepts to check whether they are meaningful and coherent. Yet, in contrast to GCEExplainer, CGNs allow effective interventions at concept level, allowing human experts to improve model performance. CDM integrates a differentiable concept distillation layer to extract node-level and graph-level concepts with an interpretable task predictor providing logic-based explanations.

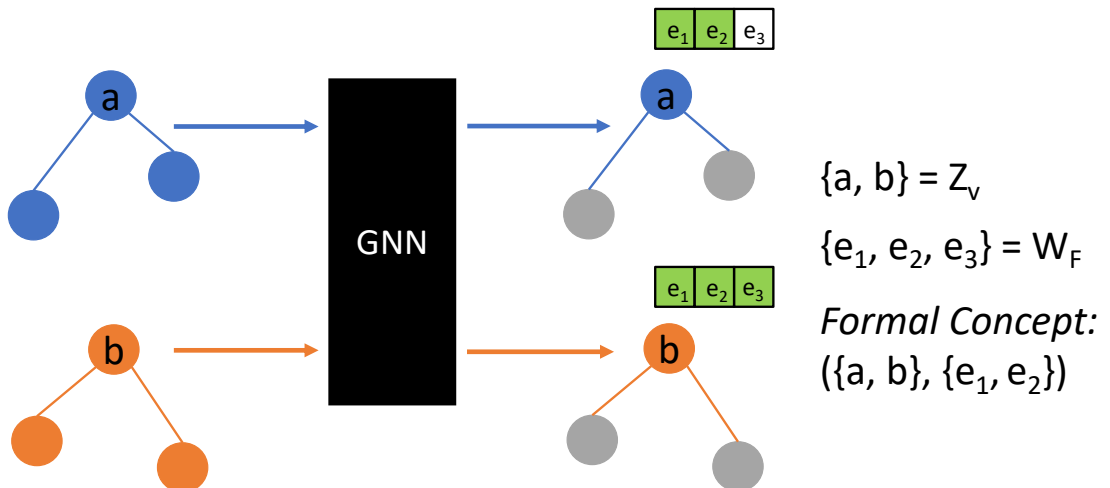


Figure 2.8: Formal Concepts in Graph Neural Networks.

Formal Concepts in Graph Neural Networks

Our work relies on concepts being inherently present in the latent space of GNNs, which can be shown using formal concept analysis based on the theory of complete lattices [93]. We provide a short formal definition of concepts, but refer the reader to [93] for a complete overview. Let us first define a formal context K as $K := (Z, W, I)$, where I is the relation between the objects Z and the attributes W . We denote the relation I of an object z with an attribute w as zIw . For a subset of objects $A \subseteq Z$, we can define the set of associated attributes $A' := \{w \in W | zIw \forall z \in A\}$. In the same manner, we can define a subset of attributes $B \subseteq W$ via a subset of objects $B' := \{z \in Z | zIw \forall w \in B\}$. This allows to represent a formal concept of the context K as a pair (A, B) , where $A' = B$ and $B' = A$, as the relation I allows us to map from objects to attributes and vice versa. This also achieves a hierarchical ordering, where an ordered set of all concepts in a context is $\underline{\mathcal{B}}(Z, W, I)$, the concept lattice of the context.

In the setting of GNNs, we can apply the theory of concept lattices in the following way. Let the nodes of a graph or set of graphs be our set of objects Z_v . Let the feature vectors found when aggregating across a node’s neighborhood be our set of attributes W_f . This set of attributes is dependent on the number of GNN layers, which leads to more distant neighbors being taken into account. Let the relation I_e associate our objects Z_v with the feature attributes W_f . Given this definition, it becomes evident that concepts are inherently present in the latent space of GNNs. Let us illustrate this with an example, visualized in Figure 2.8. Assume a GNN with a single layer, which means that W_f will be the feature vectors found when aggregating a node’s feature vector with those of all of its neighbors. Each node, representative of an object in Z_v (a and b in Figure 2.8), is associated with a feature vector, representative of an attribute in W_f (e_1 , e_2 and e_3 in Figure 2.8). Then nodes with similar features and neighborhoods will map to the same set of attributes and can be formally represented as a concept $((\{a, b\}, \{e_1, e_2\}))$ in Figure 2.8). Moreover, this implies that nodes forming a concept will be clustered in the activation space, which we exploit in the concept distillation step of CDM.

Concept Distillation

The first CDM step consists of extracting node-level clusters corresponding to concepts from the GNN’s latent space. This is based on the observation that the arrangement of the activation space shows similarities to human perceptual judgement [343], as shown by GCExplainer [182] for GNNs, and the application of concept lattice theory [93]. However, in contrast to GCExplainer, in CDM this step is differentiable and integrated in the network architecture, allowing gradients to optimize clusters in GNN embeddings. Specifically, we implement this differentiable clustering using a normalized softmax activation on the node-level embeddings \mathbf{h}_i , associating each node with one cluster/concept. This operation returns for each node a fuzzy encoding $\mathbf{q}_i \in [0, 1]^s$:

$$\tilde{\mathbf{q}}_i = \frac{\exp(\mathbf{h}_i)}{\sum_{u=1}^s \exp(\mathbf{h}_{iu})}, \quad \mathbf{q}_i = \frac{\tilde{\mathbf{q}}_i}{\max_i \tilde{\mathbf{q}}_i + \epsilon} \quad (2.12)$$

where s is the size of the encoding vector. CDM then clusters nodes considering the similarity of their fuzzy encodings \mathbf{q}_i . Specifically, CDM groups the samples together depending on their

Booleanized encoding $\mathbf{r}_i \in \{0, 1\}^s$:

$$\mathbf{r}_{iu} = \begin{cases} 1 & \text{if } \mathbf{q}_{iu} \geq \tau \\ 0 & \text{otherwise} \end{cases} \quad (2.13)$$

where $\tau \in [0, 1]$ is conventionally set to 0.5. In particular, two samples a and b belong to the same cluster if and only if their encodings \mathbf{r}_a and \mathbf{r}_b match. For example, consider the two node embeddings $\mathbf{h}_a = [-1.2, 2.3]$ and $\mathbf{h}_b = [2.2, 1.8]$. For these inputs, the normalized softmax would return the fuzzy encodings $\mathbf{q}_a = [0.029, 0.971]$ and $\mathbf{q}_b = [0.599, 0.401]$, respectively. As their Booleanizations $\mathbf{r}_a = [0, 1]$ and $\mathbf{r}_b = [1, 0]$ do not match, we can then conclude that the two nodes belong to different clusters. Notice how our concept encoding is theoretically justified via concept lattices and is highly efficient, as it allows to learn up to 2^s different concepts on GNN embeddings \mathbf{h}_i of size s . This way the GNN can dynamically find the optimal number of concepts/clusters, thus relieving users from this burden. In fact, users just need to choose an upper bound to the number of concepts s rather than an exact value, as when using k-Means like in GCExplainer. In order to account for graph classification, the concept encodings for a graph are pooled before being passed to the interpretable model predicting the task, as explained in the next paragraph.

Interpretable Predictions

The second CDM step consists of using the distilled concepts to make interpretable predictions for downstream tasks. In particular, the presence of concepts enables pairing GNNs with existing concept-based methods which are explainable by design, such as Logic Explained Networks (LENs, [55]). LENs are neural models providing simple concept-based logic explanations for their predictions. Specifically, LENs can provide class-level explanations which makes our approach the first at providing unique global explanations for GNNs. Given the formal definition of concepts, they naturally lend themselves as propositions for the logic explanations produced by LENs. CDM uses a LEN as the readout function f for the classification, applying it on top of concept representations \mathbf{q}_i . For graph classification tasks, the input data is composed of a set of t graphs $G^j \in \{(V^j, E^j)\}_{j=1}^t$, where each graph is associated with a task label $y^j \in Y$. In this setting, GNN-based models predict a single label for each graph G^j by pooling its node-level encodings \mathbf{q}_i^j to aggregate over multiple concepts:

$$\hat{y}_i = \text{LEN}_{\text{node}}(\mathbf{q}_i), \quad \hat{y}^j = \text{LEN}_{\text{graph}}\left(\frac{1}{n_j} \sum_{i=1}^{n_j} \mathbf{q}_i^j\right) \quad (2.14)$$

where n_j is the number of nodes associated with graph j . In our implementation, we use the entropy-based layer to implement LENs [20]) as it can provide high classification accuracy with high-quality logic explanations. This entropy-based layer implements a sparse attention layer designed to work on top of concept activations. The attention mechanism allows the model to focus on a small subset of concepts to solve each task. It also introduces a parsimony principle in the architecture corresponding to an intuitive human cognitive bias [190]. This parsimony principle allows the extraction of simple logic explanations from the network, thus making these models explainable by design.

Concept-based and logic-based Explanations

The proposed method provides two types of explanations: concept-based and logic-based explanations. Global concept-based explanations can be extracted in a similar manner as in GCExplainer: a concept for a node or graph is extracted by finding the cluster with which a node’s embedding is associated, and visualising the samples closest to the cluster centroid. The logic-based formula provided per class broadens the explanation scope, as it indicates which neurons of the concept encoding \mathbf{q}_i are activated and representative of a class. This provides a more comprehensive explanation since a class can be associated with multiple concepts.

Concept Interventions

As in Concept Bottleneck Models [154], our approach supports human interaction at concept level. In fact, in contrast to existing post-hoc methods, an explainable-by-design approach creates an explicit concept layer which can positively react to test-time human interventions. For instance, consider a misclassified node with concept encoding $\mathbf{q}_a = [0.21, 0.93]$. Assume that the vast majority of nodes with the binary encoding $\mathbf{r}_{\text{grid_node}} = [0, 1]$ are nodes of a grid-like structure, which allows a human to label this cluster as “grid nodes”. Now, a human expert can inspect the neighborhood of the misclassified node and realize that this node belongs to a circle-like structure and not to a grid structure. As the binary encoding for the concept “circle nodes” is $\mathbf{r}_{\text{circle_node}} = [1, 1]$, the user can easily apply an intervention to correct the misclassified concept by changing its encoding to $\mathbf{q}_a := [1, 1]$. Such an update allows the interpretable readout function to act on information related to the corrected concept, thus improving the original model prediction.

2.3.4 Experiments

In our experiments we focus on the following research questions:

- **Task Accuracy and Completeness** — What is the impact of our approach on the generalization error of a GNN? Is the identified concept set complete w.r.t. the task?
- **Concept Interpretability** — Are the unsupervised concepts identified by our model meaningful? Do they match ground truths or human expectations?
- **Explanation Performance** — Are concepts pure and coherent? Are the logic explanations provided accurate and simple enough to be interpretable?

With these questions in mind, we hypothesize that our approach can: (i) obtain similar task accuracy w.r.t. a standard GNN; (ii) extract the ground truth graph concepts aligned with human expectations, and (iii) identify pure concepts as well as simple and accurate logic explanations.

Metrics

In our evaluation, we measure model performance and interpretability based on five metrics. We measure model performance via *classification accuracy* to compare the generalization error of CGNs w.r.t. their equivalent vanilla GNNs. To evaluate model interpretability, we compute *concept completeness* [334], which assesses whether the concepts discovered are sufficient to describe the

downstream task. Following [334], we use a decision tree [35] to predict the task labels given the concept encoding associated with each input instance. We also examine concept coherence via *concept purity* [182]. Following [182], we measure concept purity by considering the graph edit distance of samples’ neighborhoods within each cluster/concept. Having checked concept quality, we evaluate logic explanations in terms of their accuracy and complexity. We calculate the *accuracy of logic explanations* using the learnt logic formulas for classifying test samples based on their concept encoding as done by [55]. This mirrors the computation of concept completeness, however, instead of a decision tree, we use the learnt logic formulas for classification. Lastly, we evaluate the *complexity of logic explanations* by measuring the number of terms in logic rules [55]. We compute all metrics on test sets across five random weight initializations and report their means and 95% confidence intervals using the t-distribution. We do not measure classical explainability metrics, such as sensitivity and sparsity [354], as they apply to explainers of models rather than explainable-by-design networks themselves.

Datasets

We perform the experiments on the same set of datasets as the Graph Neural Network Explainer (GNNE explainer, [336]), as subsequent research establishes them as benchmarks [179, 182, 306].

Node Classification

We use five synthetic node classification datasets put forward by [336], which have a ground truth motif encoded. A ground truth motif is a subgraph, which a successful explainability technique should recognize. The first dataset is BA-Shapes, which consists of a single graph where the base structure is a Barabási-Albert (BA) graph [18] of width 300, which has 80 house motifs and 70 random edges attached to it. The dataset has 4 classes, with the goal of discriminating between a node being part of the base graph or the top, middle or bottom of a house structure. The second dataset is BA-Community, generated by the union of two BA-Shapes graphs. Here, the task is to classify a node into 8 classes, which represent graph membership and the structural role of the node as in BA-Shapes. The third dataset is BA-Grid, which is a BA graph of width 300 with 80 3-by-3 grids attached to it. The goal is to classify whether a node is part of the base graph of a grid structure. The fourth dataset is Tree-Cycles, formed by a binary tree of depth 8 with 60 cycle structures of 6 nodes attached to it. The task is to classify between a node belonging to the tree or cycle structure. Lastly, the fifth dataset is Tree-grid, which consists of a binary tree of depth 8, which has 80 3-by-3 grid structures attached. The classification task is the same, asking to discriminate between a node being part of the tree or grid structure.

Graph Classification

We also include two real-world datasets to evaluate model performance on less structured data and on graph classification tasks. The first dataset is Mutagenicity [195], which is a collection of graphs representing mutagenic and non-mutagenic molecules. The task is to identify a molecule as mutagenic or non-mutagenic. The second dataset is Reddit-Binary [195], which is a collection of graphs representing Reddit discussion threads where nodes represent users and edges represent interactions. A challenge in evaluating these datasets is that there are no ground truth motifs.

However, [336] suggests the ring structure and nitrogen dioxide compound in Mutagenicity, and the star-like structure in Reddit-Binary as desirable motifs to be recovered. Figure 2.6 provides further statistics on the datasets, such as the graph size and number of classes.

Table 2.6: An overview of key markers of the datasets.

Dataset	Classification Problem	Number of Graphs	Graph Size	Number of Features	Number of Classes
BA-Shapes	Node	1	700	1	4
BA-Community	Node	1	1400	1	8
BA-Grid	Node	1	1020	1	2
Tree-Cycles	Node	1	871	1	2
Tree-Grid	Node	1	1231	1	2
Mutagenicity	Graph	4337	30.32 (on average)	14	2
Reddit-Binary	Graph	2000	429.63 (on average)	1	2

Baselines and Setup

To address our research questions, we compare our approach against an equivalent convolutional vanilla GNN explained by GCExplainer. Specifically, we perform a quantitative evaluation by comparing the averages and confidence intervals obtained for each metric. We perform a qualitative evaluation by comparing the concepts extracted and whether they recover the desired motifs. Notice that we do not focus on other post-hoc explainability methods, such as GNExplainer [336], PGExplainer [179] or PGM-Explainer [306], as to the best of our knowledge GCExplainer is the only explainability method providing global concept-based explanations for GNNs. We do later provide a brief comparison of the proposed method with GNExplainer and ProtGNN [349] for completeness.

For each of the datasets, we use 80% of the data for training and 20% for testing. The examples in each split vary across seeds due to the different random initialization. We select the models' hyperparameters, such as the number of hidden units and learning rate, using a grid search. To ensure fairness in our results, we use the same architecture capacity and hyperparameters for our model as well as for its vanilla counterpart. We initialize the hyperparameters of GCExplainer to the values determined experimentally by [182].

2.3.5 Results

Concept Graph Networks are as accurate as vanilla GNNs (Table 2.7)

Our results show that CDM allows GNNs to achieve better or comparable task accuracy w.r.t. equivalent GNN architectures. Specifically, our approach outperforms vanilla GNNs on the Tree-Cycle dataset, having a higher test accuracy (plus $\sim 8\%$ on average) and less variance across different parameter initializations. We hypothesize that this effect is due to more stable and pure concepts being learnt thanks to CDM, as we will see later when discussing the concept purity scores. We do not observe any significant negative effect of using CDM on the generalization error of GNNs.

Table 2.7: Model accuracy for the Concept-based Graph Network (CGN) and an equivalent vanilla GNN.

	Model Accuracy (%)	
	CGN	Vanilla GNN
BA-Shapes	98.11 (97.04, 99.18)	98.02 (96.40, 99.65)
BA-Community	85.67 (81.38, 89.95)	87.50 (85.56, 89.45)
BA-Grid	99.51 (98.75, 100.00)	99.71 (99.38, 100.00)
Tree-Cycle	94.97 (92.50, 97.44)	86.26 (58.58, 100.00)
Tree-Grid	95.17 (93.59, 96.75)	94.54 (93.61, 95.46)
Mutagenicity	82.40 (81.31, 83.48)	82.35 (81.64, 83.06)
Reddit-Binary	90.55 (87.95, 93.15)	91.20 (88.82, 93.58)

The Concept Distillation Module discovers complete concepts (Table 2.8)

Our experiments show that overall CDM discovers a more complete set of concepts w.r.t. the concept set extracted by GCExplainer on equivalent GNN architectures. This is particularly emphasized in the Tree-Grid, BA-Shapes and BA-Community datasets, where CDM significantly outperforms GCExplainer by up to $\sim 13\%$. For the other datasets, the proposed approach matches the concept completeness scores of GCExplainer. The completeness scores on the BA-Grid and Mutagenicity datasets are only slightly lower, however, within the margins of the confidence interval. In absolute terms, CDM discovers highly complete sets of concepts with completeness scores close to the model accuracy for the synthetic datasets.

Table 2.8: Concept completeness and purity for the Concept-based Graph Network (CGN) and an equivalent vanilla GNN.

	Concept Completeness (%)		Concept Purity	
	CGN	Vanilla GNN	CGN	Vanilla GNN
BA-Shapes	98.11 (96.85, 99.36)	93.69 (86.21, 100.00)	0.00 (0.00, 0.00)	0.00 (0.00, 0.00)
BA-Community	83.10 (78.90, 87.29)	75.74 (72.85, 78.64)	1.70 (0.43, 3.83)	1.60 (0.49, 2.71)
BA-Grid	99.61 (98.80, 100.00)	99.71 (99.38, 100.00)	0.20 (0.00, 0.76)	2.40 (0.00, 6.48)
Tree-Cycle	91.98 (83.71, 100.00)	91.16 (84.47, 97.86)	0.00 (0.00, 0.00)	0.60 (0.00, 2.27)
Tree-Grid	91.37 (84.58, 98.16)	78.48 (76.17, 80.79)	0.00 (0.00, 0.00)	0.00 (0.00, 0.00)
Mutagenicity	63.40 (58.84, 67.96)	63.95 (60.14, 67.77)	1.00 (0.00, 3.78)	0.60 (0.00, 2.27)
Reddit-Binary	75.91 (61.16, 90.66)	73.10 (58.44, 87.75)	0.40 (0.00, 1.51)	0.00 (0.00, 0.00)

Ablation Study on the Size of the Concept Embedding Size

In order to verify the effectiveness and robustness of our approach, we perform an ablation study on the concept embedding size s . More specifically, we control the upper bound of the size of the concept lattice while observing the concept completeness score for different values of s . We conduct this ablation study on the BA-Shapes dataset using the values 2, 6, 10, 12 and 14 for s . Table 2.9 summarises the results obtained. We observe that the completeness score is stable for different values of $s \sim 10$.

The Concept Distillation Module identifies meaningful concepts (Table 2.10)

CDM discovers high-quality concepts, which are meaningful to humans. Similar to GCExplainer, our results demonstrate that CDM can discover concepts corresponding to the ground truth motifs embedded in the toy datasets. For example, our approach recovers the “house motif” in BA-Shapes.

Table 2.9: The concept completeness score for a CDM trained on the BA-Shapes dataset for different concept embedding sizes.

Concept Embedding Size s	Concept Completeness (%)
2	70.35 (44.26, 96.44)
6	94.09 (89.83, 98.35)
10	98.11 (96.85, 99.36)
12	97.57 (96.48, 98.60)
14	97.58 (95.48, 99.67)

Table 2.10: The Concept Distillation Module detects meaningful concepts matching the expected ground truth. Blue nodes are the instances being explained, while orange nodes represent their p -hop neighbors. Similar motifs are identified by GCEExplainer.

	BA-Shapes	BA-Grid	Tree-Grid	Tree-Cycle	BA-Community	Mutagenicity	Reddit-Binary
Ground Truth							
Extracted Concept							

Table 2.11: The Concept Distillation Module detects concepts more fine-grained than the simple ground truth motif encoded, as well as rare motifs. Blue nodes are the instances being explained, while orange nodes represent their p -hop neighbors. Notably, GCEExplainer gives no indication of rare concepts.

Ground Truth	Fine-Grained Concepts	Rare Concepts

Moreover, CDM proposes plausible concepts for the real-world datasets where ground truth motifs are lacking. In this case, the extracted concepts match the desirable motifs suggested by [336], corresponding to ring structures and the nitrogen dioxide compound in Mutagenicity, and a star-like structure in Reddit-Binary. As we use the same visualization technique as GCEExplainer the merit of our contribution lies in the discovery of a more descriptive set of concepts, which includes rare and fine-grained concepts.

The Concept Distillation Module identifies rare and fine-grained concepts (Table 2.11)

CDM discovers more fine-grained concepts than just the “house motif” suggested by GNNExplainer, as it can differentiate whether a middle or bottom node is on the far or near side of the edge attaching to the BA graph. This matches the quality of concepts extracted by GCEExplainer. In contrast to GCEExplainer, CDM also identifies rare concepts. Rare motifs are present in toy datasets through the insertion of random edges. As the proposed approach can find the optimal number of clusters/concepts dynamically, clusters of a very small size possibly represent rare motifs. To check the presence of rare concepts, we visualize the p -hop neighbors of nodes found in small clusters. For example, CDM identifies a rare concept represented as a “house” structure attached to the BA graph via the top node of the house in the BA-Shapes dataset. This represents a rare concept as it is generated by the insertion of a random edge. We confirm this observations on other toy datasets,

such as BA-Community and Tree-Cycle, where motifs with random edges are clearly identified. We have not identified rare concepts in BA-Grid or Tree-Grid, which may be attributed to the random edges being distributed within the base graph, which has a less definite structure. Due to the lack of expert knowledge, we cannot confirm whether the rare motifs found in Mutagenicity and Reddit-Binary align with human expectations.

The Concept Distillation Module identifies pure concepts (Table 2.8)

CDM discovers high-quality concepts, which are coherent across samples, as measured by concept purity. Our approach discovers concepts with nearly optimal purity scores on toy datasets, with a graph edit distance close to zero. For these datasets, CDM provides either better or comparable purity scores when compared to GCExplainer. CDM provides slightly worse purity scores in both the Mutagenicity and Reddit-Binary datasets. However, also in this case the absolute purity of CDM is almost optimal.

The Concept Distillation Module provides accurate logic explanations (Table 2.12, Table 2.13)

LEN allows CDM to provide simple and accurate logic explanations for task predictions. The accuracy of the logic explanations extracted reaches at least 90% for the BA-Shapes, BA-Grid and Tree-Cycle datasets, indicating that CDM derives a precise description of the model decision process. Relating the accuracy of explanations back to the model accuracy, we observe that the explanation accuracy is bounded by task performance, as already noticed by [55]. This explains the slightly lower logic explanation accuracy on the real-world datasets, which can be ascribed to the absence of definite ground-truth concepts and to the task being more complex. Besides being accurate, logic explanations are very short, with a complexity below 4 terms. In conjunction with the explanation accuracy, this means that CDM finds a small set of predicates which accurately describes the most relevant concepts for each class.






















Table 2.12: Accuracy and complexity of logic explanations found using the Concept Distillation Module. Accuracy is computed using logic formulas to classify samples based on their concept encoding. Complexity measures the minterms in logic formulas.

	Logic Explanation Accuracy (%)	Logic Explanation Complexity
BA-Shapes	96.56 (92.17, 100.95)	3.10 (2.75, 3.45)
BA-Community	81.43 (78.20, 84.66)	3.85 (3.09, 4.61)
BA-Grid	99.61 (98.86, 100.36)	1.30 (0.74, 1.86)
Tree-Cycle	90.49 (78.43, 102.55)	1.90 (1.22, 2.58)
Tree-Grid	89.66 (82.71, 96.62)	2.20 (1.07, 3.33)
Mutagenicity	59.94 (44.99, 74.90)	2.60 (0.88, 4.32)
Reddit-Binary	71.84 (54.10, 89.59)	1.60 (1.08, 2.12)

The Concept Distillation Module supports human interventions (Figure 2.9)

Supporting human interventions is one of the main benefits of more interpretable architectures that learn tasks as a function of concepts. In contrast to vanilla GNNs, CDM enables interventions at concept-level, which allows human experts to correct mispredicted concepts. Similarly to Concept

Table 2.13: An example of a concept-based logic explanations discovered by the Concept Distillation Module per dataset. Blue nodes are the instances being explained, while orange nodes represent their p -hop neighbors. For Mutagenicity the color of each node represents a different chemical element. The logic formulae describe how the presence of concepts can be used to infer task labels. For example, the first logic rule states that the task label “middle nodes in house motifs” ($y = 2$) can be inferred from the concepts: “middle node with attaching edge on the near side” or “middle node with attaching edge on the far side”.

Dataset	Concept-based Logic Explanation	Ground Truth Concepts
BA-Shapes	$y = 2 \leftarrow$  OR 	 <i>Node in house motif</i>
BA-Grid	$y = 1 \leftarrow$ 	 <i>Node in grid motif</i>
Tree-Grid	$y = 1 \leftarrow$  OR 	 <i>Node in grid motif</i>
Tree-Cycle	$y = 1 \leftarrow$  OR  OR 	 <i>Node in circle motif</i>
BA-Community	$y = 3 \leftarrow$  OR  OR 	 <i>Node in house motif</i>
Reddit-Binary	$y = \text{“Q/A”} \leftarrow$  OR 	 <i>Star motifs</i>
Mutagenicity	$y = \text{“mutagenic”} \leftarrow$ 	 <i>Ring motifs or NO₂</i>

Bottleneck Models [154], our results show that correcting concept assignments significantly improves the model test accuracy to over 98% for the synthetic datasets, achieving 100% test accuracy on BA-Grid and BA-Shapes. We also observe an increase in task accuracy in BA-Community, however, the increase is much more gradual. Most notably, in both real-world datasets CDMs allow GNNs to improve their task accuracy by up to $\sim +10\%$ with less than 10 interventions.

Ablation Study on Tau

In order to verify our choice of tau ($\tau = 0.5$), we run an ablation study. We adapt the value for tau in 0.1 intervals and calculate results for the BA-Shapes and BA-Community datasets in line with our previous evaluation. We collect the completeness score and number of clusters found, as these metrics are most indicative of the effect on the explanation scope provided. Table 2.14 summarises the results. The optimal value for tau varies across the three datasets. Going by the completeness score, the optimal values are 0.6 and 0.8 for BA-Shapes, BA-Community, respectively. However, based on the number of clusters found, which can indicate a more rare set of concepts being found, the optimal values for tau would be 0.1 and 0.9 respectively. It can be argued that concept completeness is a better indicator, as it directly correlates the concept to the prediction of the output label, nevertheless, it easily glosses over rare concepts. This trade-off must be considered, wherefore, it can be argued that choosing tau at 0.5 is a robust and conventional parameter setting, as differences in results are minute. Nevertheless, for optimal results tau should be finetuned, as it is dependent on the dataset.

Qualitative Comparison to GNNExplainer and GCExplainer

We perform a qualitative comparison of the explanations produced using CDM, GCExplainer [182] and GNNExplainer [336]. We limit ourselves to a qualitative comparison against GCExplainer here, as we have already performed a quantitative comparison above. We compare the explanations produced by CDM against those of GNNExplainer, as GNNExplainer is the most prominent explainer

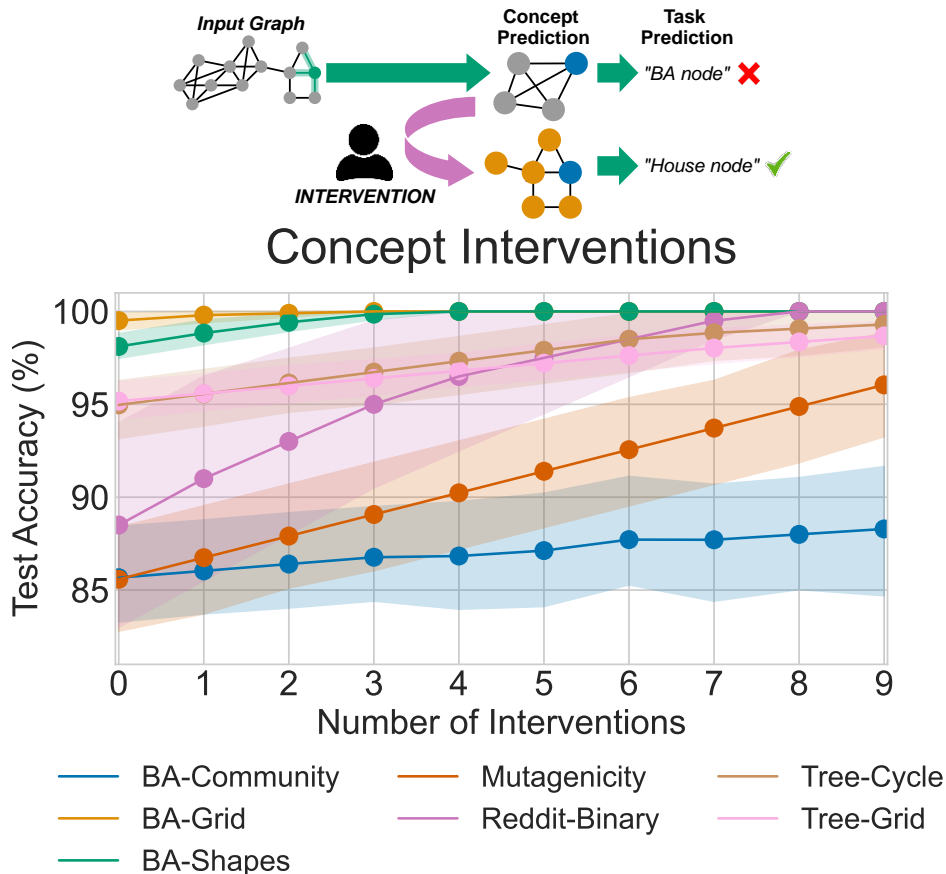


Figure 2.9: The Concept Distillation Module supports interventions at concept-level, allowing human experts to correct mispredicted concepts, increasing human trust in the model [261]. This interaction significantly improves task performance, achieving almost 100% accuracy on synthetic datasets.

for GNN, forming seminal work. However, we do not perform a quantitative evaluation against GNNExplainer, as the explanations are not concept-based and thus are evaluated in a different manner. For this reason, we refrain from also evaluating against other comparable explainers, such as PGExplainer [179] and PGM-Explainer [306]. We select GNNExplainer over these explainers as it is the seminal work in the field. GCEExplainer and GNNExplainer are applied on the vanilla GNN. We focus on an evaluation of the BA-Shapes and BA-Community datasets, as the ground truth motifs to be extracted are known for these datasets.

BA-Shapes (Figure 2.10)

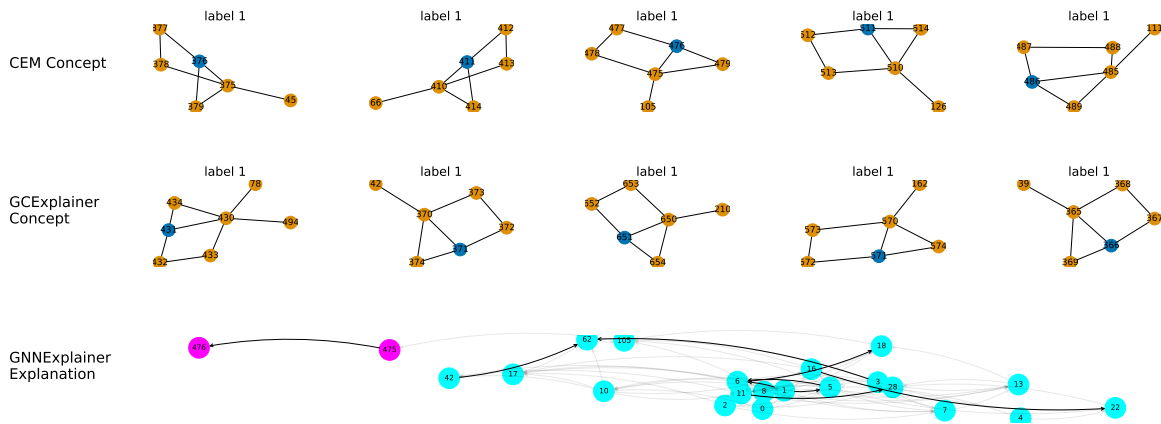
Figure 2.10 shows the explanations provided by CDM, GCEExplainer and GNNExplainer for a node, which is part of the middle of a house. Both CDM and GCEExplainer successfully identify the house structure, which is the motif that should be recovered. CDM performs slightly better than GCEExplainer, as it does not include a concept with a random edge. In contrast, the explanation provided by GNNExplainer does not visualise the house structure in full. Only the middle nodes of the house are visualised (purple), as well as a large part of the BA graph (turquoise). It can be argued that the explanations provided by CDM and GCEExplainer are more intuitive, however, GNNExplainer highlights important edges.

We struggle to reproduce the quality of explanations presented by [336] for GNNExplainer. We

Table 2.14: The concept completeness score and number of clusters discovered when varying the value of τ .

Tau	BA-Shapes		BA-Community	
	Completeness (%)	Number of Clusters	Completeness (%)	Number of Clusters
0.1	48.04 (42.64, 53.44)	24.80 (20.38, 29.22)	59.18 (53.99, 64.37)	50.00 (31.33, 68.67)
0.2	55.07 (49.64, 60.50)	22.60 (20.03, 25.17)	59.54 (52.37, 66.72)	60.00 (32.00, 89.00)
0.3	58.39 (51.61, 65.17)	21.80 (19.76, 23.84)	62.24 (51.86, 72.62)	68.20 (35.17, 101.23)
0.4	58.83 (53.55, 64.12)	20.80 (18.25, 23.34)	62.58 (52.21, 72.95)	81.60 (34.31, 128.89)
0.5	59.59 (55.01, 64.17)	21.40 (19.98, 22.82)	62.17 (49.49, 74.86)	86.80 (36.54, 137.06)
0.6	60.40 (56.82, 63.97)	20.60 (17.13, 24.07)	62.64 (49.54, 75.73)	85.20 (35.51, 134.89)
0.7	59.18 (50.00, 68.36)	21.40 (15.41, 27.39)	63.56 (49.03, 78.09)	88.00 (37.55, 138.45)
0.8	58.89 (47.68, 70.09)	20.00 (13.98, 26.02)	63.89 (49.25, 78.53)	80.40 (29.86, 130.94)
0.9	54.83 (44.62, 65.04)	21.00 (15.18, 26.82)	58.42 (43.88, 72.97)	97.20 (33.88, 160.52)

Explanations for Node 476 in the BA_Shapes Dataset found using different Techniques

**Figure 2.10:** Concept-based explanations produced using the Concept Distillation Module and GCExplainer, as well as the explanation produced by GNNEExplainer for a node in the BA-Shapes dataset. In the explanations, the blue nodes are the nodes clustered together, while the orange nodes are the p -hop neighborhood. GNNEExplainer has its own coloring, where the purple nodes are node part of the middle of the house and the turquoise nodes are part of the BA base graph.

first adapted the threshold to observe an effect on the explanations, however, this only impacted the visualisation of the important edges. We fix the threshold at 0.8 after this. We then examined the implementation of GNNEExplainer used. To ensure that the quality of explanations is not the fault of the PyTorch Geometric [84] implementation of GNNEExplainer, we also used the implementation provided by the Deep Graph Library [308]. After obtaining similar results, we exhaustively visualize the explanations for class 1. We present a selection in Figure 2.11. In summary, we fail to produce the house motif using GNNEExplainer, as the explanations provided mostly emphasize the importance of the BA base graph.

BA-Community (Figure 2.12)

Lastly, we compare the explanations for a node in the BA-Community dataset (Figure 2.12). Similar to our previous observations, both CDM and GCExplainer successfully identify the house structure. More importantly, they both identify the existence of random edges to explain the node. In contrast, the explanation provided by GNNEExplainer is more elusive, highlighting mostly the BA base structure. In conclusion, it can be stated that the concept representations for CDM and GCExplainer are



Figure 2.11: A selection of explanations produced using GNNE explainer for nodes of class 1. The purple nodes are nodes part of the middle of the house, while the turquoise ones are part of the BA base graph.

almost identical, which can be attributed to the same visualisation technique being used. However, we refer the reader back to the quantitative evaluation in the results section, which highlights the strengths of CDM over GCE explainer.

Quantitative Comparison to ProtGNN

While ProtGNN citezhang2022protgmn is not concept-based, it has a similar aim of producing an interpretable model as opposed to post-hoc explanation. We compare CDM to ProtGNN via classification accuracy on a synthetic node-classification dataset and real-world graph classification dataset, as [349] do not explain how the quality of their explanations should be evaluated. We perform the evaluation on ProtGNN+, which uses a novel conditional subgraph sampling module for

Explanations for Node 434 in the BA_Community Dataset found using different Techniques

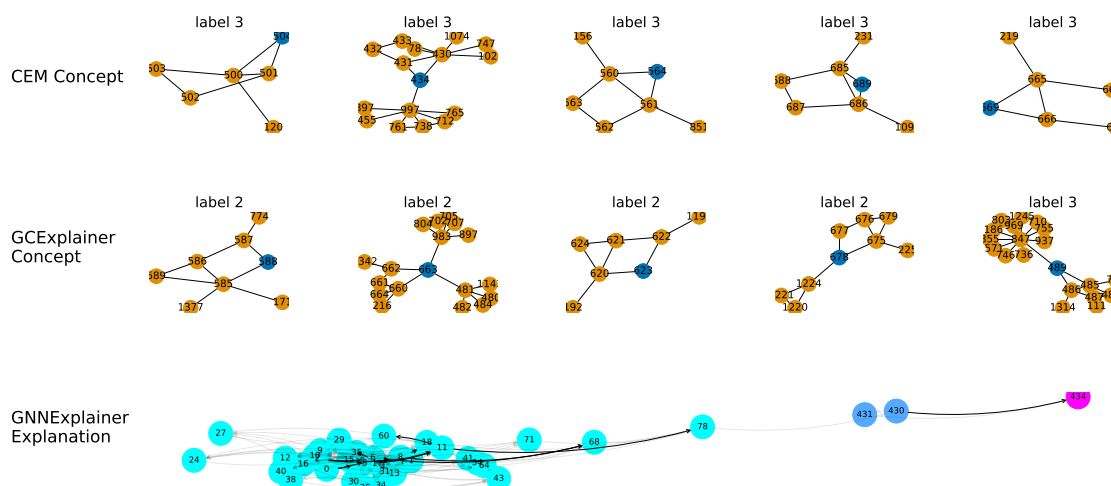


Figure 2.12: Concept-based explanations produced using the Concept Distillation Module and GCExplainer, as well as the explanation produced by GNNEExplainer for a node in the BA-Community dataset. In the explanations, the blue nodes are the nodes clustered together, while the orange nodes are the p -hop neighborhood. GNNEExplainer has its own coloring, where the purple node is the 'top' node in the house, the blue nodes are the 'middle' of the house and the turquoise nodes are part of the BA base graph.

improved efficiency and interpretability. CDM outperforms ProtGNN+ on the BA-Shapes dataset, achieving an accuracy of 98.11% (97.04%, 99.18%) in comparison to 96.94% (95.38%, 98.51%). In contrast, CDM and ProtGNN+ achieve similar accuracy on the Mutagenicity dataset with 82.4% (81.3%, 83.5%) and 81.7% (75.4%, 88.1%), respectively. We note that we only compute the results for Mutagenicity for 2 seeds, as training ProtGNN took significantly longer than training CDM, requiring 12 hours for 500 epochs on the same hardware.

There are significant architectural differences between CDM and ProtGNN. Firstly, ProtGNN requires to define the number of class prototypes, while CDM only requires to define an upper bound via the concept embedding size s . Moreover, CDM allows to extract fine-grained subgraphs and the ability for human intervention. Moreover, ProtGNN does not provide formal explanations, which can be evaluated quantitatively. Lastly, we found that ProtGNN runs significantly slower than CDM, though some issues may be alleviated via optimising the implementation.

2.3.6 Discussion

Concept Graph Networks are accurate and self-explaining

In summary, our results demonstrate that CDM makes GNNs explainable by design without impairing their task performance. Our approach extracts high-quality concept-based and logic explanations. Our experiments show that the extracted concepts are pure, meaningful and interpretable, while task-specific logic explanations are simple and accurate. We also demonstrate that CDM supports human interventions at concept level, which is one of the main advantages of explainable-by-design architectures.

Strengths and Limitations

The main limitation of our work is the association of only one concept per sample. However, this also applies to GCExplainer, as well as to state-of-the-art unsupervised explainability methods for convolutional networks, such as ACE [98]. The second main limitation pertains the p -hop neighborhood visualization technique inherited from GCExplainer. Visualizing a concept by simply exploring the p -hop neighborhood may include nodes which are not relevant for identifying a concept. More specifically, the concept visualization technique could be improved by performing largest common subgraph matching across the samples representing a concept. However, such an approach would be extremely expensive in terms of computations even for small graphs and it would not scale for large concepts. In terms of novelties, the proposed approach is the first of its kind in terms of making GNNs explainable by design. Secondly, the approach allows to find the optimal number of concepts dynamically. While the size of the embedding space must still be defined, this size is just an upper bound, alleviating the user from the burden of tuning this hyperparameter as in other explainability methods, such as GCExplainer or ACE. This dynamic adaptation often produces a high number of clusters/concepts. While a higher number of concepts may appear redundant and be more complex to reason about, the extracted concepts accurately describe the dataset, as indicated by the high concept completeness scores and rare concepts found. Moreover, logic-based formulas allow to filter through the concepts relevant for each class. We note, however, that as stronger interpretable models are deployed, there are risks of societal harm which we must be vigilant to avoid.

2.3.7 Conclusions

In this work, we address the lack of human trust in GNNs caused by their opaque reasoning. To this aim, we propose the Concept Distillation Module which makes GNNs explainable by design. We demonstrate that the proposed method allows to discover and extract high-quality concept-based explanations. The proposed approach makes GNNs explainable by design without a reduction in performance, while also allowing for human intervention. Human intervention allows to alleviate dataset biases, further increasing trust in the model. The increased understanding of the model’s working through the proposed approach fosters an increase in trust and may open up the possibility to use GNNs in more high-stake scenarios.

2.4 Explainable-by-design Machine Learning Model for Overlapping Fluorophores Separation Based on Fluorescence Lifetime

In fluorescence microscopy imaging, the ability to discriminate between different fluorophores based on their temporal fingerprint is highly desirable. In this theoretical study we propose a deep learning method to separate the signal contribution of two spectral overlapping fluorophores in a time-resolved fluorescence microscopy image. Our method exploits a CNN-based network that analyzes both the temporal information and the 2D spatial features. Since the purpose of the network is to separate contributions from different fluorophores, the neural network is divided into two sections, each separating one of the two different fluorescence time decay components. We focus on improving the explainability of the process, making our approach highly interpretable to facilitate a better understanding of the underlying mechanisms.

2.4.1 Introduction

To understand the intricate connections between sub-cellular components and macro-molecular complexes that make up a cell, it is crucial to simultaneously label and visualize different species of bio-molecules [71]. The most widely employed approach to achieve this goal involves utilizing spectrally separable fluorophores, to unmix the signals with a filter on the emission band of each fluorophore. However, a significant spectral overlap between the fluorophores poses a challenge, which can be mitigated by carefully selecting fluorophores that do not exhibit such interference. Nevertheless, the choice of non-spectrally overlapping fluorophores limits the number and type of sub-cellular components that can be labelled simultaneously. Hence, the ability to discriminate between fluorophores based on their temporal characteristics, independent of their emission spectra, as the fluorescent lifetime, assumes paramount importance. Numerous methods have been proposed to distinguish fluorophores based on their temporal or spectral fingerprints. Examples include the phasor approach [72] and SPLIT (Separation of Photons by Lifetime Tuning) [159]. However, these methods rely on the first term(s) of the Taylor temporal series calculated pixel by pixel. Consequently, they are unable to capture valuable information embedded in the non-linear components and information sharing across pixels. Furthermore, these methods are user-dependent and sensitive to the signal-to-noise ratio, making them susceptible to failure in cases involving significant spectral overlap or when the prior information provided is incorrect or incomplete. To overcome these limitations, we propose leveraging an explainable-by-design [46] deep neural network (as [269]), which learns the temporal decay model in a data-driven manner, addressing the shortcomings of the previous linear-based methods. Existing neural network-based methods for separating fluorophores with spectral overlap [60] lack information on temporal characteristics, raising uncertainty about the alignment of the separation with the physical model of image formation. In contrast, our

method, based on the theoretical model, provides comprehensive output, including lifetime values and unmixed images, facilitating insights into underlying biological processes.

2.4.2 Data and Methods

Data

The data consist of image sequences with dimensions $W \times H \times T$, representing the temporal decay of fluorescence in biological samples labelled with multiple fluorophores. In this case study, we assume that the fluorophores are spectrally overlapping.

Theoretical model

The theoretical image formation model states that the image sequence $Image$, of dimensions $W \times H \times T$, can be separated into two components $Image_1$ and $Image_2$ using the formula:

$$\begin{aligned} Image(x, y, t) &= Image_1(x, y, t) + Image_2(x, y, t) \\ x &\in \{1, \dots, W\}, y \in \{1, \dots, H\}, t \in \{1, \dots, T\} \end{aligned} \quad (2.15)$$

Each $Image_i$ can be further divided into two parts: once a spatial distribution A is determined, a temporal decay component $\vec{v}_i(t)$ is incorporated, according to the physical law of fluorescence decay over time [232]. Lastly, a Poisson noise \mathcal{P} is added to the temporal decay.

$$Image_i(x, y, t) = A_i(x, y) \times \mathcal{P}(\vec{v}_i(t)) \quad i \in \{1, 2\} \quad (2.16)$$

The temporal decay of both images is obtained through a convolution between the temporal decay function $e^{-\frac{t}{\tau}}$ and the Instrument Response Function (IRF).

$$\vec{v}_i(t) = f(t | \tau_i) = e^{-\frac{t}{\tau_i}} * IRF(t) \quad i \in \{1, 2\} \quad (2.17)$$

$$IRF(t) = \frac{1}{2\pi\sigma_t} e^{-\frac{t^2}{2\sigma_t^2}} \quad (2.18)$$

where the standard deviation $\sigma_t = \frac{FWHM}{2\sqrt{2\ln 2}}$ with FWHM being dependent on the measurement instrument. In a real-world scenario, Poisson noise often affects the data due to the measurement instrument. However, in constructing the neural network, we do not take this into account because the Poisson distribution is characterised by the mean, so the network will learn to estimate this parameter even in the presence of noise. Furthermore, we want the prediction to be deterministic: this is important for many practical applications, as it allows to make reliable and consistent estimations. Optical microscopy images are diffraction limited with a resolution limit that can be estimated through the so-called Abbe's law. An additional blurring effect can occur due to factors such as focus and lens imperfections. For these reasons, the transfer function of an optical

microscope can be modelled with a Gaussian Point Spread Function (PSF).

$$A_i(x, y) = Phantom_i * PSF(x, y) \quad i \in \{1, 2\} \quad (2.19)$$

$$PSF(x, y) = \frac{1}{2\pi\sigma_x\sigma_y} e^{\left(-\frac{x^2}{2\sigma_x^2} - \frac{y^2}{2\sigma_y^2}\right)} \quad i \in \{1, 2\} \quad (2.20)$$

The parameters are set according to the Abbe law ([1]) for diffraction limit: the standard deviations $\sigma_x = \sigma_y = \frac{FWHM}{2\sqrt{2\ln 2}}$ with the Full Width at Half Maximum $FWHM = \frac{\lambda_{exc}}{2NA}$. The wavelength excitation λ_{exc} and the numerical aperture NA are dependent on the experiment.

Synthetic data generation

The theoretical model discussed above is therefore used to generate synthetic images to train the proposed model. In fact, when an adequate number of image sequences are unavailable, data can be generated from the theoretical model (section 2.4.2). Synthetic images ($Phantom_1$ and $Phantom_2$) of tubulin and nuclei, which resemble biological applications, can be created by varying shape parameters such as the number and size of filaments, the number and size of molecules within the nucleus, the dimensions of the nucleus, and the intensities. Subsequently, two values of τ_1 and τ_2 can be set. At this point, all the necessary quantities can be generated to obtain the final image. Synthetic data generation has a significant time advantage over manual acquisition.

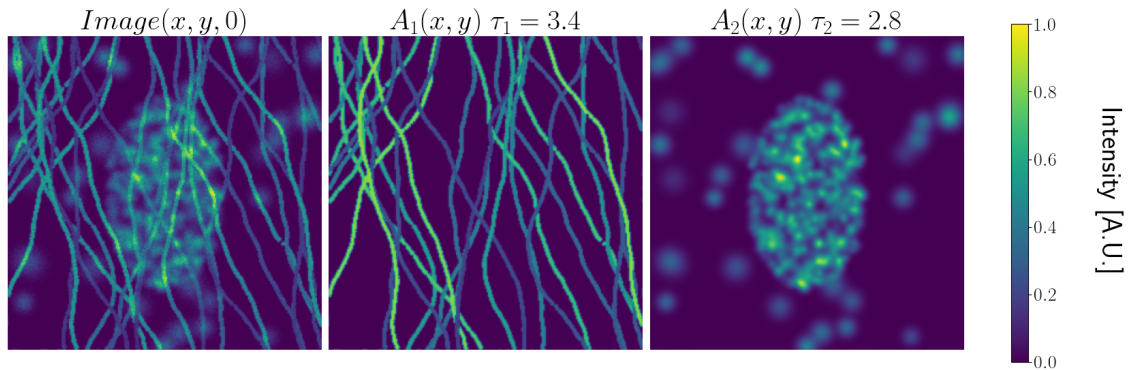


Figure 2.13: Example of a datum. The first image represents a single time instant of the image sequence. This is followed by the two separate components with their respective temporal decay rates.

Model

Figure 2.14 illustrates the neural network architecture. The network consists of a CNN that takes an image sequence $Image$ of dimensions $W \times H \times T$ as input. This CNN combines spatial and temporal information to capture the essence of the image sequence. The output of the CNN is then fed into four separate CNNs: two of them (CNN_{τ_1} and CNN_{A_1}) model the first component, while the other two (CNN_{τ_2} and CNN_{A_2}) model the second. These five CNNs form the trainable part of the model. The outputs of this section, τ_1 , A_1 , τ_2 , and A_2 , represent the interpretable part of the network. They not only reveal the values of τ_1 and τ_2 used to separate the two signals, but also show the images A_1 and A_2 , which represent the specific sections of the image sequence that contribute

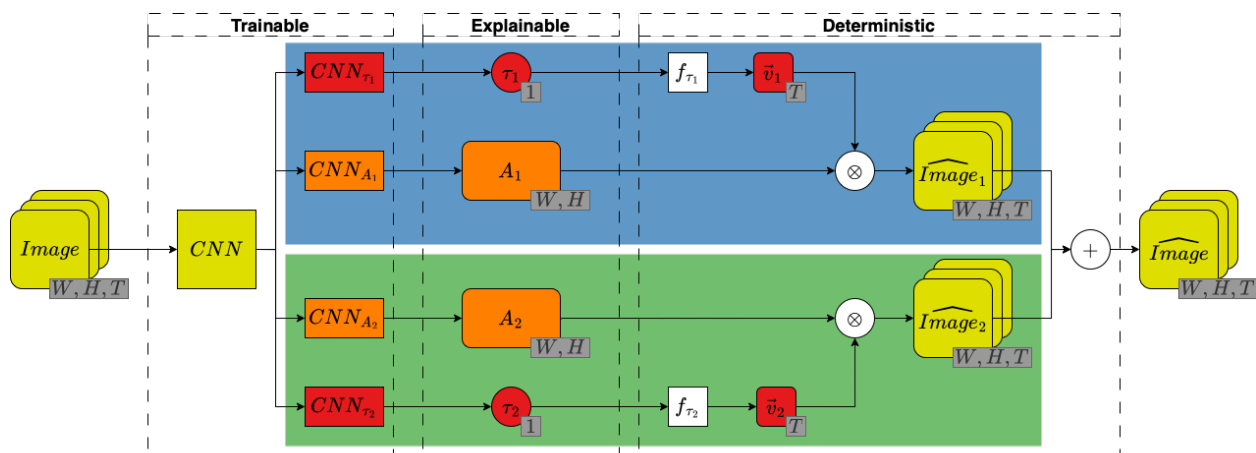


Figure 2.14: Design of the proposed model. Each vector’s dimensions are shown in the bottom right-hand corner in grey. In orange and red are shown the parts estimating the spatial and temporal features, respectively. In blue and green are represented the sections that model component 1 and component 2, respectively. The network is divided into three sections. A trainable one, consisting of 5 CNNs, one to perform feature extraction and 4 for prediction. The second section provides an explanation of the network’s output. A final deterministic one reconstructs the input image.

to each component. To train the network, it is necessary to reconstruct the original image from these four outputs. This can be easily achieved by first transforming each τ_i into a vector \vec{v}_i of dimension T using the function f_{τ_i} . Using an outer product, it is then possible to reconstruct two separate image sequences, \widehat{Image}_1 and \widehat{Image}_2 , which can be summed to form the complete image sequence \widehat{Image} . This final part of the network is deterministic, so it does not require any weight to be updated. The whole network is then trained to make \widehat{Image} match $Image$. This can be done by employing a Mean Squared Error Loss. Each $Image$ comes from the synthetic image training dataset, but in the presence of real data, those can also be used. So far, CNNs are not specified since the proposed model is valid beyond the specific CNN used. In our experiments, we specifically used DeepLabV3 model with a MobileNetV3-Large backbone [47] for both CNN , CNN_{A_1} and CNN_{A_2} , while for $CNN_{\tau_{au_1}}$ and $CNN_{\tau_{au_2}}$ the SqueezeNet model architecture [134] has been applied.

2.4.3 Discussion

The advantages of the new network structure can be summarized as follows:

- Thanks to the properties of CNNs, the network simultaneously takes into account both the temporal and spatial components of the input data. This allows the network to capture changes in the fluorescence signal over time and take advantage of the spatial information to predict the temporal decay. This is especially important because classical methods typically ignore the spatial component since they consider a single pixel.
- There is no need to know any τ_i or A_i values in advance. Therefore, only the initial image sequences are required, and the network learns how to separate the signals on its own. However, if these parameters are known, one could consider adding an additional loss on these parameters to improve the model’s performance.

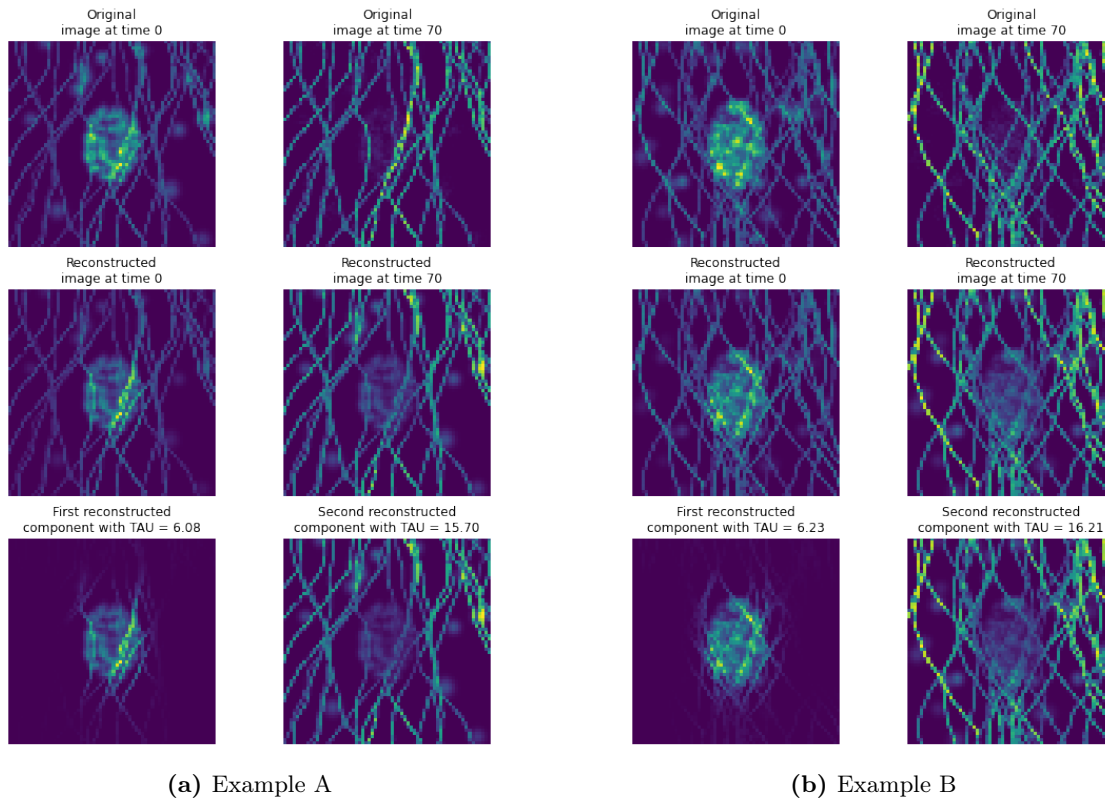


Figure 2.15: Two examples from the test set to show the network’s functioning. In the first line, we can find the original image at two different instants of time. In the second line, the image reconstructed by the neural network is shown at the same time instants. The third row presents the two components separated by the network, with their respective estimated τ_i .

- If only one of the two parameters, τ or A , were available, the network could be trained to reconstruct it. However, a network trained in this way would not have any constraints on reconstructing the original image sequence. So, it could violate the photon flux conservation, predicting something physically incorrect because the calculations are not forced to add up correctly. This is not the case with our model because, having a deterministic section, it is forced to respect physical constraints.
- Our network provides not only the image as in [60] but also the value of the temporal decay τ . This feature has significant implications in biomedical applications, where expert biologists can use this information to gain insights into the underlying biological processes.
- Unlike other neural network approaches [60], the functioning of the network is easily interpretable. Specifically, information about the fluorescent lifetime value τ and image A are provided separately. From these outputs, an expert can evaluate the performance and gain insight into the inner workings of the network. For example, an expert can use the network outputs to identify tubulin and nuclei and investigate potential biological mechanisms underlying the sub-cellular components.

2.4.4 Results

Figure 2.15 shows an example of how the network works. We can see how the network is able to almost faithfully reconstruct the image given as input. The network has some difficulty in correctly estimating the decay of the nuclei. In fact, we can see how part of it is still visible at a time instant where, in the original image, it is no longer present. We can also note how the model manages to distinguish accurately between the two components, failing slightly when the two cross each other. Although the results are not perfect, we believe they are a great first step for the application of explainable-by-design neural networks in molecular biology.

2.4.5 Conclusion

In this work, we proposed a novel deep learning approach for the reconstruction of multi-channel fluorescence image sequences in the presence of spectral overlap between fluorophores. In this study, we introduce a novel deep learning approach for separate fluorescence images when fluorophores exhibit spectral overlap. While the primary application scenario of our method is for spectrally overlapping fluorophores, where spectral filtering based on emission wavelength fails to separate, it is versatile enough to be employed even when this assumption does not hold. Notably, our model can also be trained on real data without the need to know the ground truth decomposition. We decided to use synthetic data because of their twofold advantage: they are faster to acquire and they allow network evaluation by comparing decomposition. Our model is designed to enhance the reliability of the network. Specifically, it leverages a physical model of fluorescence decay over time, enabling the separation of the fluorescence signal into spatial and temporal components. This makes the network structure more justified, as it reflects a physical law. Moreover, one of the main advantages of our approach is its ability to estimate the lifetime τ . The estimation of these parameters holds particular significance in biological studies as it contributes insights into the dynamic nature of biological processes, facilitating a deeper understanding of the underlying mechanisms. In conclusion, our proposed method holds the potential to revolutionize the processing and analysis of fluorescence lifetime microscopy data, enabling more accurate and efficient exploration of complex biological systems. Future work will involve evaluating the performance of the proposed neural architecture on both synthetic and real-world datasets, validating its effectiveness in achieving the envisioned outcomes. Additionally, a comparative study against state-of-the-art methods will be conducted to establish the superiority of our approach.

Chapter 3

Robust Losses for AI Systems

In the ever-evolving landscape of AI applications, robustness is a cornerstone of Trustworthy Artificial Intelligence. AI systems often operate in environments characterized by dynamic data distributions, adversarial perturbations, and the presence of noisy labels. These challenges can undermine the performance and reliability of conventional models trained with standard loss functions designed for ideal scenarios. As such, there is an urgent need to bolster AI systems against these adversities. Robust losses, specially tailored for such scenarios, emerge as a compelling solution.

Noisy or mislabeled data is a common occurrence, whether due to human error in labeling or inherent ambiguity in the data. Robust losses provide a mechanism for models to learn and generalize effectively even in the presence of noisy labels. By penalizing inconsistencies between predictions and noisy labels, these loss functions enable the AI system to maintain its accuracy.

Adversarial attacks pose a significant threat to AI systems, as malicious actors can manipulate input data to deceive models. Robust losses can act as a barrier against such attacks by penalizing model responses to adversarial inputs. This proactive defense mechanism strengthens the AI system's resistance to adversarial perturbations, making it more reliable in real-world, high-stakes applications.

Dynamic environments often introduce outliers or shifts in data distributions that can mislead AI models. Robust losses are engineered to encourage models to identify and adapt to such changes, promoting resilience and maintaining performance even in the face of unforeseen shifts in data patterns.

This chapter explores the multifaceted nature of robust losses, emphasizing their pivotal role in addressing noisy labels, adversarial attacks, and data shifts. By incorporating these concepts, AI systems can fortify their resistance to challenges and uncertainties, ultimately fostering trustworthiness in their predictions and decisions.

3.1 Robust Training of Sequential Recommender Systems with Missing Input Data

In the realm of sequential recommender systems, understanding users’ preferences based on their past actions is paramount. Yet, the susceptibility of these models to input perturbations has limited their practicality. Addressing this, we present an innovative approach to mitigate the impact of missing input items, a challenge that has been overlooked. Our method involves a novel training process that anticipates data loss and employs an optimization loss to predict multiple future items. Extensive evaluations on diverse datasets and recommender models underscore its effectiveness. Notably, our approach enhances NDCG@10 by up to 18% with one missing item and an impressive 230% with five missing items, underscoring its substantial impact on system resilience and performance. This work sheds light on the intricate dynamics of sequential recommendation and offers a potent solution to real-world data limitations.

3.1.1 Introduction

Sequential recommendation models have raised interest in recent years for their promising increasing performance in various domains such as e-commerce, health and education [36, 225]. However, machine learning models are sensitive to input perturbations [112], and particularly, sequential recommendation models were shown to be vulnerable to even a single change in the training data [205, 206]. The robustness of recommender systems to data perturbations is a desired property and is essential in various domains. Suppose a user regularly uses an e-commerce platform to buy clothes. The platform collects data on the user’s past purchases and browsing behavior to make personalized recommendations for future purchases. However, the user decides to take a break from the platform for a few weeks and shops for clothes elsewhere. During this break, the e-commerce platform is unable to collect data on the user’s behavior, resulting in missing data. When the user returns to the platform, the recommender system must take into account the missing data and still provide personalized recommendations based on the user’s past purchases. Missing data can even be dangerous in some domains, such as healthcare [292], where patients might have been treated at different clinics, and this might result in incorrect diagnoses or treatments. Specifically, in sequential recommendation systems, the recommendation is based on the sequence of user actions, so the most recent actions might have an even stronger effect on the generated recommendations. Considering this, we explore the impact of missing data in the last items of the sequence and how to mitigate it by training the models differently. To the best of our knowledge, this is the first work verifying that existing sequential recommender systems suffer from this effect and applying a method to make sequential recommender models more robust to this type of data perturbation. We can summarise our contributions as follows:

- Our investigation shows that several sequential recommendation models heavily rely on the last items in the sequence.

- We apply a modified training method to make the models more robust to such missing data perturbations.
- Our model outperforms (as measured by Hit Rate and Normalized Discounted Cumulative Gain) classical models in cases of missing data while maintaining or improving performance in the next item prediction task.

3.1.2 Related Work

Sequential Recommendation

Sequential recommendation is a subfield of recommendation systems [239] that focuses on recommending items to users based on their recent interactions. The goal of sequential recommendation is to predict the next item a user will likely interact with, given their previous interactions. One of the earliest sequential recommendation methods is the Markov Chain model [89, 237, 260], which models users' interactions as a Markov process and uses the transition probabilities between items to make recommendations. Recently, there has been a growing interest in using deep learning techniques for sequential recommendation. These methods include using deep neural networks, such as Recurrent Neural Networks (RNN) [123], Long Short-Term Memory (LSTM) [329], Gated Recurrent Units (GRUs) [54, 124] and attention mechanisms [141, 165, 283], to model users' interactions and make recommendations, allowing the model to focus on the most relevant parts of the user's interaction history when making recommendations. Additionally, there has been an increased focus on Explainable AI in sequential recommenders [203, 346], some of which are based on counterfactuals [50, 51, 95, 285, 291], which are aimed at making the recommendations more tailored to the user [311, 314, 345] and providing more transparency into the decision-making process of the model. Overall, the field of sequential recommendation is rapidly evolving, with a wide range of methods and techniques being proposed and evaluated.

Robustness of recommender systems

One of the main challenges in the field of recommendation systems is ensuring the robustness of the models to data perturbations [39, 69, 208]. Data perturbations refer to small changes in the input data, such as missing values or noisy observations, that can significantly impact the model's performance. Many common recommendation methods are sensitive to such perturbations [205, 206] and can lead to poor performance or even complete failure. Recently, there has been an increased focus on developing robust sequential recommenders that can handle data perturbations. One approach is to use regularization techniques, such as dropout [92, 166], to reduce the impact of noise in the input data. Another approach is to use ensemble methods, such as bagging [303] and boosting, to combine the predictions of multiple models. Another area of research on robustness is the use of generative models, such as Variational Autoencoders (VAEs) [168] or Generative Adversarial Networks (GANs) [321], to learn the underlying distribution of the data and generate new samples, which can be used to augment the training data [170] and improve the robustness of the models. Additionally, there has been work on imputation techniques [135, 324], to infer the missing data to improve recommender systems, and on training instability [287]. Finally, other works focus on methods for evaluating the robustness of the model without using ranking evaluation metrics

but rather by assessing the stability of the generated rankings in the presence of missing items in the data [205, 206]. Overall, robustness is a critical issue in sequential recommendation. There are many ongoing efforts to develop methods that can handle data perturbations and improve the performance of the models in practice.

3.1.3 Setting

The setting in question consists of N_U users and N_I items. Each user u_i has interacted with at least one item I_j at a given time $t_{i,j}$. The goal of a recommender system is to predict the compatibility between a given user and the items with which it has not yet interacted, knowing the items the user has interacted with. In the Sequential Recommendation case, the problem takes the form of predicting the next item in a sequence: given a sequence of i items $\{I_1, I_2, \dots, I_i\}$ with which a user u has interacted, the goal is to predict the $i + 1$ -th item (I_{i+1}). A sequential neural network takes as input a sequence of at most L elements and performs, for each time step, the prediction of N_I values. These represent the estimated compatibility between user u , to which the sequence of items belong, and *all*¹ items I .

Classic Training Method

The goal of the network, at each timestep i , is to predict the next item in the sequence I_{i+1} . During network training, if the number of possible items N_I is too large, it becomes intractable to calculate predictions for all of them, therefore, only a chosen few are calculated. In particular, computing the one corresponding to the next item, called *positive item*, and at least one corresponding to an item that is irrelevant, called *negative item*, chosen randomly at each epoch. An attempt is then made to increase the former value at the expense of the latter. Notably, the negative items are indeed chosen randomly, but excluding items already in the input sequence. To achieve this, the loss used for the models we have considered is the Binary Cross-Entropy; this is typically used in a classification scenario.

The definition of loss, positive items, and negative items are defined leads to a specific ranking that the network should achieve at time step i of a sequence of L elements. The ranking can be described to be as follows: first the positive item I_{i+1} , followed by, in indifferent order, all the other items in the sequence I_j such that $j \in \{1, \dots, i, i + 2, \dots, L, L + 1\}$, and finally, again in indifferent order, all the remaining items I_j such that $j \notin \{1, 2, \dots, L + 1\}$. This particular ranking would result in zero loss.

We can simplify, as shown in Figure 3.1a, the functioning of the network by imagining that for the sequence $[I_1]$, the model should output item I_2 , for the sequence $[I_1, I_2]$ it should output item I_3 , and so on.

3.1.4 Problem Statement

A sequential recommender system receives as input a sequence $S = \{I_1, I_2, \dots, I_i\}$ and tries to predict the next item in the sequence, item I_{i+1} . How would the network behave if the last item

¹“all” is used for the sake of simplicity. In reality, prediction is often not done for all items (e.g., during training, for evaluation, or only for items not in the input sequence). More details are in the corresponding sections.

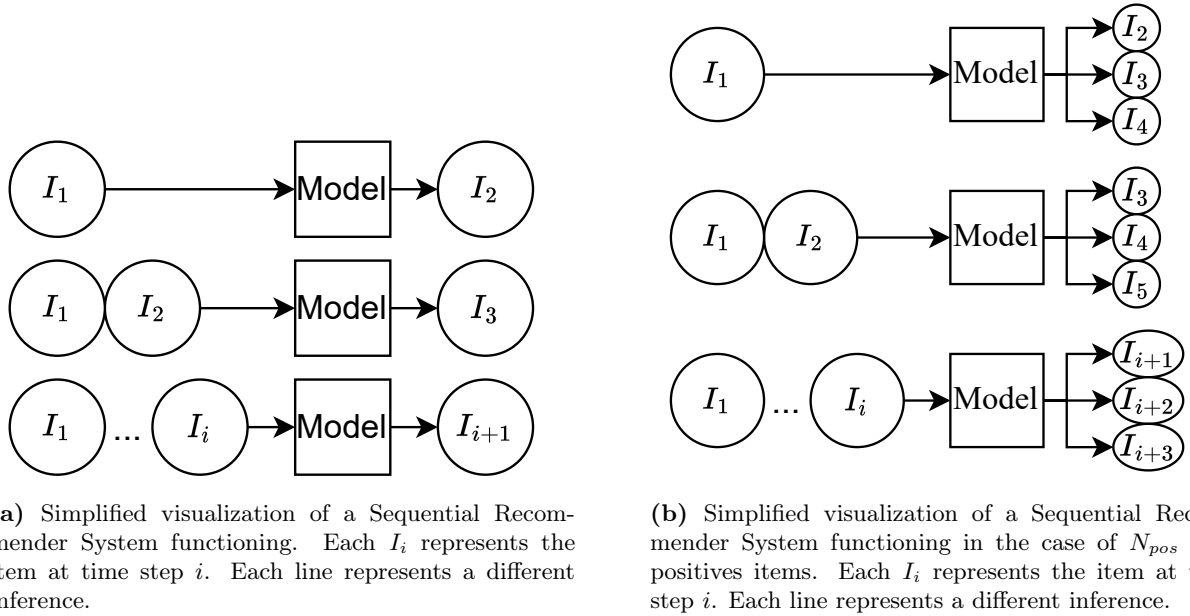


Figure 3.1: Simplified visualizations in the two different scenarios

I_i is missing? If the last item is removed from the sequence $[I_1, I_2]$, we would be left only with the sequence $[I_1]$. Since the network is trained to predict only item I_2 , it will have no preference on predicting I_3 . Furthermore, considering an out-of-sequence division of the dataset (more info on data division in Section 3.1.5), removing two items from the sequence results in replicating a training sequence. Thus, the effect of removal may be even more detrimental if the model is overfitting on the training set. Lack of user-item interactions in real-world scenarios pose a challenge for sequential recommenders. For example, a streaming service offering a movie trilogy may not have data on a user's interaction with the second movie if it was watched on a different platform. This could result in the recommender suggesting the second movie as the top recommendation without considering the third. This issue also applies to non-sequential items like sequels to movies or books. We start by demonstrating that existing sequential recommender models suffer from this effect, and then we devise a method to make them robust to this type of data perturbation.

3.1.5 Methodology

More Positive Items

We assume that, in a real scenario, an ideal model should yield, at a given time step i , a ranking containing, in order, all future items in the sequence, and only upon finishing these, all other (negative) items. Therefore, the solution we have applied is to choose N_{pos} positive items, such that the network learns to simultaneously predict N_{pos} future instances. Please note that we are not trying to predict the whole sequence of future interactions but only the relevance of the items at time step i . In this case, the loss would become as in 3.1.

$$\ell_{\text{BCE,mp}}(\vec{x} \mid \vec{pos}, \vec{neg}) = - \sum_{j \in \vec{pos}} \log(x_j) - \sum_{j \in \vec{neg}} \log(1 - x_j) \quad (3.1)$$

where \vec{x} represents the output of the network, $\vec{pos} = \{p_1, \dots, p_{N_{pos}}\}$ the identifier of the N_{pos}

positive items, and $\vec{neg} = \{n_1, \dots, n_{N_{neg}}\}$ those of the selected N_{neg} negative items. The loss takes the same form as that presented in [286]. Although the authors mention the loss function, its potential for improving the model’s robustness in the face of missing data has not been explored. Our work fills this gap by showing how this loss function can be effectively used to increase the robustness of the model and improve its performance in the presence of missing data. Replicating the simplified illustration in Figure 3.1a, we can visualize the idea of predicting multiple positive items as in Figure 3.1b.

Margin Loss

Considering more positive items poses a clear limitation, as it becomes more challenging for the next item to rank high in the network’s ranking. This is because the Binary Cross-Entropy loss does not distinguish between positive items; a perfect model would rank all P positive items in the first P positions, regardless of their order. This might limit the model performance, as the item may end-up in the P -th position, thus reducing common metrics that take into account the order of results, such as NDCG. To solve this problem, we decide to use the Margin Loss. Given pairs of inputs x_1 and x_2 , and a preferred ordering of them y , such that $y = 1$ if we assume that the first input should be ranked higher than the second input, vice-versa for $y = -1$, the margin loss ℓ takes values according to $\ell_{\text{MRG}}(x_1, x_2, y | \text{margin}) = \max(0, y(x_2 - x_1) + \text{margin})$. This tells us that if the network outputs for the two inputs respect the expected ordering and are at least *margin* apart, the loss is equal to 0; otherwise, it is proportional to the distance between them. Input pairs are formed between all pairs of positive items. The expected order is the order in which the user interacted with them: an item at a time step i must come first in ranking than one at a time step $i + k$. The Margin Loss formula is $\ell_{\text{MRG, pos}}(\vec{x} | \vec{pos}, \text{margin}) = \sum_{c=1}^{N_{pos}} \sum_{k=c+1}^{N_{pos}} \max(0, x_{p_k} - x_{p_c} + \text{margin})$, where \vec{x} represents the output of the network, $\vec{pos} = \{p_1, \dots, p_{N_{pos}}\}$ the identifier of the N_{pos} positive items and *margin* the margin value. The equation holds only if the order of the identifiers of the positive elements follows the expected order.

Mixed Loss

The margin loss applied on the positive items is not enough to train the neural network as we desire. It is always necessary to discourage the model from predicting negative items. We, therefore, decide to use it in conjunction with the traditional Cross-Entropy loss. This naturally brings up the need to add some hyperparameters to weigh the importance of the two losses. We also separate the components of the Binary Cross-Entropy loss pertaining to positive items and negative items. This Mixed Loss formula is $\ell_{\text{MIX}}(\vec{x} | \vec{pos}, \vec{neg}, \text{margin}) = l_{\text{BCE, pos}} + \lambda_1 l_{\text{BCE, neg}} + \lambda_2 \ell_{\text{MRG, pos}}(\vec{x} | \vec{pos}, \text{margin})$.

Experiments

Datasets

We select three datasets that are widely used in this field [53, 209]: MovieLens-1M [117], MovieLens-100K [117] and Amazon Beauty [200]. The first two are movie ratings taken from the MovieLens

Table 3.1: Dataset statistics after preprocessing

Dataset	Users	Items	Actions /User Average	Actions /User Median	Actions
MovieLens 1M	6040	3706	165	96	1M
MovieLens 100k	943	1682	106	65	100K
Amazon Beauty	2417	2821	5	5	12K

website² and differ on the period they were collected and the size of the set. The third dataset³ contains reviews and metadata from Amazon, spanning May 1996 - Oct 2018. The three datasets have 165, 106 and 5 interactions per user, respectively.

The statistics for all the considered datasets are shown in Table 3.1.

Models

We select three sequential recommendation models. The first, GRU4REC [124], is an RNN based on Gated Recurrent Unit. SASRec [141], on the other hand, is a sequential self-attention based model that uses an attention mechanism to make predictions based on a relatively small number of actions. TiSASRec (Time Interval aware Self-attention based sequential recommendation) [165] is instead a modification of this that adds to the input the time intervals between elements in the sequence.

Preprocessing

Consistent with other work, we use implicit ratings, so we do not consider the score but simply the existence of an interaction of a given user with a given product. Given a user u , the products he interacted with are ordered in a sequence S_u based on the timestamp. An out-of-sequence split (i.e. the last two items in each sequence are kept aside to be the target output of validation and test, respectively, while the rest of the sequence is used for training) is performed to partition the data into training, validation and test sets, in line with what has been done by other works in the same domain.

Evaluation

In line with what has been done in other works involving Neural Recommenders [141], in order to avoid to avoid heavy computation, the evaluation is carried out in the following way: the prediction made by the network is taken for the positive item (the next item in the sequence) and 100 items chosen randomly, not in the input sequence. The predictions (for 100 negative items + the positive item) are then sorted according to the values obtained; this represents the final ranking.

²<https://movielens.org>

³<https://nijianmo.github.io/amazon/index.html>

We want to emphasize that while we use multiple positive items during training, this is not done during the evaluation phase. The reason for this decision is that changing the evaluation method could naturally result in our proposed losses appearing better, thus rendering the comparison invalid. By adhering to the traditional evaluation setting, we align ourselves with the evaluation methods used in other works in this field. However, we acknowledge that this places our proposed method at a disadvantage compared to the baseline method for obvious reasons. We are training the model to predict multiple items to increase its robustness, but only one of these items will be used during evaluation. On the other hand, the baseline model focuses solely on a single positive item, the same one used for evaluation, which inherently gives it an advantage. In Section 3.1.6, we will demonstrate how our model still manages to achieve superior results. In addition to the standard metrics, to evaluate the sensitivity of the models in cases of input data perturbations, we utilized the recently introduced metric Rank List Sensitivity (RLS)[206], enabling to compare rankings produced with and without perturbations. RLS is defined as $RLS = \frac{1}{N} \sum_i^N sim(R_{A,i}, R_{B,i})$. where N is the number of samples, sim is a similarity function, $R_{A,i}$ and $R_{B,i}$ are two rankings produced for sample i . In our specific case, A represents the ranking when sample i is unaltered, while B represents the case when the input sequence is perturbed, i.e. items are removed. The similarity (sim) of two rankings R_a and R_b can be calculated using the Jaccard similarity [136], but it does not consider order. On the other hand, Rank-biased Overlap (RBO) is more valuable for a recommendation system as it considers top-ranked items as more significant using specific weighting (see Equation 3.2).

$$JAC(R_A, R_B) = \frac{|R_A \cap R_B|}{|R_A \cup R_B|} \quad RBO(R_a, R_b) = (1 - p) \sum_{i=1}^k p^{i-1} \frac{|R_A[1:i] \cap R_B[1:i]|}{i} \quad (3.2)$$

Hyperparameter Optimization

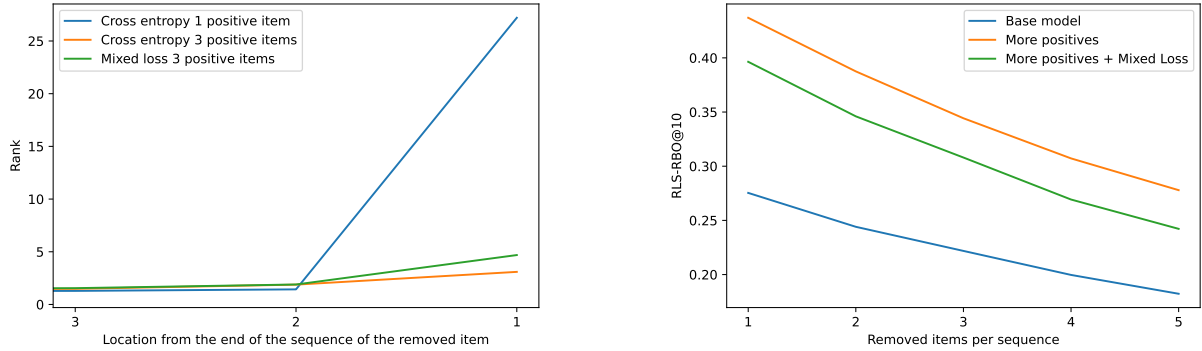
The hyper-parameters to be optimised are the number of positive items to be used, the number of negative items to be used and the Mixed Loss parameters. The number of positive items N_{pos} and negative items N_{neg} varies in the set $\{1, 3, 10\}$, and the Mixed Loss parameters λ_1 and λ_2 in the set $\{1, 10^{-1}, \dots, 10^{-5}\}$.

Implementation

All code is written in Python 3. In particular, with Pytorch and Pytorch Lightning.

3.1.6 Results

In this section, we present experimental results showing the strong reliance of sequential recommender models on the last items in the sequence as well as the performance of the proposed training method to mitigate this effect.



(a) Rank of the previous top-ranked item when removing an item from the input sequence in a specific position

(b) Rank List Sensitivity using Rank-Biased Overlap@10 for SASRec Model on MovieLens-1M dataset with different number of missing items

Figure 3.2: The effect of removing the last items in a sequence with three training methods

Last Items Importance

Figure 3.2a visualizes the effect of removing an item at different positions in the sequence on the model outputs, using the SASRec model and the MovieLens-1M dataset, has on the ranking of the top item. We identify this item with the term *previous top-ranked item*: that item which, prior to the input data perturbation, was at the top of the ranking. In the case of the base model, removing the last item can push the previously top-ranked item by over 25 positions on average. While SASRec is trained using dropout, this does not seem to be sufficient to make it robust to missing data. In contrast, when the model is trained with more positive items, the removal of the last item results in a significantly lower drop in the ranking of the previous top item: 5 positions or less. We also observe that the difference between the different models becomes less pronounced as we move towards the earlier items in the sequence. These results demonstrate that incorporating more positive items in the training process and using our proposed Mixed Loss can help to mitigate the impact of missing data and improve the robustness of Sequential Recommender Systems.

Performance of Different Training Methods in Cases of Missing Last Items

NDCG@10 score is used to gauge the impact of the modified training method on the performance of the models. Figure 3.3 provides a visualization of the results. The results are also expressed integrally in Tables 3.2, 3.3 and 3.4.

A Clear Advantage in Handling Missing Data

One striking observation is that the models trained with more positive items and the Mixed Loss consistently outperform the base model when it comes to dealing with missing data. Although the base model performs slightly better in the absence of missing data, the new models are able to sustain their performance even as the number of missing items increases. This is especially evident in the case of the Amazon Beauty dataset, where the new training method is able to maintain acceptable performance as the missing data becomes more prominent.

As mentioned in Section 3.1.5, the slight predominance of the base model in the absence of missing data is expected because the evaluation setting naturally favors the base model: both the

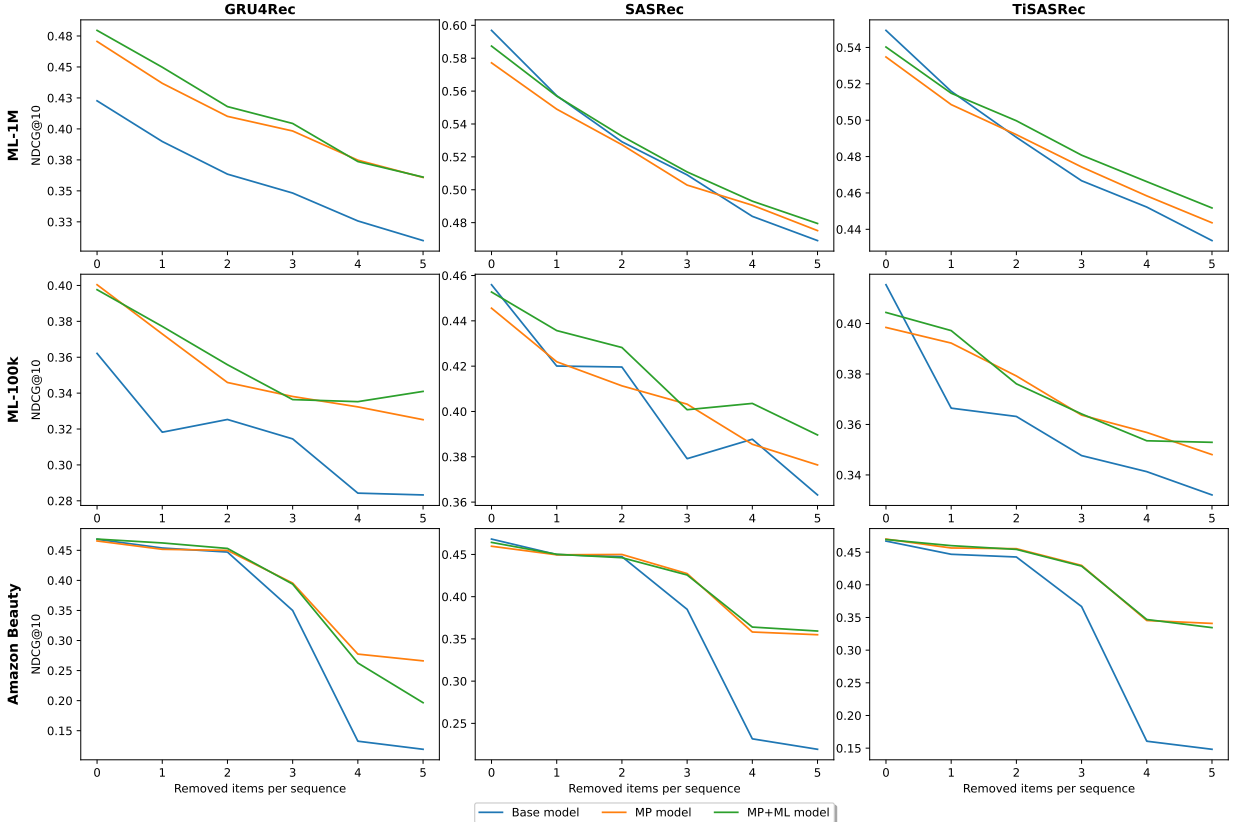


Figure 3.3: NDCG@10 with different number of missing items for each model-dataset pair

Table 3.2: Results in terms of ranking evaluation (NDCG@10 and HR@10) and robustness metrics (RLS with Jaccard or RBO) for GRU4Rec model and the considered datasets, varying the number of items removed from the end of the sequence. To assist visualization leading zeroes are removed.

Dataset	Missing Items	NDCG@10			HR@10			RLS-JAC@10			RLS-RBO@10		
		Base	MP	MP+ML	Base	MP	MP+ML	Base	MP	MP+ML	Base	MP	MP+ML
ML-1M	0	.4227	.4706	.4795	.6618	.7098	.7134	—	—	—	—	—	—
	1	.3898	.4368	.4498	.6356	.6843	.6873	.0489	.0867	.0794	.0313	.0550	.0489
	2	.3635	.4101	.4180	.6162	.6579	.6606	.0442	.0752	.0676	.0293	.0475	.0403
	3	.3482	.3983	.4044	.5881	.6452	.6475	.0392	.0672	.0604	.0252	.0415	.0358
	4	.3257	.3748	.3737	.5642	.6214	.6167	.0376	.0628	.0552	.0241	.0398	.0334
5	.3099	.3608	.3611	.5440	.6081	.6038	.0347	.0568	.0502	.0219	.0346	.0298	
ML-100k	0	.3621	.4004	.3976	.6182	.6607	.6713	—	—	—	—	—	—
	1	.3182	.3730	.3772	.5705	.6288	.6490	.1216	.2284	.2292	.0809	.1689	.1680
	2	.3253	.3459	.3558	.5779	.6161	.6193	.1181	.2123	.2214	.0756	.1547	.1606
	3	.3145	.3381	.3364	.5493	.5875	.5938	.1102	.2047	.2060	.0712	.1524	.1517
	4	.2843	.3323	.3352	.5154	.5907	.5843	.1084	.1947	.1959	.0689	.1433	.1435
5	.2833	.3252	.3410	.5080	.5663	.5832	.0967	.1826	.1801	.0624	.1304	.1312	
Amazon Beauty	0	.4683	.4656	.4687	.5114	.5077	.5060	—	—	—	—	—	—
	1	.4539	.4517	.4623	.5072	.4969	.5056	.7110	.6633	.6780	.4987	.4858	.4676
	2	.4471	.4499	.4531	.5027	.4990	.5064	.6472	.5829	.6100	.4793	.4550	.4386
	3	.3500	.3955	.3938	.4059	.4688	.4737	.4059	.4026	.3827	.2947	.3185	.2915
	4	.1325	.2774	.2626	.1849	.3620	.3562	.2413	.3298	.2390	.1400	.2017	.1783
5	.1191	.2662	.1966	.1676	.3442	.2892	.1583	.2092	.1164	.0773	.1122	.0930	

loss function and the evaluation technique consider only a single item. We emphasize the significance that our model, trained in a manner that deviates slightly from the traditional evaluation setting, is able to retain minimal performance loss in the same setting while gaining robustness to missing data.

Table 3.3: Results in terms of ranking evaluation (NDCG@10 and HR@10) and robustness metrics (RLS with Jaccard or RBO) for SASRec model and the considered datasets, varying the number of items removed from the end of the sequence. To assist visualization leading zeroes are removed.

Dataset	Missing Items	NDCG@10			HR@10			RLS-JAC@10			RLS-RBO@10		
		Base	MP	MP+ML	Base	MP	MP+ML	Base	MP	MP+ML	Base	MP	MP+ML
ML-1M	0	.5969	.5772	.5874	.8222	.8142	.8207	—	—	—	—	—	—
	1	.5572	.5490	.5570	.7925	.7925	.7962	.4116	.6157	.5584	.2754	.4370	.3964
	2	.5292	.5274	.5326	.7768	.7748	.7783	.3625	.5348	.4849	.2441	.3875	.3460
	3	.5090	.5028	.5108	.7647	.7594	.7634	.3276	.4731	.4287	.2218	.3443	.3080
	4	.4838	.4906	.4931	.7452	.7425	.7520	.2933	.4215	.3741	.1997	.3072	.2693
	5	.4691	.4752	.4795	.7316	.7344	.7411	.2676	.3778	.3368	.1823	.2779	.2422
ML-100k	0	.4559	.4456	.4527	.7349	.7349	.7455	—	—	—	—	—	—
	1	.4200	.4219	.4357	.7243	.6988	.7232	.2734	.6457	.5411	.1668	.4636	.3732
	2	.4196	.4113	.4282	.6999	.6978	.7179	.2565	.6217	.5179	.1599	.4458	.3615
	3	.3792	.4032	.4008	.6607	.6861	.6935	.2441	.5916	.4920	.1508	.4268	.3448
	4	.3878	.3855	.4036	.6755	.6734	.6925	.2350	.5565	.4707	.1450	.4083	.3333
	5	.3632	.3764	.3896	.6511	.6670	.6797	.2226	.5261	.4369	.1409	.3869	.3106
Amazon Beauty	0	.4682	.4597	.4643	.5038	.5075	.5067	—	—	—	—	—	—
	1	.4499	.4496	.4502	.4959	.5032	.5037	.5730	.6087	.6164	.4076	.4332	.4238
	2	.4474	.4500	.4461	.5034	.5053	.5077	.5123	.5314	.5375	.3827	.4152	.3954
	3	.3850	.4273	.4257	.4510	.5108	.5086	.3967	.4531	.4442	.3043	.3585	.3335
	4	.2317	.3582	.3640	.3055	.4759	.4879	.2987	.3934	.3800	.2193	.2909	.2818
	5	.2193	.3549	.3593	.2931	.4763	.4878	.2579	.3718	.3602	.1806	.2522	.2563

Table 3.4: Results in terms of ranking evaluation (NDCG@10 and HR@10) and robustness metrics (RLS with Jaccard or RBO) for TiSASRec model and the considered datasets, varying the number of items removed from the end of the sequence. To assist visualization leading zeroes are removed.

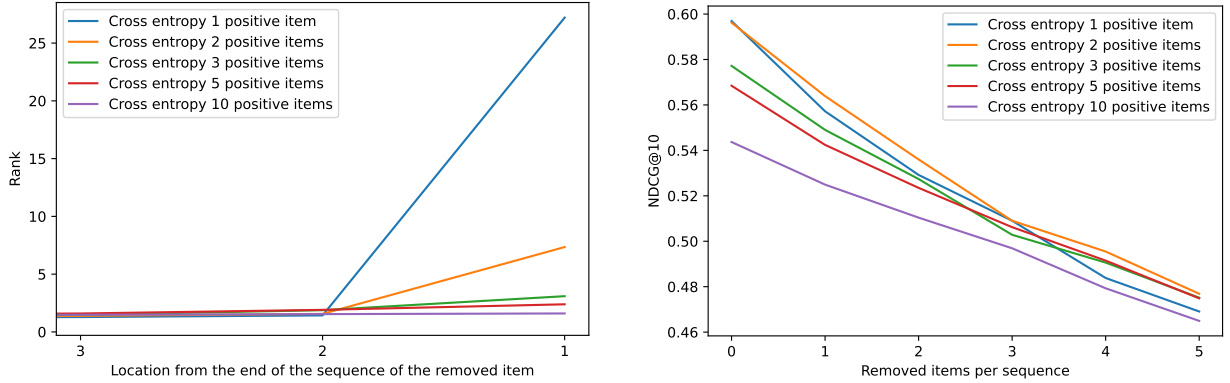
Dataset	Missing Items	NDCG@10			HR@10			RLS-JAC@10			RLS-RBO@10		
		Base	MP	MP+ML	Base	MP	MP+ML	Base	MP	MP+ML	Base	MP	MP+ML
ML-1M	0	.5494	.5348	.5402	.7823	.7760	.7773	—	—	—	—	—	—
	1	.5159	.5086	.5148	.7523	.7522	.7550	.3622	.5714	.5605	.2373	.4069	.3946
	2	.4906	.4920	.4997	.7387	.7386	.7444	.3070	.4968	.4769	.2000	.3592	.3396
	3	.4667	.4743	.4808	.7166	.7242	.7311	.2685	.4401	.4202	.1747	.3215	.3028
	4	.4522	.4584	.4662	.7043	.7144	.7194	.2348	.3891	.3669	.1536	.2880	.2649
	5	.4338	.4436	.4517	.6889	.7070	.7126	.2072	.3488	.3270	.1347	.2590	.2369
ML-100k	0	.4154	.3984	.4044	.6935	.6702	.6670	—	—	—	—	—	—
	1	.3665	.3922	.3972	.6394	.6426	.6490	.2704	.5377	.5238	.1638	.3869	.3708
	2	.3632	.3792	.3761	.6108	.6225	.6246	.2406	.4855	.4735	.1424	.3547	.3394
	3	.3477	.3637	.3642	.6182	.6172	.6182	.2194	.4546	.4418	.1292	.3360	.3186
	4	.3413	.3568	.3535	.5938	.6161	.6151	.2020	.4227	.4113	.1194	.3145	.2959
	5	.3321	.3481	.3529	.5822	.6034	.6076	.1965	.4004	.3858	.1163	.3021	.2819
Amazon Beauty	0	.4670	.4700	.4693	.4994	.5106	.5077	—	—	—	—	—	—
	1	.4467	.4564	.4600	.4911	.5077	.5056	.5819	.6580	.6559	.3931	.4641	.4575
	2	.4426	.4553	.4542	.4944	.5101	.5101	.5402	.6353	.6403	.3765	.4598	.4563
	3	.3667	.4297	.4286	.4241	.5097	.5130	.4135	.5438	.5512	.3096	.3784	.3676
	4	.1606	.3453	.3469	.2122	.4886	.4878	.2567	.4031	.3969	.2155	.2892	.2655
	5	.1483	.3409	.3345	.1953	.4882	.4866	.1926	.3183	.3013	.1831	.2338	.2015

Length of sequences

It is worth noting that the average sequence length of the three datasets is vastly different (see Section 3.1.5). The results in Figure 3.3 indicate that the impact of missing data is much less severe for datasets with longer sequences, such as ML-1M. The model trained with the classic training method is even able to compensate for this deficiency, particularly in the case of SASRec. However, as the average sequence length decreases, such as in the ML-100k dataset, the robustness to missing data seems to decline rapidly, and the difference between the models becomes more pronounced when the number of missing items increases. This trend is especially evident in the Amazon Beauty dataset, where the difference between the models is particularly noticeable when

the number of missing items is higher than 2, probably because the average length of the sequences for this dataset is 5.

Rank List Stability



(a) Rank of the previous top-ranked item when removing an item from the input sequence in a specific position

(b) NDCG@10 for SASRec Model on MovieLens-1M dataset with different number of missing items

Figure 3.4: Study on the Number of Positives

The robustness of the models in the face of item removal at the end of the input sequence is illustrated through the Rank List Stability with Rank-biased Overlap metric in Figure 3.2b. It is evident that the new models exhibit higher stability, with Cross-Entropy with more positive items proving to be even more robust than the Mixed Loss model. We observed a similar trend for all datasets and models, so only one plot is presented; further results can be found in the additional repository. While the multiple positive model (MP) provides in most cases higher performance in the Rank List Stability metrics compared to the model with multiple positive and the mixed loss (MP+ML), it is worth noting that MP+ML provides higher performance on the HR@10 and NDCG@10. This can be explained by the fact that MP is not optimized using the ranks of the positive items as done by the mixed loss (MP+ML model). However, for precisely the same reason, MP benefits from higher stability.

More specifically, both models are trained to predict, at time t and for a given input sequence, N_{pos} positive items, specifically $[p_t, p_{t+1}, \dots, p_{t+N_{pos}}]$. However, the MP model is trained with a loss function that does not consider the order of the positive items: the same sequence in reverse order would yield the same loss value. As discussed in Section 3.1.5, in the classic evaluation setting, only p_t is used during evaluation. If the loss function treats p_t equally important as the other positive items $[p_{t+1}, \dots, p_{t+N_{pos}}]$, it is more likely to be ranked lower, thus reducing metrics such as NDCG and Recall. On the other hand, the model using the Mixed Loss, which aims to prioritize the position of p_t at the top of the ranking, has an advantage in achieving higher metrics in this regard.

Study on the Number of Positives

To understand the impact of the number of positive items used for training, experiments were performed using different numbers of positive items for just one model, SasRec, and one dataset, MovieLens-1M, due to the computational time required. As seen in Figure 3.4a, as the number of positive items increases, the change in ranking for previous top-ranked items decreases significantly.

However, Figure 3.4b shows that the performance in the absence of missing data degrades as the number of positive items increases. This trend begins to change as the number of missing items increases, and the gap between the new models and the base model narrows, with the latter's performance deteriorating more.

3.1.7 Implications of the Research Findings

The findings of this study hold both theoretical and practical implications that contribute to the advancement of sequential recommender systems and their application in real-world scenarios. By addressing the specific challenges posed by missing input data, our research offers a novel perspective on enhancing the robustness and reliability of these systems.

Theoretical Implications

1. **Uncovering Last-Item Dependence:** Our research uncovers the strong reliance of sequential recommender systems on the last items in the input sequence. This revelation contributes to a deeper understanding of the dynamics within these systems, emphasizing the need for strategies that can mitigate the performance degradation caused by missing items.
2. **New Training Paradigm:** The introduction of a training approach that anticipates data loss and simulates prediction of multiple future items presents a paradigm shift in the methodology for handling missing input data. This approach establishes a theoretical foundation for designing more resilient recommender systems.

Practical Implications

1. **Real-World Data Challenges:** In real-world scenarios, complete user action sequences are often not available due to various constraints. Our research highlights the practical significance of addressing this data scarcity and provides a concrete solution to mitigate the negative effects of missing items, improving the usability of recommender systems.
2. **Enhanced System Resilience:** The proposed training method significantly improves the performance of sequential recommender systems when faced with missing items. This directly translates into a more reliable and user-centric experience, thus benefiting various domains, such as e-commerce, content recommendation, and personalized services.
3. **Impact on User Satisfaction:** The performance enhancement demonstrated by our approach can lead to improved user satisfaction by providing more accurate and relevant recommendations, even when there are gaps in the available data. This practical outcome can foster greater user engagement and loyalty.
4. **General Applicability:** The effectiveness of our method across various datasets and recommender models underscores its general applicability. This widens its potential adoption and impact, making it a valuable tool for researchers and practitioners alike.

3.1.8 Discussion and Conclusions

Our findings show that the last items in a sequence have a significant impact on the predictions of sequential recommenders, and their removal results in unstable rankings. However, by incorporating multiple future items in the training process, model robustness can be improved. Our results demonstrate that the proposed training methods improve rankings stability (RLS metric) and performance (HR and NDCG) on various popular sequential recommender models (SasRec[141], TiSasRec[165], and GRU4Rec[124]) and datasets. In contrast, the performance without missing data is not noticeably affected but even improves for specific models/datasets. Using more positive items with Cross-Entropy loss improves robustness of sequential recommenders to removal of elements at the end of the input sequence. However, increasing the number of future items excessively can lead to stability increase at the cost of decreased performance. Mixed Loss, combining Cross-Entropy with Margin Loss, can prioritize the next item over other positives. Our method opens up opportunities for further research in the field. Future work may focus on the development of a loss function that balances performance and robustness as the number of positive items increases, as well as modifying the method for models that use bi-directional connections (e.g., [283]). Moreover, our proposal is easily extendable to other approaches, as it is solely tied to a different training method and not to a specific architecture. To summarize, our work represents a step forward in improving the robustness of sequential recommender models. We demonstrate the strong influence of the last items in a sequence and the effectiveness of our method in mitigating the impact of missing data. Overall, we expect that our findings and proposed methods will be a valuable tool in the field of sequential recommender systems.

3.2 Integrating Item Relevance in Training Loss for Sequential Recommender Systems

Sequential Recommender Systems (SRSs) are a popular type of recommender system that leverages user history to predict the next item of interest. However, the presence of noise in user interactions, stemming from account sharing, inconsistent preferences, or accidental clicks, can significantly impact the robustness and performance of SRSs, particularly when the entire item set to be predicted is noisy. This situation is more prevalent when only one item is used to train and evaluate the SRSs. To tackle this challenge, we propose a novel approach that addresses the issue of noise in SRSs. First, we propose a sequential multi-relevant future items training objective, leveraging a loss function aware of item relevance, thereby enhancing their robustness against noise in the training data. Additionally, to mitigate the impact of noise at evaluation time, we propose multi-relevant future items evaluation (MRFI-evaluation), aiming to improve overall performance. Our relevance-aware models obtain an improvement of 1.58% of NDCG@10 and 0.96% in terms of HR@10 in the traditional evaluation protocol, the one which utilizes one relevant future item. In the MRFI-evaluation protocol, using multiple future items, the improvement is 2.82% of NDCG@10 and 0.64% of HR@10 w.r.t the best baseline model.

3.2.1 Introduction

Recommender systems have become an integral part of our daily lives [344], as they assist us in making decisions by suggesting items we might like based on our preferences and behaviors [239]. In recent years, Sequential Recommender Systems (SRSs) have emerged as a promising solution to improve the accuracy and relevance of recommendations by incorporating the temporal aspect of user-item interactions. These systems aim to predict the next item a user is likely to interact with based on their past interactions, taking into account the sequence of actions and their temporal order [225].

Traditional evaluation metrics, such as Hit Rate (HR) and Normalized Discounted Cumulative Gain (NDCG), fail to capture the complexity of sequential data when evaluated by measuring how well they predict a single future item [141, 165, 283]. The reason is that the hypothesis of a single "relevant" item might not reflect the users' true intentions or preferences, particularly when considering noisy sequences in real-world scenarios, as argued by [27, 310]. For example, users accidentally clicking on items, or performing multiple actions quickly, greatly affects the evaluation results, as shown by [112, 206]. Hence, we propose a novel and groundbreaking approach to eval and training of SRSs: we depart from the single-relevant item approach typically considered in the current literature by leveraging multiple future items to account for noise in the sequences. Assuming that only the next item in the database is relevant, disregarding the potential relevance of other future items, is not only unrealistic but also overlooks valuable information contained in the entire dataset. We show that by considering multiple relevant future items during evaluation, the impact of noisy items is reduced, and models that anticipate users' future preferences beyond immediate predictions are

incentivized.

Our contributions are two-fold: (i) we propose a sequential multi-relevant future items training objective, defined by a loss function that takes into account the item relevance; (ii) we present a novel evaluation setting called Multi-Relevant Future Items Evaluation (MRFI-evaluation), which aims to mitigate the impact of noise during the evaluation of RecSys models. These contributions collectively address the challenges of noise in both the training and evaluation stages of RecSys, to improve their overall performance and robustness against noise.

Our experiments show that the proposed method provides a more accurate and robust solution for evaluating and training SRSs. We conducted experiments on four datasets typically used in this research domain, using SASRec [141] and TiSASRec [165], two widely cited SRSs. Our experiments show that a model trained with the proposed loss yields state-of-the-art results, in terms of NDCG@10 and Recall@10 scores, in both MRFI and traditional evaluation protocols.

3.2.2 Related Work

Sequential Recommender Systems

Sequential Recommender Systems (SRSs) are a class of recommender systems that personalizes recommendations to users based on their historical interactions with items in a sequence [309], capturing its temporal dynamics. SRSs have received considerable attention in the research community in recent years [344] and have been applied in various domains [344], including movies [104, 117, 220], music [251, 252], and e-commerce [133, 250]. Various techniques have been developed to model the temporal dependencies in the sequences, including Markov Chain models, Recurrent Neural Networks (RNNs), and Attention mechanisms. Markov Chain models are a type of probabilistic model that assumes the future state of a sequence only depends on the current state; for this reason, they struggle to capture complex dependencies in long-term sequences [89, 90]. RNNs are a type of neural network architecture that can capture the long-term dependencies in sequential data. They have shown great potential in modeling sequential data, and they have been used to develop various SRSs, such as session-based recommenders [123–125, 164, 170], context-aware recommenders [4, 157, 329], and graph neural networks [45, 82, 223]. Attention mechanisms have recently gained attention in SRSs due to their ability to dynamically weigh the importance of different parts of the sequence [141]. By doing so, attention mechanisms can better capture the important features in the sequence and improve the prediction accuracy [312, 352].

Evaluating Sequential Recommender Systems

Evaluating Sequential Recommender Systems (SRSs) has been a topic of great interest in recent years. Both [282] and [350] examined common data splitting methods for SRSs and discussed why commonly used evaluation methods are ill-defined, suggesting appropriate offline evaluation for SRSs. In particular, they showed that existing evaluation protocols do not consider the temporal dynamics of user behavior, which can affect the accuracy of the recommendations. In [138], it is shown that the current evaluation protocols for SRSs can lead to data leakage, where the model learns information from the test data that is not available during training. They address the problem by proposing an evaluation methodology that considers the global timeline of data samples

in the evaluation of SRSs. A metric called Rank List Sensivity (RLS) is introduced in [206] to evaluate the discrepancy between two rankings, so to evaluate models' sensitivity with respect to training data. Finally, [19] presented an evaluation methodology specifically designed to evaluate the precision of algorithms for the Search Shortcut Problem. This metric considers item relevance, which allows an effective evaluation.

3.2.3 Methodology

Current Evaluation Protocol

In line with previous research employing SRSs [141, 165, 283], the current evaluation method involves shuffling one positive item (the next item in the sequence) with 100 random negative items not part of the input sequence. These items are then ranked based on their relevance scores determined by the model. The resulting rank is typically evaluated with metrics such as Normalized Discounted Cumulative Gain (NDCG) and Hit Rate (HR), using various cut-offs, typically 10.

Problems with the current evaluation protocol

The current evaluation protocol assumes only one item is relevant to a user, which may not be true in real-world scenarios with multiple relevant interactions. Users' history can be noisy due to account sharing, inconsistent preferences, or accidental clicks, as discussed in [310]. For instance, in e-commerce, many clicks don't lead to purchases, and some receive negative reviews. Evaluating a model on one noisy item negatively impacts its performance, and this aspect is disregarded in the current evaluation protocol. Furthermore, assuming that all other future items, except the next one, are irrelevant is both strong and unrealistic, disregarding valuable available data.

Multi-Relevant Future Items: a new evaluation protocol

To address the aforementioned problem, we propose a new evaluation protocol for SRSs called Multi-Relevant Future Items (MRFI). In MRFI, we make the Assumption 1, that a good ranking should not only contain the single future next item in the sequence, but the whole sequence of future items in the correct order. In this way, when the test set presents noisy items, the effect is mitigated as more items are considered during evaluation. Thus, models that disregard ranking noisy items are less penalized. Moreover, this protocol rewards models that anticipate user preferences beyond immediate predictions.

Assumption 1. *Given a user u and its ordered interactions' sequence $[I_1, I_2, \dots, I_i]$, the ideal ranking of length K is the sequence of future items $[I_{i+1}, I_{i+2}, \dots, I_{i+K}]$ arranged user's interaction temporal order.*

To evaluate the performance of SRSs under Assumption 1, we use traditional evaluation metrics for sequential recommendation models such as NDCG and HR. To have multiple future items per evaluation, we split the user's history differently. Given a sequence $[I_1, I_2, \dots, I_L]$, in the traditional evaluation protocol, only item I_L is reserved for testing. In our proposed evaluation protocol, the sequence $[I_1, I_2, \dots, I_{L-K}]$ is allocated for training the model, while the sequence $[I_{L-K+1}, I_{L-K+2}, \dots, I_L]$ is used for testing purposes.

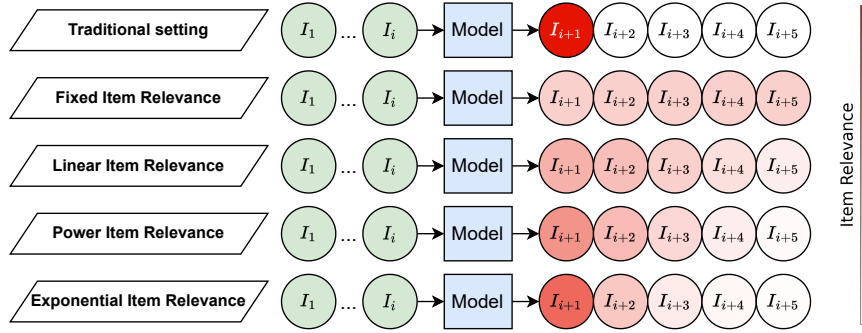


Figure 3.5: A visualization of how the various loss scaling strategy weigh the item relevance.

Item Relevance

The MRFI evaluation protocol requires higher ranking capabilities than the traditional evaluation protocol. The original evaluation protocol evaluates the ability to rank a single item and treat all other items as irrelevant to the user. Conversely, in the new MRFI evaluation protocol, it is important to define item relevance to give importance to multiple future items and scale their importance according to their position in the sequence.

Definition 3.2.3.1. Given a ranking $[I_1, I_2, \dots, I_K]$ of length K , we define the item relevance function $r: \mathbb{N} \rightarrow [0, 1]$

To compare item relevance on sequences of different lengths K , we define that the item relevance r sums to one.

Definition 3.2.3.2. Given a ranking $[I_1, I_2, \dots, I_K]$ of length K , item relevance r must satisfy $\sum_{i=1}^K r(i) = 1$

Item relevance should assign lower importance to interactions further in the future. We establish that the relevance of an item at time t cannot be lower than the relevance at time $t + 1$.

Definition 3.2.3.3. Given a ranking $[I_1, I_2, \dots, I_K]$ of length K , r must satisfy $r(i+1) \leq r(i) \quad \forall i \in \{1, 2, \dots, K\}$

Our approach to item relevance is inspired by [19] who proposed a similarity function, which takes into account the item relevance, to evaluate query recommendation using collaborative filtering. They suggested four different functions for item relevance: $r(i) = 1$, $r(i) = K - i$, $r(i) = (K - i)^2$, and $r(i) = e^{K-i}$ with $i \in \{1, 2, \dots, K\}$. We refer to these functions respectively as *Fixed*, *Linear*, *Power*, and *Exponential*. The first assigns equal relevance to all items, while the others assign higher relevance to the next items in the sequence at the expense of the more distant ones. We show a visualization of these functions in Figure 3.5. The functions can be easily normalized to comply with the Definition 3.2.3.2. It should be noted that our evaluation protocol generalizes the traditional one because we can revert to it by setting maximum importance to the next item and zero importance for all other future items.

Relevance-based Loss

To explicitly integrate the item relevance in the training of the SRSs, we propose a Relevance-based loss, a modification of the Binary Cross-Entropy, that leverages multiple future positive items at

training time. [286] propose a similar formulation that assigns equal relevance to all future items (equivalent to our Fixed formulation). However, by assigning equal importance to all items, the model ignores the natural order of interactions, making this strategy unsuitable for some tasks. To address this limitation, we propose a Sequential Multi Future Item Training regime that considers the sequential nature of the items integrating their relevance, which translates to a Relevance-based loss:

$$\ell(\vec{x} \mid pos, neg, r) = - \sum_{i=1}^{pos} \log((\vec{x}_{pos})_i) r(pos - i + 1) - \sum_{i=1}^{neg} \log(1 - (\vec{x}_{neg})_i) \quad (3.3)$$

where pos and neg represent the number of positive and negative items, respectively. \vec{x} is the score given by the model to each item, while \vec{x}_{pos} and \vec{x}_{neg} represent respectively the subset of \vec{x} containing only the positive and negative items and r is the item relevance function defined in Section 3.2.3. In Equation 3.3, we weigh the loss of each item i by its relevance score $r(i)$. Doing so encourages the model to focus more on the relevant items while learning.

An advantage of this loss lies in its decoupling from the evaluation method used. Even though we introduce an evaluation protocol that incorporates multiple future items, it doesn't necessarily have to be used only with a loss that considers multiple future items. Similarly, the loss can be used not only for this specific type of evaluation. This means that the loss can be easily incorporated into any model that uses a loss function, and the model can be tested using the traditional evaluation protocol, potentially leading to improved results, as demonstrated in our findings.

3.2.4 Experimental Setup

We assess our techniques using four datasets derived from real-world use cases: MovieLens[118] 1M and 100k, and Foursquare [328] Tokyo (TKY) and New York City (NYC). We select the Self-Attentive Sequential Recommendation (SASRec) [141] and the Time Interval Aware Self-Attention for Sequential Recommendation TiSASRec [165] models for our experiment, as both have consistently demonstrated exceptional performance across multiple benchmarks and garnered significant recognition in the literature. For fairness, we retain the original hyper-parameters tuned on the baseline model. This decision should not be perceived as a drawback but rather as an untapped potential inherent to our proposed methodology: the traditional loss model has already been fine-tuned for optimum performance, while our loss model has yet to be exploited to its fullest potential. The code is available in our Github Repository (<https://github.com/andreabac3/Integrating-Item-Relevance-in-Training-Loss-for-Sequential-Recommender-Systems>).

3.2.5 Results

In this Section, we present the results by answering the following three research questions:

- **RQ1:** Can we mitigate the impact of noise by introducing an alternative to the single-item evaluation protocol?
- **RQ2:** Can item relevance improve an SRS's performance when incorporated into the training mechanism?

- **RQ3:** What is the impact of the number of future items on evaluation metrics and model training performance?

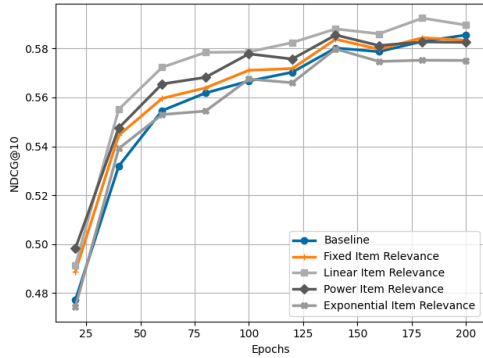
During our experiments, we vary (i) the number of evaluation positives; (ii) the number of training positives; (iii) the item relevance function. From here on, we will denote by *our* the models trained with our proposed training loss. We refer to the original model trained on a single future relevant item as Baseline, while with *baseline*, we indicate both Baseline and Fixed models. To validate the effectiveness of our method compared to Baseline and Fixed, we employed the one-sided Wilcoxon signed-rank test [320] with Bonferroni correction [32] and a significance level of 0.05. The Wilcoxon test was chosen as a non-parametric alternative to the paired t-test due to the non-normal distribution of the data.

		ML-1M		ML-100k		Foursquare NYC		Foursquare TKY		
Model		NDCG	HR	NDCG	HR	NDCG	HR	NDCG	HR	
SASRec	Traditional Evaluation Protocol	Baseline [141] (a)	0.5989	0.8273	0.4514	0.7359	0.6706	<u>0.7673</u>	0.7274	0.8029
		Fixed [286] (b)	0.5983	0.8265	0.4520	<u>0.7423</u>	0.6911	0.7655	0.7358	0.8042
		Linear	0.6112^{ab}	0.8326^b	0.4680^{ab}	0.7434	0.6843 ^a	0.7608	0.7422^{ab}	0.8190^{ab}
		Power	<u>0.6061^{ab}</u>	<u>0.8296</u>	<u>0.4614^b</u>	0.7359	0.6869 ^a	0.7608	<u>0.7385^a</u>	<u>0.8090</u>
		Exponential	0.5918	0.8205	0.4493	0.7253	<u>0.6894^a</u>	0.7701	0.7369 ^a	0.8068
	MRFI Evaluation Protocol	Baseline [141](a)	0.3482	0.6249	0.2047	0.5259	0.3117	0.6772	0.3757	0.7382
		Fixed [286](b)	0.3530	<u>0.6357</u>	0.2091	<u>0.5462</u>	0.3167	0.6972	<u>0.3834</u>	<u>0.7552</u>
		Linear	0.3577^{ab}	0.6416^{ab}	<u>0.2177^{ab}</u>	0.5434 ^a	0.3245^{ab}	0.6942 ^a	0.3728	0.7504 ^a
		Power	<u>0.3575^{ab}</u>	0.6353 ^a	0.2237^{ab}	0.5506^a	0.3175 ^a	0.6865 ^a	0.3796	0.7524 ^a
		Exponential	0.3410	0.6312 ^a	0.2051	0.5299	<u>0.3197^a</u>	<u>0.6953^a</u>	0.3846^a	0.7558^a
TiSASRec	Traditional Evaluation Protocol	Baseline [141] (a)	<u>0.5681</u>	0.8012	0.4247	<u>0.7190</u>	0.6190	0.6999	0.6643	0.7475
		Fixed [286] (b)	0.5597	<u>0.8030</u>	0.4223	0.7031	<u>0.6319</u>	<u>0.7018</u>	0.6719	0.7414
		Linear	0.5743^{ab}	0.8048	<u>0.4293</u>	0.7137	0.6301 ^a	0.6934	0.6758^{ab}	<u>0.7423</u>
		Power	0.5655 ^b	0.7990	0.4319^b	0.7147	0.6323^a	0.7073	0.6675	0.7340
		Exponential	0.5609	0.7978	0.4272	0.7211^b	0.6282 ^a	0.6944	<u>0.6724^a</u>	0.7375
	MRFI Evaluation Protocol	Baseline [141] (a)	0.3181	0.6017	0.1935	0.5153	0.2535	0.6189	0.3154	0.6952
		Fixed [286] (b)	<u>0.3253</u>	<u>0.6140</u>	<u>0.1974</u>	<u>0.5226</u>	0.2555	0.6235	0.3114	0.6891
		Linear	0.3321^{ab}	0.6194^{ab}	0.1948	0.5259^a	0.2523	0.6229	0.3202^{ab}	<u>0.6986^{ab}</u>
		Power	0.3206	0.6111 ^a	0.2007^a	0.5221 ^a	<u>0.2581</u>	0.6264^a	<u>0.3188^a</u>	0.6984 ^{ab}
		Exponential	0.3125	0.6062 ^a	0.1936	0.5212	0.2603^a	<u>0.6259^a</u>	0.3176	0.6992^{ab}

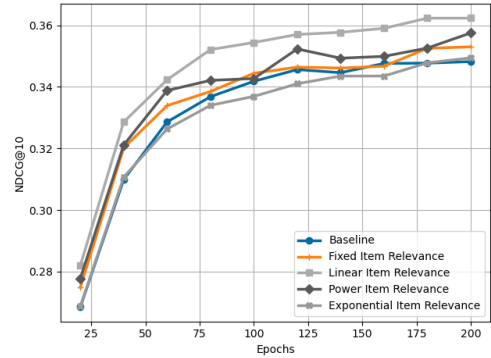
Table 3.5: Table shows the values of the NDCG@10 and HR@10 metrics that the ten models obtain on the four datasets in the *traditional evaluation protocol* and *new evaluation protocol*. **Bold** shows the best result for each column, underlined the second best. Superscripts ^a and ^b indicate that the result is statistically significantly better than Baseline or Fixed, respectively.

RQ1.

As explained in Section 3.2.3, taking into account fewer relevant items in evaluating the results might not suffice to pick the best model. Consider, for example, a simple scenario where the user u will interact with the items I_1 , I_2 , and I_3 in the future. Model A outputs the ranking $[I_{50}, I_1, I_{100}]$, while model B outputs $[I_2, I_1, I_3]$. Using only I_1 as a relevant item makes both models appear to perform similarly. Conversely, considering all three future items, model B is significantly superior to model A . Having provided a theoretical rationale for why the new evaluation setting is superior to the traditional one, which extends beyond our results, we can state that any model, trained in any way, can be tested using this evaluation protocol. This could reveal that models that perform poorly in the traditional evaluation protocol, instead perform well in the new one, possibly



(a) Traditional evaluation protocol



(b) MRFI evaluation protocol

Figure 3.6: NDCG@10 at varying training epochs for all 5 models on the MovieLens 1M dataset

subverting current rankings of models and approaches.

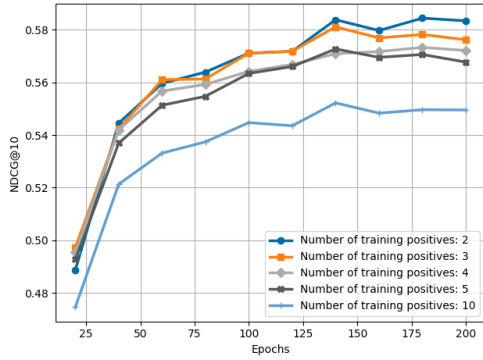
Table 3.5 compares various models in both the traditional and MRFI evaluation protocol, including the two baselines and the models trained with our proposed item relevance-based loss. Our models exhibit superior performance in the traditional evaluation protocol, although some differences are less prominent. Conversely, the MRFI evaluation protocol reveals more pronounced disparities in performance, enabling a better identification of a superior model. Hence, the MRFI evaluation protocol is more robust to the noise introduced in 3.2.3 and can better identify performance differences between models and reward the model with the best ranking performance as evaluated by means of NDCG and HR.

RQ2.

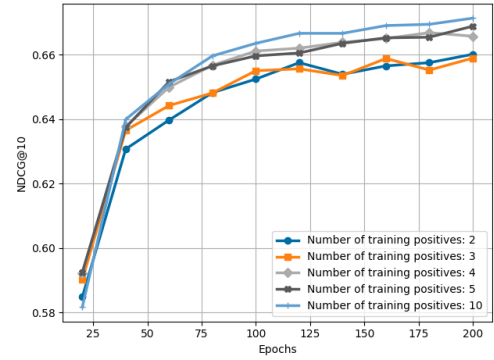
As seen from Table 3.5, in the traditional and MRFI evaluation protocol, models that integrate the item relevance are better (statistically significantly) or on-par (not statistically significant) than the baselines for at least one of the metrics, regardless of the tested architecture. Figure 3.6 shows the performance using NDCG@10 of SASRec model for the MovieLens-1M dataset for both the traditional and MRFI evaluation protocol. We have decided to present this combination because it seems to be the prevailing choice in the literature. We can see that the Baseline, which does not consider multiple future items during training, has a consistently slower convergence rate in both evaluation protocols than the item relevance-aware models. Instead, Linear item relevance obtains the fastest convergence. This shows that the item relevance loss allows the training of better models that yield higher performances in both evaluation protocols. To summarize, our relevance-aware models obtain an improvement of 1.58% of NDCG@10 and 0.96% in the traditional evaluation protocol, while in the new evaluation protocol, the improvement is 2.82% of NDCG@10 and 0.64% of HR.

RQ3.

To assess this question, we conduct an ablation study in which we vary the number of training and evaluation positive items. In Figure 3.7, we report the results of all models using 2, 3, 4, 5, and 10 training positive items. In the traditional evaluation protocol with Fixed (Figure 3.7a), increasing the number of training positive items seems to impact the performances negatively: the more the training positive items, the worse the performance. In the traditional evaluation protocol,



(a) Traditional evaluation protocol - Fixed Item Relevance



(b) MRFI - Linear Item Relevance

Figure 3.7: NDCG@10 at varying training epochs for the new item-relevance models and training positives on MovieLens-1M

the behavior is generally expected, as we only evaluate the model with one positive item; the other positive items used for training can only confound the model. Instead, we see something different in the new evaluation protocol (Figure 3.7b), where we evaluate models with ten positive items. Linear shows interesting results: performance increases with the number of training positives. This suggests that our models still have room for further improvement. Although not shown here, a similar but less pronounced result is seen using the Traditional evaluation protocol.

3.2.6 Conclusions

In this work, we challenged the assumption made in Sequential Recommendation Systems (SRSs) of considering only the immediate next item in a sequence for prediction. We have relaxed the evaluation protocol to better assess a model’s performance and designed an item relevance loss to optimize the model to predict multiple future items. Our experiments demonstrated the importance of more positive items in both training and evaluation of SRSs. Results show that when trained in the multiple relevant item regime, our systems outperform the state-of-the-art models 1.2% in NDCG@10 and 0.88% in HR@10 in the original evaluation protocol. In the new evaluation protocol, the improvement is 1.63% of NDCG@10 and 1.5% of HR. Among the item relevance variants we experimented with, the Linear approach outperforms the others and demonstrates its potential usefulness in practical applications.

3.3 Leveraging Inter-rater Agreement for Classification in the Presence of Noisy Labels

In practical settings, classification datasets are obtained through a labelling process that is usually done by humans. Labels can be noisy as they are obtained by aggregating the different individual labels assigned to the same sample by multiple and possibly disagreeing, annotators. The inter-rater agreement on these datasets can be measured while the underlying noise distribution to which the labels are subject is assumed to be unknown. In this work, we: (i) show how to leverage the inter-annotator statistics to estimate the noise distribution to which labels are subject; (ii) introduce methods that use the estimate of the noise distribution to learn from the noisy dataset; and (iii) establish generalization bounds in the empirical risk minimization framework that depend on the estimated quantities. We conclude the paper by providing experiments that illustrate our findings.

3.3.1 Introduction

Supervised learning has seen enormous progress in the last decades, both theoretical and practical. Empirical risk minimization is used as a learning framework [300], which relies on the assumption that the model is trained with iid (independent and identically distributed) sampled data from the joint distribution between features and labels. As a consequence of generalization bounds, when this assumption is satisfied, any desired performance can be achieved as long as enough training data is available. However, in many real-world applications, due to flaws during the data collection and labeling process, the assumption that the training data is sampled from the true feature-label joint distribution does not hold. Training data is often annotated by human raters who have some non-zero probability of making mistakes. It has been reported in [279] that the ratio of corrupted labels in some real-world datasets is between 8.0% and, 38.5% . As a consequence of the presence of incorrect labels in the training dataset, the aforementioned assumption is violated and hence performance guarantees based on generalization bounds no longer hold.

This gap between theory and practice raises the question of whether it is possible to learn from datasets with noisy labels while still having performance guarantees. This question has received a lot of attention lately and has already been answered positively in some cases [198, 214]. Indeed multiple works have introduced learning algorithms that can cope with datasets with incorrect labels while guaranteeing desirable performance through provable generalization bounds. However, these solutions do not solve the entirety of the problem due to the fact that they rely on precise knowledge of the error rate to which the labels are subject, which is often unknown in practice. Several works [214, 325, 333] attempt to address this issue by introducing techniques to estimate such an error rate. Some of these methods have the drawback of relying on assumptions that do not always hold in practice, such as the existence of anchor samples [214]. Ideally, it would be desirable to design learning algorithms that are both robust to noisy labels, and for which performance guarantees can be provided.

An approach, often used in industry to reduce the impact of errors made by human raters, is to label the same dataset multiple times by different annotators. Then the individual labels are combined to reduce the probability of erroneous labels in the dataset, two popular approaches are majority vote or soft labeling. In these cases inter-annotator agreement (IAA) scores (like Cohen’s kappa [56] and Fleiss’ kappa [86]) provide measurable metrics that are directly related to the probability of error present in the labels.

Since the IAA holds a direct relationship with the error rate associated with the human raters, one could potentially estimate the error rate and leverage this estimate to modify the learning algorithms with the objective of making them robust to the resulting noise in the labels. This is the main direction we explore in this work.

Motivation and Contributions: This work is motivated by two main points: (i) to the best of our knowledge there are no published results that indicate how to leverage the IAA statistics to estimate the label noise distribution; and (ii) the generalization bounds of existing noise tolerant training methods often rely on **unknown** quantities (like the true noise distribution) instead of on quantities that can be measured (like the IAA statistics).

Our contributions are the following: (i) we provide a methodology to estimate the label noise distribution based on the IAA statistics; (ii) we show how to leverage this estimate to learn from the noisy dataset; and (iii) we provide generalization bounds for our methods that depend on **known** quantities.

3.3.2 Related works

Our work is related to literature on three main topics: (i) robust loss function design, (ii) label aggregating and (iii) noise rate estimation.

Robust loss functions In classification tasks, the goal is to obtain the lowest probability of classification error. The 0 – 1 loss counts how many errors a classifier makes on a given dataset and is often used in the evaluation of the classifier. However, it is rarely used in optimization procedures because it is non-differentiable and non-continuous. To overcome this, many learning strategies use some convex *surrogates* of the 0 – 1 loss function (e.g. hinge loss, squared error loss, cross-entropy).

It was proved ([100], [99]) that *symmetric* loss functions, that are functions for which the sum of the risks over all categories is equivalent to a constant for each arbitrary example, are robust to label noise. Examples of symmetric loss functions include the 0 – 1 loss, the Ramp Loss and (softmax) Mean Absolute Error (MAE). In [348] authors show that even if MAE is noise tolerant and categorical cross entropy (CCE) is not, MAE can perform poorly when used to train DNN in challenging domains. They also propose a loss function that can be seen as a generalization of MAE and CCE. Several other loss functions that do not strictly satisfy the symmetry condition have also been proposed to be robust against label noise when training deep neural networks [83, 188, 313].

[198] presents two methods to modify the surrogate loss in the presence of class-conditional random label noise. The first method introduces a new loss that is an unbiased estimator for a given surrogate loss, and the second method introduces a label-dependent loss. The paper provides generalization bounds for both methods, which depend on the noise rate of the dataset and the complexity of the hypothesis space.

Labels aggregation When constructing datasets for supervised learning, data is often not labelled by a single annotator, rather multiple imperfect annotators are asked to assign labels to documents. Typically, separate labels are aggregated into one before learning models are applied [67, 231]. In our work, we propose to exploit a measure of the agreement between annotators to explicitly calculate the noise of the dataset. Recently some works revisited the choice of aggregating labels. In [224] authors explore how to train LETOR models with relevance judgments distributions instead of single-valued relevance labels. They interpret the output of a LETOR model as a probability value or distribution and define different KL divergence-based loss functions to train a model. The loss they proposed can be used to train any ranking model that relies on gradient-based learning (in particular they focused on transformer-based neural LETOR models and on the decision tree-based GBM model). However, the authors do not directly estimate the noise rates in the annotations or study how learning from these noisy labels affects the generalization error of the models trained with the methods they introduce. In [317] the authors analyze the performance of both label aggregation and non-aggregation approaches in the context of empirical risk minimization for a number of popular loss functions, including those designed specifically for the noisy label learning problem. They conclude that label separation is preferable to label aggregation when noise rates are high or the number of labellers/annotations is insufficient. [219] and [297] exploit the availability of multiple human annotations to construct soft labels and concludes that this increases performance in terms of generalization to out-of-training-distribution test datasets and robustness to adversarial attacks. [57] focus on efficiently eliciting soft labels from individual annotators.

Noise rate estimation A number of approaches have been proposed for estimating the noise transition matrix (i.e. the probabilities that correct labels are changed for incorrect ones) [187, 214, 357]. Usually, these methods use a small number of anchor points (that are samples that belong to a specific class with probability one) [121]. In particular, [214] proposed a noise estimation method based on anchor points, with the intent to provide an ‘end-to-end’ noise-estimation-and-learning method. Due to the lack of anchor points in real data, some works focused on a way to detect anchor points in noisy data, [325, 333]. In [333] the authors propose to introduce an intermediate class to avoid directly estimating the noisy class posterior. [342] also propose an iterative noise estimation heuristic that aims to partly correct the error and pointed out that the methods introduced by [214] and [333] have an error in computing anchor points, and provide conditions on the noise under which the methods work or fail. [325] provides a solution that can infer the transition matrix without anchor points. Indeed they use the instances with the highest class posterior probabilities for noisy data as anchor points. Our work differs from the mentioned work that uses anchor points because we do not need to assume the existence of anchor points or to have a validation set to learn the noise rate and we only use noisy data to train our model, moreover we neither aim to detect anchor points in the noisy data. Also, most of these works do not study the generalization properties of the proposed models, while we also address this problem and find bound that depend on the estimated noise transition matrix.

Another approach is based on the clusterability condition, that is an example belongs to the same true class of its nearest-neighbors representations. [356] presented a method that relies on statistics of high-order consensus among the 2 nearest-neighbors noisy labels.

3.3.3 Problem formulation

Notation

In this paper we follow the following notation. Matrices and sets are denoted by upper-case and calligraphic letters, respectively. The space of d -dimensional feature vectors is denoted by $\mathcal{X} \subset \mathbb{R}^d$.

We denote by C the number of classes and by e_j the j -th standard canonical vector in \mathbb{R}^C , namely the vector that has 1 in the j -th position and zero in all the other positions. $\mathcal{Y} = \{e_1, \dots, e_C\} \subset \{0, 1\}^C$ is the label set. Feature vectors and labels are denoted by x and y , respectively. \mathcal{D} is the joint distribution of the feature vectors and labels, i.e. $(x, y) \sim \mathcal{D}$. The sampled dataset of size n is denoted by $\widehat{\mathcal{D}} = \{(x_i, y_i)\}_{i=1}^n$. $f(x)$ denotes the output of the classifier f for feature vector x and is a C dimensional vector. All vectors are column vectors.

We denote by $\ell(t, y)$ a generic loss function for the classification task that takes as input C dimensional vectors t and y . In practice t will contain the prediction of the model, and y will be the ground-truth label as a one-hot encoded vector. Namely $\ell : [0, 1]^C \times \mathcal{Y} \rightarrow \mathbb{R}$.

Background

We consider the classification problem within the supervised learning framework, where the ultimate goal is to minimize the ℓ -risk $R_{\ell, \mathcal{D}}(f) = \mathbb{E}_{(x, y) \sim \mathcal{D}}[\ell(f(x), y)]$, for some loss function ℓ . We denote by \mathcal{D} the joint distribution of feature vectors x and labels y . In practice, since the distribution is unknown instead of minimizing $R_{\ell, \mathcal{D}}(f)$ we minimize an empirical risk over some sampled dataset $\widehat{\mathcal{D}}$:

$$\widehat{R}_{\ell, \widehat{\mathcal{D}}}(f) = \frac{1}{n} \sum_{i=1}^n \ell(f(x_i), y_i) = \mathbb{E}_{(x, y) \sim \widehat{\mathcal{D}}}[\ell(f(x), y)]. \quad (3.4)$$

In this work, we assume that the true labels y_i are unknown and consider two scenarios, both of which rely on H annotators.

Scenario I

In this scenario we have access to the H labels provided by the annotators for each sample, where $y_{i,a}$ refers to the label provided by the a -th annotator for the i -th sample. For a given feature vector x_i the distribution of labels provided by annotator a is given by its noise transition matrix T_a , which is defined as follows:

$$(T_a)_{i,j} := \mathbb{P}(y_a = j | y = i) \quad (3.5)$$

Assumption 2. *We assume that all annotators have the same noise transition matrix (i.e. $T_a = T$ for all a), that T is symmetric and that its diagonal elements are larger than 0.5 (i.e. $\mathbb{P}(y_a = i | y = i) > 0.5, \forall i \in \{1, \dots, C\}$).*

Note that by definition T is right stochastic and hence also doubly stochastic. It is also strictly diagonally dominant and therefore non-singular.

Proposition 5.0.1. *T is positive definite.*

Proof. Since T is symmetric it follows that all eigenvalues are real. Combining the fact that it is strictly diagonally dominant with Gershgorin's theorem we conclude that all eigenvalues lie in the range $(0, 1]$ and hence T is positive definite. \square

Assumption 3. We assume that the annotators are conditionally independent on the true label y :

$$\mathbb{P}(y_a, y_b | y) = \mathbb{P}(y_a | y) \mathbb{P}(y_b | y). \quad (3.6)$$

We now define the IAA matrix M_{ab} between annotators a and b as follows:

$$(M_{ab})_{i,j} := \mathbb{P}(y_a = i, y_b = j) \quad (3.7)$$

Proposition 5.0.2. Leveraging Assumption 3 the agreement matrix $M_{a,b}$ can be written as follows:

$$M_{a,b} = T_a^T D T_b \quad (3.8)$$

$$D := \text{diag}\{\nu\} \quad (3.9)$$

$$\nu := [\mathbb{P}(y = 1), \dots, \mathbb{P}(y = C)]^T. \quad (3.10)$$

Due to Proposition 5.0.1 and the fact that D is positive definite, it follows that all matrices $M_{a,b}$ are invertible.

Assumption 4. We assume that the class probabilities (and hence D) are known.

Due to Assumption 2, all annotators share the same noise transition matrix T . Therefore M_{ab} is independent of a and b , and from now on, we remove this dependency in the notation (i.e. we get $M = T^T D T$). Furthermore, since T is invertible and D is diagonal and positive definite, it follows that M is also positive definite.

Note that since we have access to all the labels provided by the H annotators for all the samples, we can obtain an estimate of M which we denote \widehat{M} .

Assumption 5. We assume that \widehat{M} is a consistent estimator.

For the case of two annotators, one possible consistent estimator $\widehat{M}_{a,b}$ that exploits its symmetry condition is given by:

$$(\widehat{M}_{a,b})_{i,j} = \sum_{k=1}^n \frac{\mathbb{1}(y_{a,k}=i, y_{b,k}=j) + \mathbb{1}(y_{a,k}=j, y_{b,k}=i)}{2n} \quad (3.11)$$

If the annotators have the same transition matrix, M will be the same for all pairs of annotators. So we can estimate M , in the case of $H \geq 2$ by averaging the estimators \widehat{M}_{ab} obtain by 3.11 for all possible pairs of annotators. The estimator in this case can be written as

$$(\widehat{M})_{i,j} = \frac{1}{H(H-1)} \sum_{a=1}^H \sum_{\substack{b=1 \\ b \neq a}}^H \sum_{h=1}^n \frac{\mathbb{1}(y_{a,h}=i, y_{b,h}=j)}{n}. \quad (3.12)$$

Scenario II

In the second scenario, for each i -th sample we are given a unique label \tilde{y}_i that is produced by aggregating the H individual labels according to some known aggregating policy (like majority vote). In this case, since we do not have access to the individual annotations we assume that \widehat{M} is provided.

The probability that label y_i is corrupted to some other label \tilde{y}_i is given by the *aggregated noise transition matrix* $\Gamma \in [0, 1]^{C \times C}$, where $\Gamma_{ij} := \mathbb{P}(\tilde{y} = j | y = i)$ is the probability of the true label i being flipped into a corrupted label j and C is the number of classes. Note that by definition Γ is a right stochastic matrix that is determined by T , the amount of annotators H and the aggregating policy. We will study both the case where $\Gamma = T$, and the case in which there exists a generic Lipschitz function ϕ so that $\Gamma^{-1} = \phi(T)$.

There are different policy choices to construct the dataset that lead to $\Gamma = T$. If we decide to use only one annotator, for instance a , to build the final dataset, namely for each sample $\tilde{y}^i = y_a^i$ we have $\Gamma = T_a$. Or if annotators are homogeneous, i.e. they have the same noise transition matrix T , and to build the final dataset we decide to randomly select the label of one of the annotators we have that $\Gamma = T$.

Even restricting ourselves to the case of homogeneous annotators, depending on the rule with which we build the dataset we can have a more complex relationship between the matrix T and Γ .

We also obtain generalization bounds in the case where an estimate of the agreement matrix M is not available and we only have access to a scalar representation of the inter-annotator agreement, in particular we consider the case where the Cohen's κ is given.

Objective

The objective in both scenarios is to: i) use \widehat{M} to estimate the noise transition matrices (T and Γ); ii) leverage these estimates to be able to learn from the noisy dataset in a more robust manner; and iii) obtain generalization bounds for the resulting learning methods.

3.3.4 Main results

We divide the main contributions in three sections. In the first section we show how to estimate the noise matrices T . Next we indicate how to leverage these estimates to learn for the datasets with noisy labels. Finally we obtain bounds, depending on the Rademacher complexity of the class of functions, on the generalization gap for a bounded and Lipschitz loss function

Estimation of the noise transition matrices

We start stating the following Lemma that allows us to write the unknown matrix T (and its inverse), as a function of D and M .

Lemma 5.1. If $D^{\frac{1}{2}}$ commutes with T we have that:

$$T = U\Lambda^{\frac{1}{2}}U^T \quad (3.13)$$

$$T^{-1} = U\Lambda^{-\frac{1}{2}}U^T \quad (3.14)$$

$$D^{-\frac{1}{2}}MD^{-\frac{1}{2}} = U\Lambda U^T \quad (3.15)$$

where $U\Lambda U^T$ is the eigenvalue decomposition of $D^{-\frac{1}{2}}MD^{-\frac{1}{2}}$ (i.e. U is some orthogonal matrix and Λ is a diagonal positive definite matrix).

A detailed discussion of when the commutativity assumption is satisfied is included in 3.0.2. The proof of the previous Lemma can be find in 3.0.3.

Note that we could use Lemma 5.1 to estimate T as follows:

$$\hat{T} = \hat{U}\hat{\Lambda}_M^{\frac{1}{2}}\hat{U}^T \quad (3.16)$$

where $\hat{U}\hat{\Lambda}_M^{\frac{1}{2}}\hat{U}^T$ is the eigenvalue decomposition of $D^{-\frac{1}{2}}\hat{M}D^{-\frac{1}{2}}$. However such estimate can result in matrices that are not doubly stochastic, or diagonally dominant due to estimation errors. A more accurate estimate of T could be obtained as $\hat{T} = \pi(\hat{U}\hat{\Lambda}_M^{\frac{1}{2}}\hat{U}^T)$ where π is a projection operator to the set of doubly stochastic, positive definite matrices with diagonal elements greater than 0.5 and non-negative entries (which is a convex set). We can obtain such projection by solving the following optimization problem:

$$\hat{T} = \pi(\hat{U}\hat{\Lambda}_M^{\frac{1}{2}}\hat{U}^T) = \underset{B}{\operatorname{argmin}} \|B - \hat{U}\hat{\Lambda}_M^{\frac{1}{2}}\hat{U}^T\|_2^2 \quad (3.17)$$

$$\begin{aligned} & B = B^T \\ & \sum_j B_{i,j} = 1 \quad \forall i \\ \text{s.t.} & \\ & B_{i,j} \geq 0 \quad \forall i, j \\ & B_{i,i} \geq 0.5 \quad \forall i \end{aligned}$$

Note that this optimization problem is convex because the constraints are linear and for symmetric matrices it holds that $\|\hat{T} - \hat{U}\hat{\Lambda}_M^{\frac{1}{2}}\hat{U}^T\|_2^2 = \lambda_{\max}(\hat{T} - \hat{U}\hat{\Lambda}_M^{\frac{1}{2}}\hat{U}^T)$, which is a convex function of \hat{T} .

To summarize, T can be estimated as follows. First, obtain an estimate of M . Then obtain the eigenvalue decomposition of $D^{-\frac{1}{2}}\hat{M}D^{-\frac{1}{2}} = \hat{U}\hat{\Lambda}\hat{U}^T$ (note that this decomposition always exists because $D^{-\frac{1}{2}}\hat{M}D^{-\frac{1}{2}}$ is symmetric). Finally obtain the estimate as: $\hat{T} := \pi(\hat{U}\hat{\Lambda}^{\frac{1}{2}}\hat{U}^T)$.

Note that once the estimate of \hat{T} is obtained, $\hat{\Gamma}$ can be obtained since we assumed the label aggregating policy to be known.

Lemma 5.2. Let $M_{a,b}$ be the agreement matrix for annotators a and b defined in Eq. (3.7) and $\widehat{M}_{a,b}$ be the estimated agreement matrix defined in Eq. (3.11) and let $\|\cdot\|_p$ be the matrix norm induced by the p vector norm. For every $p \in [1, \infty]$ and for every $\delta > 0$, with probability at least $1 - \delta$

$$\|M_{a,b} - \widehat{M}_{a,b}\|_p \leq \sqrt{\frac{C^2}{2n} \ln \frac{2C^2}{\delta}}. \quad (3.18)$$

where \mathbb{P}^n denotes the probability according to which the n training samples are distributed, i.e. we are assuming that the samples are independently drawn according the probability \mathbb{P} .

Proof. The proof can be found in Appendix 3.0.3. \square

From Lemma 5.2 it follows that if \widehat{M} is estimated as in 3.12, since \widehat{M} is an average of \widehat{M}_{ab} it also holds that for every $p \in [1, \infty]$ and for every $\delta > 0$, with probability at least $1 - \delta$

$$\|M - \widehat{M}\|_p \leq \sqrt{\frac{C^2}{2n} \ln \frac{2C^2}{\delta}}. \quad (3.19)$$

Theorem 5.3. *Let T be the noise transition matrix defined as in 3.5 and \widehat{T} its estimate (defined as in 3.17).*

With probability at least $1 - \delta$:

$$\|T - \widehat{T}\|_2 \leq \frac{C(\sqrt{C} + 1)\lambda_{\max}(D)}{\lambda_{\min}(\widehat{T})} \sqrt{\frac{1}{2n} \ln \frac{2C^2}{\delta}} \quad (3.20a)$$

$$\|T^{-1} - \widehat{T}^{-1}\|_2 \leq \frac{9C(\sqrt{C} + 1)\lambda_{\max}(D)}{\lambda_{\min}(\widehat{T})^2} \sqrt{\frac{1}{2n} \ln \frac{2C^2}{\delta}} \quad (3.20b)$$

$$\text{for } n > \frac{C^2(\sqrt{C}+1)^2(\ln(2C^2))^2}{2\lambda_{\min}(\widehat{T})^2}.$$

Proof. The proof can be found in 3.0.3. \square

From the previous theorem we can notice that the error in estimation of T decays as $\frac{1}{\sqrt{n}}$ as a function of n .

Learning from noisy labels

In this section, we show how to leverage the estimates of the error rates to train the models.

Posterior distribution of true labels as soft-labels

It is noteworthy that if we have access to the labels provided by all annotators, the posterior probabilities of the true labels can be calculated leveraging T and Bayes' Theorem as follows:

$$\underbrace{\mathbb{P}(y_i = c | y_{1,i}, \dots, y_{H,i})}_{:=p_{c,i}} \propto \nu_c \prod_{h=1}^H \underbrace{\mathbb{P}(y_{h,i} | y_i = c)}_{:=T_{c,y_{h,i}}} \quad (3.21)$$

we recall that $\nu_c = \mathbb{P}(y_i = c)$ and that the conditional probabilities on the r.h.s. are given by T . In our case, we can use our noisy transition estimates to estimate the posterior probabilities of the true labels, and afterwards, we can use these posteriors to train the classifier.

Lemma 5.4. For infinite annotators, the posterior distribution over every sample calculated using the true T converges to the Dirac delta distribution centred on the true label almost surely (i.e. $\lim_{H \rightarrow \infty} p_{c,i} \stackrel{\text{a.s.}}{=} \mathbb{1}(y_i = c)$).

Proof. See Appendix 3.0.3. □

We can use the posterior distributions as soft-labels defining the following loss for the i -th sample:

$$\ell(f(x_i), y_{1,i}, \dots, y_{H,i}) = \ell(f(x_i), \bar{p}_i) \quad (3.22)$$

where $\bar{p}_i = [p_{1,i}, \dots, p_{C,i}]^T$. Or we can use the posterior distributions to weight the loss function at the i -th sample evaluated at each of the possible labels:

$$\ell(f(x_i), y_{1,i}, \dots, y_{H,i}) = \sum_{c=1}^C p_{c,i} \ell(f(x_i), e_c) \quad (3.23)$$

where e_c is the vector in \mathbb{R}^C with 1 in the c -th position. Notice that for categorical cross-entropy loss, the two functions defined above correspond, but in general, they define two different loss functions.

Note that these soft labels are different from the ones obtained by averaging the annotator's labels as is done in [317]. The method using the posteriors exploits the T matrix and thus more information than the simple mean of the values of the losses among annotators. We, therefore, expect this to yield better results than the aggregation using the mean proposed in [317]. These considerations are supported by the empirical results we obtained on synthetic datasets (see 3.3.6).

Robust loss functions

Another way to leverage the estimate of T is to use robust loss functions, like the forward and backward loss functions presented in [198, 214]. Let $\ell(t, y)$ be a generic loss function for the classification task, with a little abuse of notation we define $\ell(t) = [\ell(t, e_1), \dots, \ell(t, e_C)]^T$. The backward and forward loss functions are defined in 3.24a and 3.24b, respectively.

$$l_b(t, y) = (\widehat{\Gamma}^{-1} \ell(t)) y \quad (3.24a)$$

$$l_f(t, y) = (\ell(\widehat{\Gamma}^T t)) y \quad (3.24b)$$

To explain the notation in 3.24a we are first doing the dot product between the matrix Γ^{-1} and the vector $\ell(t)$ and then the dot product of the resulting vector with y . These losses leverage aggregated labels and therefore different aggregating techniques can be used, like majority vote. Another possible aggregating technique that leverages the posterior probabilities is to assume that the true label is the one that corresponds to the class that has the highest posterior probability.

Generalizations gap bounds

In this section, we derive generalization gap bounds for the backward loss that depends on the noise transition matrix estimated in 3.17. Since we are only addressing the problem for the backward loss, from now on we will denote the backward loss by l .

Remark 1. *If $\ell(t, y)$ is Lipschitz with constant L , the loss function $l(t, y)$ is Lipschitz with Lipschitz constant $\|\Gamma^{-1}\|_2 L$.*

We will prove the following theorem in the case of $\Gamma = T$. We emphasize that all the results apply also when $\Gamma^{-1} = \phi(T^{-1})$ and that the function that associate Γ^{-1} and T^{-1} , ϕ is Lipschitz

with respect to the norm p , i.e. there exists a Lipschitz constant $L_{\phi,p}$ s.t. $\|\phi(T^{-1}) - \phi(\widehat{T}^{-1})\|_p \leq L_{\phi,p} \|T^{-1} - \widehat{T}^{-1}\|_p$. The only difference is that in the bound we will have a factor $L_{\phi,p}$.

It has been proved, first in [198] (Lemma 1) for the binary classification task and then in general for the multi-class case in [214] (Theorem 1) that $l(t, y)$ is an unbiased estimator for ℓ , i.e.

$$\mathbb{E}_{\tilde{y}|y}[l(t, \tilde{y})] = \ell(t, y).$$

Lemma 5.5. Let ℓ be a bounded loss function, so that the image of ℓ is in $[0, \mu]$, and s.t. ℓ is Lipschitz in the first argument with Lipschitz constant L . Let $\widehat{R}_l(f)$ be the empirical risk for the loss l and let $R_{l,\mathcal{D}}$ be the risk for a loss l under the distribution \mathcal{D} , with l unbiased estimator for the loss ℓ . We denote by \hat{l} the backward loss obtained using \widehat{T} .

$$\sup_{f \in \mathcal{F}} |\widehat{R}_l(f) - R_{l,\mathcal{D}}(f)| \leq \left[L\lambda_{\min}(\widehat{T}^2) + \frac{\mu\lambda_{\min}(D)}{\lambda_{\min}(\widehat{T})^2} \sqrt{\frac{1}{n} \ln\left(\frac{4C}{\delta}\right)} \right] \mathfrak{R}_n(\mathcal{F})g(C).$$

with $g(C) = 6C^2(\sqrt{C} + 1)$

Theorem 5.6. Let l be an unbiased estimator for ℓ defined as in 3.24a, Denoting $\hat{f} = \operatorname{argmin}_f(\widehat{R}_l(f))$. It holds that

$$R_{\ell,\mathcal{D}}(\hat{f}) - \min_{f \in \mathcal{F}} R_{\ell,\mathcal{D}}(f) \leq \left[2L\lambda_{\min}(\widehat{T}^2) + \frac{\mu\lambda_{\min}(D)}{\lambda_{\min}(\widehat{T})^2} \sqrt{\frac{1}{n} \ln\left(\frac{4C}{\delta}\right)} \right] \mathfrak{R}_n(\mathcal{F})g(C)$$

with $g(C) = 6C^2(\sqrt{C} + 1)$

The proofs of Lemma 5.5 and 5.6 can be find in 3.0.3. We observe that in all the previous theorems, the bounds found are always decreasing as one over the square root of the number of samples. The above theorem gives us a performance bound for the classifier found minimizing the backward loss l , i.e. the unbiased estimator of the loss ℓ on the noisy dataset. The bounds found depend on, the Rademacher complexity of the function space and the Lipschitz constant of the loss function. The importance of these bounds lies in the fact that they allow us to obtain performance bounds for a model trained with noisy data that depends on values that we can estimate from the noisy dataset. In particular, there is no dependence on the true noise transition matrix of the annotators, as in other work [198] which is instead a quantity that cannot be known a priori having access only to the training data. More in detail the bound depends on the estimate noise transition matrix, the number of classes in the dataset, the Rademacher complexity and the Lipschitz constant, which we can take as known a priori and on the distribution of ground truth, which in many cases it makes sense to assume uniform.

3.3.5 Cohen's κ

We can also consider the case where an estimate of the IAA matrix M is not available and we only have access to a scalar representation of the inter-annotator agreement like Cohen's κ . In this case, we can only estimate one parameter and hence the matrix T has to be parameterized by a single parameter that can be estimated.

One particular example is the case where the noise is uniform among classes. Under these hypotheses, T is a matrix with all values $1 - p$ on the diagonal and $\frac{p}{C-1}$ off the diagonal.

Lemma 5.1 (Relationship between p and κ). In the case of classification with uniform noise for two homogeneous annotators with noise rate p , i.e if a is one annotator, $\mathbb{P}(y_a = i | y = j) = p$ if $i \neq j$. If the distribution of the ground-truth labels is uniform, it holds that:

$$p = (1 - C^{-1})(1 - \sqrt{\kappa}) \quad (3.25)$$

with κ the Cohen’s kappa coefficient of the two annotators (see 3.0.1).

Proof. The proof can be found in 3.0.3. □

If T is assumed to be of the form described above (with all diagonal elements equal to $1 - p$ and all off-diagonal entries equal), it has one eigenvalue equal to 1 and all the rest are equal to $1 - pC(C - 1)^{-1}$ (this follows from the fact that in this case T can be written as a weighted summation of the identity and a rank-one matrix). Hence using 3.25 we get that $\lambda_{\min}(T) = \sqrt{\kappa}$. The bounds from Theorem 5.6 holds replacing $\lambda_{\min}(T)$ with $\sqrt{\kappa}$. This allows us to obtain a bound for the generalization gap of a classifier trained with backward loss even in the case where a single statistic on the agreement between annotators is provided.

3.3.6 Experimental results

We performed experiments to validate the effectiveness of the method we propose for estimating \hat{T} by studying the error in the estimation as a function of the number of samples. We also performed experiments to show how the estimated T can be leveraged to train classifiers in the presence of noise labels. In particular, we performed experiments for a classification task on a synthetic dataset and on the CIFAR10-N dataset, comparing the performance of a classifier trained using labels obtained by some baseline aggregation method with the performance of a classifier trained using the distribution of posteriors obtained from the estimation of T (3.21) as soft-labels.

Estimation of T With these experiments, we aim to validate the theoretical results of 3.3.4. We generate various matrices T that is symmetric, stochastic and diagonally dominant, the exact details about the generation of T can be found in 3.0.4. For each annotator, we produce their prediction according to the matrix T . We run experiments for the number of annotators $H = 10, 7, 3, 2$. We report here the results for $H = 10$, and 4 classes, all the other plots are in 3.0.4. In C.2 (as well as the plots in the Appendix) we can be observed that the error in the estimation decreases as $\frac{1}{\sqrt{n}}$ with n number of samples, which is in agreement with the bound provided in 5.3. We also observed that, as expected, the estimation becomes more accurate as the number of annotators increases.

Classification task with synthetic data We consider a classification task with a synthetic dataset. The features are generated uniformly in $[0, 1]^2$. The assignment of labels (y) is done by following the label distribution established for each experiment, separating the space with lines parallel to the bisector of the first and third quadrants. More information on how the class distributions are generated can be found in Appendix 3.0.4.

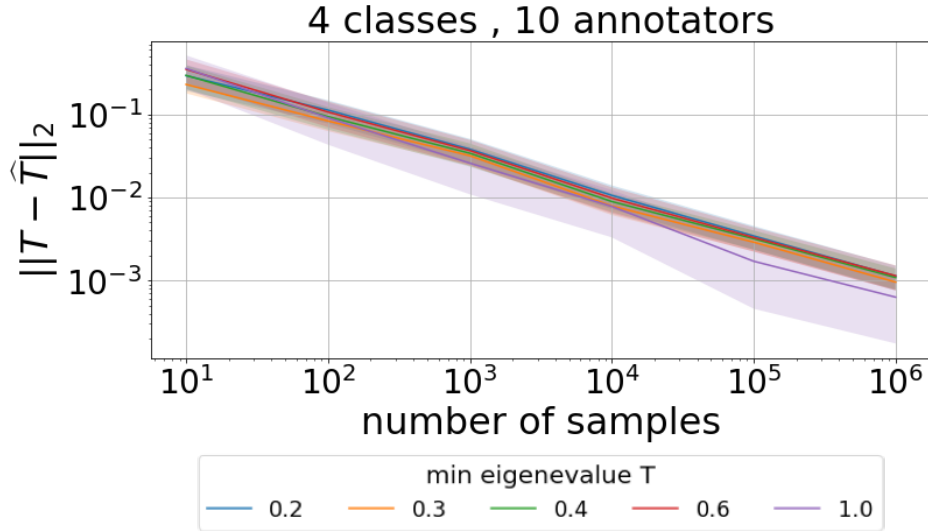


Figure 3.8: Error in the estimation of T for 4 classes and 10 annotators. The plots are obtained by averaging different admissible matrices T (see 3.0.2) and averaged over matrices that have the same minimum eigenvalues rounded to the first decimal.

For each dataset, annotations are generated according to the noise transition matrix T . Various combinations of T are tested that respect the assumptions of symmetry, stochasticity and diagonally dominance, as well as being commutative with D (more details can be found in Appendix 3.0.2). The number of annotators is variable in the set $\{3, 5\}$. See Appendix 3.0.4 for implementation details.

Losses We use categorical cross entropy as loss function. We use both hard labels and soft labels to train the models.

To train the models with hard labels an aggregation method is needed to obtain one final label from the annotators. We consider random and majority votes. In random aggregation, the final label is randomly picked from the labels of the annotators. In the majority vote the final label is the one with the most amount of votes (the mode), if the mode is not unique, we randomly choose one of the most voted classes. As soft labels, we consider the relative frequency among annotators and the posterior distribution according to 3.21. In the case of frequency for each sample we average the one-hot encoded annotations. Notice that random, majority vote and frequency soft labels do not leverage the estimate of T while the posterior does. In 3.9 we report the results for 4 classes with distribution $(0.4, 0.1, 0.4, 0.1)$ and 3 annotators.

We use accuracy with respect to a clean dataset as a performance metric. Our results show that using the posteriors distribution, as soft labels, allows for better performance than using the average of the labels assigned by annotators and then using majority vote or random aggregation.

Our method is shown to be more robust to the noise and is also the one with less variance in the results. This confirms our hypotheses that by leveraging the matrix \hat{T} better classification accuracy can be achieved.

Experiments on CIFAR10-N The CIFAR10-N dataset⁴ contains CIFAR-10 train images with noisy labels annotated by humans using Amazon Mechanical Turk. Each image is labelled by three

⁴<http://www.noisylabels.com>

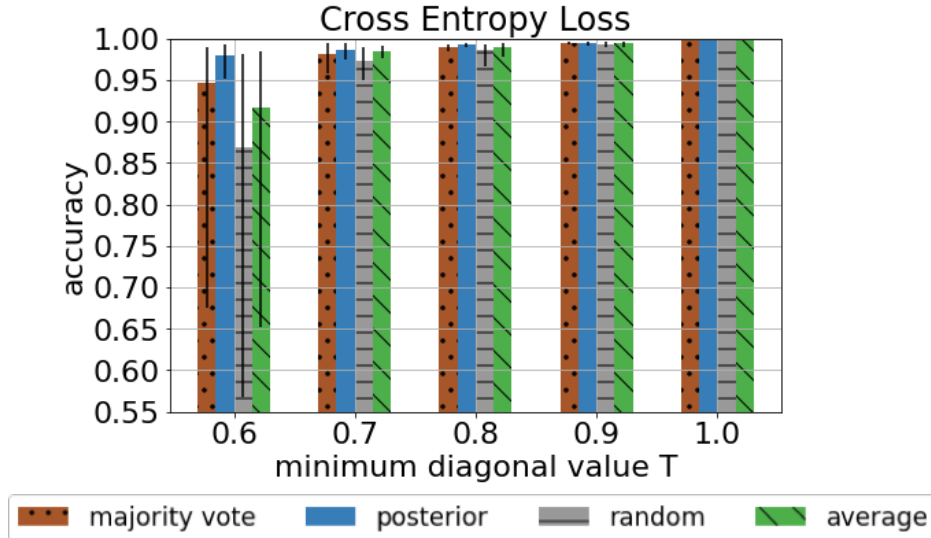


Figure 3.9: Comparison between the performance of Cross Entropy Loss using majority vote, random aggregation method or the posteriors (posterior) and relative frequency (average) as soft labels. On the y-axis the accuracy on a clean dataset and on the x-axis the values of the minimum on the diagonal of T . Small values of the minimum diagonal value mean a noisy dataset, while the minimum is 1 in the noise-free case. The results are obtained for 3 annotators and 4 classes, by averaging on different admissible matrices T (see 3.0.2) that have the same minimum diagonal values rounded to the first decimal. The error bands show the maximum and minimum performance for each method.

independent annotators. Table 3.6 shows the accuracy achieved using the different aggregation methods. For this experiment, we used Resnet34 [147] with and without pre-training. In both cases, our approach of aggregation achieves the best performance. Note that in this dataset there are no guarantees that the assumptions we made on T are satisfied, however, the method is still applicable with positive results.

Aggregation Method	Pretrained	Not-Pretrained
random	0.718 ± 0.035	0.579 ± 0.023
majority vote	0.740 ± 0.017	0.590 ± 0.006
average	0.762 ± 0.012	0.637 ± 0.016
posteriors (ours)	0.794 ± 0.005	0.652 ± 0.014

Table 3.6: Test Accuracy on CIFAR10-N with Resnet34

3.3.7 Concluding remarks

We have addressed the problem of learning from noisy labels in the case where the dataset is labelled by annotators that occasionally make mistakes. We have introduced a methodology to estimate the noise transition matrix T of the annotators given the IAA. We further showed different techniques to leverage this estimate to learn from the noisy dataset in a robust manner. We have shown theoretically that the methods we introduce are sound. We supported our methodology with some experiments that confirm our estimation of the noise transition matrix is valid and that this can be leveraged in the learning process to obtain better performance.

Limitations The main limitation of our current approach to estimating T is that it only considers the case where T is symmetric and D assumed to be known and commutes with T . Extending the results to the case where T might not be symmetric and different among annotators is one possible future research direction.

Chapter 4

Auxiliary Frameworks for Trustworthy AI Systems

While model-centric approaches are essential in developing Trustworthy AI, a comprehensive trustworthiness framework transcends the scope of these approaches. The Auxiliary Framework takes a holistic view of the AI ecosystem, with a particular focus on key areas.

In cases where the primary AI model lacks inherent explainability or is challenging to interpret, auxiliary explainability frameworks step in. These frameworks enable the generation of counterfactual explanations, offering insights into what might have happened differently if certain inputs had been changed. Even when the main model's decision-making process remains complex, counterfactual explanations provide a valuable window into understanding its behavior.

AI systems equipped with retrieval augmentation have the ability to select and fetch relevant information from a data repository, using this knowledge to inform their predictions. This feature not only enhances performance but also facilitates the interpretability of decisions. By knowing the source of the information utilized in a prediction, stakeholders can trust the model more readily, as the reasoning behind the output becomes more transparent.

In situations where abundant unlabeled data is available, active learning enables the model to choose which data points to label, and subsequently, learn from. This strategic selection of data not only accelerates the learning process but also allows the model to focus on data points that are most informative and challenging. The model actively seeks out knowledge, enhancing its competence and transparency.

In summary, this chapter underscores that achieving trustworthiness in AI requires more than just model-centric approaches. By integrating counterfactual explanations, retrieval-augmented capabilities, and active learning strategies, auxiliary frameworks broaden the horizons of AI trustworthiness.

4.1 Human-in-the-loop Personalized Counterfactual Recourse

We introduce a new framework for generating counterfactual recourse in machine learning that embraces a “human-in-the-loop” approach by incorporating user preferences. Traditional counterfactual tools neglect individual user preferences when adjusting features. To address this, we tackle recourse generation as a multi-objective optimization problem, integrating conventional constraints with user preferences. Our framework, termed **HIP-CORE**, is specifically crafted to estimate these preferences during the counterfactual generation phase. We also introduce the “Personal Validity” as a measure of the effectiveness of recourse for individual users. Through extensive theoretical and empirical analysis, we validate the benefits of our proposal. Overall, this work enhances counterfactual reasoning and paves the way for more personalized algorithmic recourse.

4.1.1 Introduction

Algorithmic decision-making systems have become ubiquitous, influencing myriad aspects of our lives, from personalized content recommendations to high stakes decisions in finance, healthcare, and justice. While these algorithms offer efficiency and scalability, their opaqueness often leads to concerns regarding fairness, accountability, and transparency. As a response to these concerns, *eXplainable Artificial Intelligence* (XAI) aims to clarify the complex workings of machine learning models, making their decisions transparent, understandable, and interpretable for end-users, including human-in-the-loop processes [322].

Human-in-the-loop refers to a collaborative approach that integrates human judgment, feedback, and decision-making into automated processes, acknowledging that there are instances where human intervention and expertise are crucial for ensuring quality, fairness, and ethical considerations of AI systems. Incorporating *human-in-the-loop* processes can enhance the accountability and the transparency of AI systems, making them more reliable and aligned with human values and preferences.

As a result of the emergence of international regulation (i.e., GDPR), increasing attention has been devoted to the *right to recourse* [304]: i.e., in the event that an individual receives an unfavorable decision from a model, he/she is also entitled to receive an actionable explanation that can make him/her proactively adapt his/her features to get a positive outcome from the model in the future. Central to this XAI endeavor is the idea of *counterfactual explanations*, a form of example-based explainability that provides insights by presenting alternative scenarios in which a given decision would change [307]. When applied for algorithmic recourse [142], rather than merely explaining why a decision was made, counterfactuals empower users with actionable insights by suggesting how they might alter inputs to achieve a desired outcome [143, 215].

Considering a real case in which a user applies for a loan and a credit-scoring model gives as output “denied”, but the user is presented with a counterfactual recourse. The counterfactual recourse must allow the user to change the output to “accepted” (i.e. *validity*), while not being too

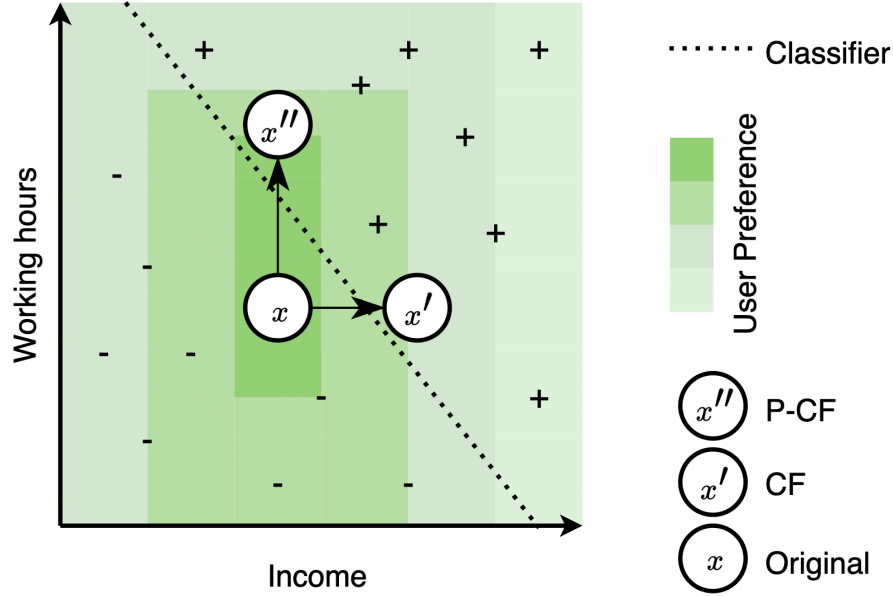


Figure 4.1: Dummy example of personalized counterfactual recourse: given the original instance x , classified in the negative class $-$, a counterfactual recourse algorithm produces x' , that asks the user to increase the income. The heat-map on the 2D plane represents the user preference over the counterfactual feature space. Our HIP-CORE framework, which takes the user preference into account, produces x'' , that is an optimal solution considering the trade-off between counterfactual validity and user-preference.

different from the user’s initial status, e.g. suggesting to change only few features, *sparsity*, and asking to change the features values minimally, *proximity*. Since the solution may not be unique depending on the definition of the problem and method used for the solution, to avoid eclipsing some potential explanations and relevant alternatives to the user (*personalization*) [316], multiple counterfactuals can be presented to the user (*diversity*) [196].

In this context, an individual has always been considered as a *rational* agent, so the objective assumptions of *proximity*, *sparsity*, and other constraints related to the underlying model or generic assumptions (a.k.a. user-agnostic), do not consider the irrationality and the subjectivity of human judgment of an algorithm output. Therefore, in this paper we consider the problem of creating counterfactual recourse that are tailored to the subjective and irrational preferences of the user: a personalized counterfactual recourse that includes the user in the generation process to estimate and integrate personal preferences in the solution.

Example. In Figure 4.1, we show a dummy example of the advantage of generating a **personalized** counterfactual recourse. The plane represents a 2D projection of the feature space hyper-plane for two features: **Working hours** and **Income**. The user x is classified in the plane as negative $-$ by the simplified line classifier (dashed line). The counterfactual *CF* data point x' is the optimal solution of a *user-agnostic* counterfactual recourse problem, where the x' counterfactual recourse solution recommends that the user increases the **Income** feature to get a positive $+$ output. Given the feasibility of interacting with the user, we consider the user preference a central factor, represented on the plane as a heat map: the user prefers darker areas. A counterfactual recourse method based on user preferences would output x'' , that is the optimal solution considering both generating a valid counterfactual (i.e. positive output) and maximizing user preferences on feature change (increasing

Working hours instead of Income), thus leading to a solution more easily achievable for the user.

The goal of this paper is to propose such a system, which we dub **HIP-CORE** (Human-In-the-Loop Preference COunterfactual REcourse).

Summary of contribution The contributions of this work are summarized as follows:

1. We formalize the problem of Personalized Counterfactual Recourse (Section 4.1.3), as a multi-objective optimization problem aggregating optimization functions for both user-agnostic metrics (i.e., validity, sparsity, proximity) and user-level metrics (i.e., preference).
2. We present our algorithmic framework, **HIP-CORE** to generate preference-driven counterfactual while estimating the preferences of the user (Section 4.1.5).
3. Key to the development of **HIP-CORE** is a mathematical framework to represent and estimate user preferences over the complex space of counterfactual feature change (Section 4.1.5).
4. We introduce a new metric called *Personal Validity*, a natural extension of Validity to incorporate users' preferences in the evaluation.
5. We assess our framework empirically on widely used benchmarks, comparing with a user-agnostic baseline, confirming the importance of including user preferences in the counterfactual recourse generation process (Section 4.1.6).

4.1.2 Related Work

Counterfactual Recourse. Defining and searching for the target counterfactual, it is not trivial in the complex feature space of the instances and the black-box classifier [10]. Traditional assumptions to search a good counterfactual instance include the classification into the opposite class (termed as *Validity*) and its similarity to the original instance [307]. The latter constraint is frequently expressed in terms of *sparsity*, trying to minimize the number of features changed, and *proximity*, which tries to minimize the magnitude of feature change. Other definitions of counterfactual instance are related to the underlying features model, such as a Structural Causal Model [143], the solution has to be coherent with respect to the causal constraints between features. If the distribution of the feature space is known (not data-agnostic), more feasible counterfactuals can be created with the a priori knowledge, such as through *data manifold closeness* [302].

Actionable Counterfactual Recourse. Actionability of counterfactual recourse [229] refers to taking into account only those feature changes that an agent can feasibly implement. The personalization of counterfactual recourse is closely related to the local actionability for a user, that in other works is pursued with an ex-ante [229] or post-hoc [196] filter on the generated counterfactuals. We integrate user preferences directly into counterfactual recourse generation, addressing the issue of actionability through explicit human judgment.

Diverse Counterfactual Recourse. Another strategy is to create a set of counterfactuals that are different one another, therefore, a set of acceptable solutions is proposed that maximize a *diversity* function [161, 196], to allow the user to choose the most appropriate one.

Personalized Counterfactual Recourse. Few recent attempts have tried to incorporate user preferences in the generation process of counterfactual recourse. The cost of adoption of the counterfactual change is the pivotal point to discriminate among user-centered approach, where the

preference is incorporated in the cost function, and user-agnostic method, where the cost function do not take into account the user preferences. Some approaches request users to specify their preferences over a set of solutions [196, 327], or attempt to quantify the cost of potential changes in advance [230, 315]. Such methodologies do not incorporate the user within the counterfactual generation loop, consequently neglecting the exploration of the counterfactual preference space.

Human-in-the-loop Algorithmic Recourse. Research on the human-in-the-loop approach, combining preference elicitation to create solutions that align with user preferences, is notably limited. In contrast from previous work, we offer a broader approach that doesn't rely on Structural Causal Models [68] or impose constraints on preference modeling [335]. Our method also encompasses the estimation of user preferences within the counterfactual recourse generation process.

None of the existing approaches include, in a unique multi-objective problem, different properties as we propose in this paper: namely, the user-centered (i.e., preference), the general (i.e., validity), and the data-specific (i.e. proximity, sparsity).

4.1.3 Problem Statement

A user $u \in \mathcal{U}$ is described as a point $x_u \in \mathcal{X}$ in a feature space $\mathcal{X} \subseteq \mathbb{R}^n$, with $n \in \mathbb{N}^+$. Users are subjected to evaluation by a black-box classifier¹ $f : \mathcal{X} \rightarrow [0, 1]$. The *algorithmic recourse problem* is defined when a user u gets a negative outcome $f(x_u) < \tau$ and needs to receive a recourse. The *counterfactual* formulation of the recourse problem provides the recourse as a new counterfactual configuration of the user point $x'_u \in \mathcal{X}$ that allows the user to get a positive outcome $f(x'_u) \geq \tau$, with $\tau \in [0, 1]$ (generally $\tau = 0.5$). Differentiating x according to the user is important because two users $u, v \in \mathcal{U}$ represented by identical vectors $x_u = x_v$, might have different preferences (see following sections). Nevertheless, in the absence of ambiguity, the subscript will be omitted.

Several requirements are typically incorporated into the standard counterfactual recourse problem: the counterfactual x' must be close to the original point x (*Proximity*), x' must change the minimum number of features of x (*Sparsity*), when producing multiple counterfactuals for the same x , they must be diverse in nature (*Diversity*). Besides these standard requirements, we introduce user preference as a key property to generate user-centered counterfactual recourse.

Definition 4.1.3.1. The preference of a user $u \in \mathcal{U}$ for a counterfactual $x'_u \in \mathcal{X}$ is a probability $\Pi_u(x'_u) = P(x'_u | x_u, u)$ that the user accepts the counterfactual instance $x'_u \in \mathcal{X}$ as a recourse, with $\Pi_u : \mathcal{X} \rightarrow [0, 1]$.

We are not defining an absolute preference of a user within the space \mathcal{X} , but rather how willing the user is to alter their current state and in what manner.

We are now ready to formalize our problem.

Problem 1 (Personalized Counterfactual Recourse Problem). Given a user $u \in \mathcal{U}$, with $x_u \in \mathcal{X}$, that received a negative outcome $f(x_u) < \tau$, from a black-box classifier $f : \mathcal{X} \rightarrow [0, 1]$, find a set of

¹Our formulation is also applicable when f is a multi-class classifier by employing the one-vs-all technique. In the opposite classification fashion, $1 - f(x_u)$ can simply be used as the classifier

$k \in \mathcal{N}^+$ counterfactual data point $C = \{x^{(1)}, \dots, x^{(k)}\}$ such that

$$\begin{aligned}
& \max_{x^{(i)}} \Upsilon(x^{(i)}, x_u) && \text{proximity} \\
& && \forall i \in \{1, \dots, k\} \\
& \min_{x^{(i)}} \Gamma(x^{(i)}, x_u) && \text{sparsity} \\
& && \forall i \in \{1, \dots, k\} \\
& \max_{x^{(i)}, x^{(j)}} \Delta(x^{(i)}, x^{(j)}) && \text{diversity} \\
& && \forall i, j \in \{1, \dots, k\}, i \neq j \\
& \max_{x^{(i)}} \Pi_u(x^{(i)}, x_u) && \text{preference} \\
& && \forall i \in \{1, \dots, k\} \\
& \text{s.t. } f(x^{(i)}) \geq \tau && \text{validity} \\
& && \forall i \in \{1, \dots, k\}
\end{aligned} \tag{4.1}$$

where $\Upsilon, \Gamma, \Delta, \Pi_u : \mathcal{X}^2 \rightarrow [0, 1]$.

Since in the recourse setting, the user's preference is initially unknown, Problem 1 cannot be solved as is. Instead, the user's preference needs to be estimated by querying user preferences (Section 4.1.5).

Counterfactual Metrics - Despite the generality of Problem 1 which could ideally adopt different definitions, in this paper we adopt definitions which are widely used in the counterfactual literature:

- **Proximity** can be defined as the Euclidean norm between a counterfactual x' and the original value x :

$$\Upsilon(x', x) = \frac{1}{\|x' - x\|_2 + 1}$$

This is inverted to map it to 0, 1 and to align with maximization.

- **Sparsity**, representing the number of modified features, can be expressed using the zero norm:

$$\Gamma(x', x) = \|x' - x\|_0$$

Since this norm is non-differentiable, it is preferable to use the absolute-value norm $\|\cdot\|_1 = |\cdot|$.

- **Diversity**, as in [196] we use a distance between the generated counterfactual, that is the cosine distance $\forall x^{(i)}, x^{(j)} \in C, i \neq j$:

$$\Delta(x^{(i)}, x^{(j)}) = 1 - \frac{x^{(i)} \cdot x^{(j)}}{\|x^{(i)}\|_2 \|x^{(j)}\|_2}$$

4.1.4 Conclusions

In this study, we aim to introduce an additional metric that replaces the traditional concept of validity, which, as a reminder, is defined as $f(x^{(i)}) \geq \tau$. With the introduction of user preferences, it becomes imperative to redefine the notion of a *valid* counterfactual. After all, if the user does not *accept* the counterfactual, can it truly be considered valid? Building upon this premise, we introduce the following measure.

Definition 4.1.4.1 (Personal Validity). Given a user $u \in \mathcal{U}$, with $x_u \in \mathcal{X}$ and the user preference probability $\Pi_u(x'_u) = P(x'_u | x_u, u)$, the Personal Validity for a counterfactual recourse $x'_u \in \mathcal{X}$ on

the classifier $f : \mathcal{X} \times \mathcal{X} \rightarrow [0, 1]$ is defined as:

$$PV(x'_u) = \Pi_u(x'_u) \cdot f(x'_u)$$

This novel metric preserves the properties of both Validity and Preference. The non-binary nature of the recourse probability captures the nuances between full acceptance and rejection of a counterfactual, thus providing a more detailed measure of validity compared to the traditional binary definition.

4.1.5 Framework

In this section, we present our iterative algorithm (Section 4.1.5), dubbed **HIP-CORE** (Human-In-the-Loop COUNTERfactual REcourse), designed to estimate user preference Π_u (Section 4.1.5) while generating candidate counterfactuals C (Section 4.1.5). Our approach is agnostic to both model and data, making it applicable for generating personalized counterfactual recourse with any black-box model.

Personalized Counterfactual Generation

Given the complexity of solving a multi-objective problem such as Problem 1, that in some setting has been show to be NP-hard [143], we transform Problem 1 into a single-objective problem, by considering a linear combination of the metrics, with the signs appropriately inverted for those metrics that are to be minimized (i.e., sparsity). Additionally, the constraint of the class flipping (i.e. score) is directly incorporated, removing the dependence on τ . Consequently, we provide the following single-objective problem.

Problem 2 (Relaxed Personalized Counterfactual Recourse Problem). Given the same setting as in Problem 1, the problem can be relaxed as follows:

$$\begin{aligned} \max_C \quad & \frac{1}{k} \sum_{i=1}^k [\lambda_{\Upsilon} \Upsilon(x^{(i)}, x) + \lambda_{\Gamma} (1 - \Gamma(x^{(i)}, x)) + \\ & + \lambda_{\Pi} \Pi(x^{(i)}, x) + \lambda_f f(x^{(i)}) + \\ & + \lambda_{\Delta} \frac{1}{k-i-1} \sum_{j=i+1}^k \Delta(x^{(i)}, x^{(j)})] \end{aligned} \quad (4.2)$$

where $\Upsilon, \Gamma, \Delta, \Pi : \mathcal{X}^2 \rightarrow [0, 1]$, $\lambda_{\Upsilon}, \lambda_{\Gamma}, \lambda_{\Delta}, \lambda_{\Pi}, \lambda_f \in [0, 1]$, such that $\lambda_{\Upsilon} + \lambda_{\Gamma} + \lambda_{\Delta} + \lambda_{\Pi} + \lambda_f = 1$

A counterfactual can be generated by solving the presented maximization problem. To effectively address this, several optimization algorithms can be employed.

The coefficients λ in Equation 4.2 allow for adjusting the importance assigned to individual metrics. They generate a challenging trade-off, between the user-agnostic counterfactual properties (i.e. score, proximity, sparsity, diversity) and the preference that is user-centered. In the experimental evaluation, we discuss the implication of the trade-off.

Counterfactual Preference Modeling

In this section, we introduce a set of assumptions to facilitate the modeling of preference, along with the resulting theorems.

Assumption 6. *The preference Π_u of a user $u \in \mathcal{U}$ remains stable in the explanation process.*

Introducing a temporal component to the problem is not a straightforward task because it would require considering users u who change both their instances x_u and their preferences Π_u over time [87]. Consequently, the same counterfactuals generated may no longer be valid at different times. For this reason, in the current work, we will not account for the temporal component.

Assumption 7. *For all users $u \in \mathcal{U}$, there exists a counterfactual explanations $x' \in \mathcal{X}$, such that the user's preference $\Pi_u(x')$ is equal to 1.*

Enforcing the preference to have a value of 1 allows us to evaluate preference as if it were a normalized metric, thus facilitating a better assessment of the quality of a counterfactual and determining if the preference optimum (1) has been reached.

Assumption 8. *The preference $\Pi_u(x')$ is maximal when $x' = x$.*

This assumption is based on the idea that users tend to maintain their current state, making the maximum preference corresponding to minimal state change. However, since they aim to flip their classification, they are willing to yield, take actions that move them away from their current state, thereby reducing their initial preference. While real-world scenarios may deviate from this pattern, we attribute such deviations to unaccounted factors in our current modeling, such as the passage of time; we defer addressing these factors to future work.

Theorem 5.1. *Let $\pi_u : \mathcal{X} \rightarrow [0, 1]$ a probability distribution. Then, $\Pi_u(x') = \frac{\pi_u(x')}{\max_{x'} \pi_u(x')}$ represents a model for a preference of a user $u \in \mathcal{U}$.*

Proof. In order for $\Pi_u(x')$ to be a preference, we need to check that it respects the above two assumptions. To check Assumption 4.1.3.1 it suffices to observe that $\pi_u(x') \in [0, \max_{x' \in \mathcal{X}} \pi_u(x')]$ therefore $\Pi_u(x') \in [0, 1]$. To check Assumption 7 note that $\Pi_u(x^*) = 1$ for the counterfactual x^* such that $\pi_u(x^*) = \max_{x'} \pi_u(x')$, implying that $\Pi_u(x^*) = 1$. \square

The above theorem allows us to perform sampling of counterfactuals, with the probability directly proportional to the user's preference value.

Assumption 9. *The preference Π_u of a user $u \in \mathcal{U}$ is feature-independent and, in particular, we assume that $\Pi_u(x') = P(x'|x, u) = \frac{1}{n} \sum_{i=1}^n P(x'_i|x_i, u)$, where x_i is the value of feature i .*

We assume independence among individual features to describe the joint preference Π_u as a product of single-feature preferences. However, we opt for summation instead of multiplication to address the issue of a preference value of 0, i.e., $P(x'_i|x_i, u) = 0$. This will avoid $\Pi_u(x') = 0$, making preference estimation impossible.

Given the independence of preference Π_u from features, we can introduce two extreme scenarios, represented by features for which the user has no desire or ability to change (e.g., place of birth) or holds no specific preference.

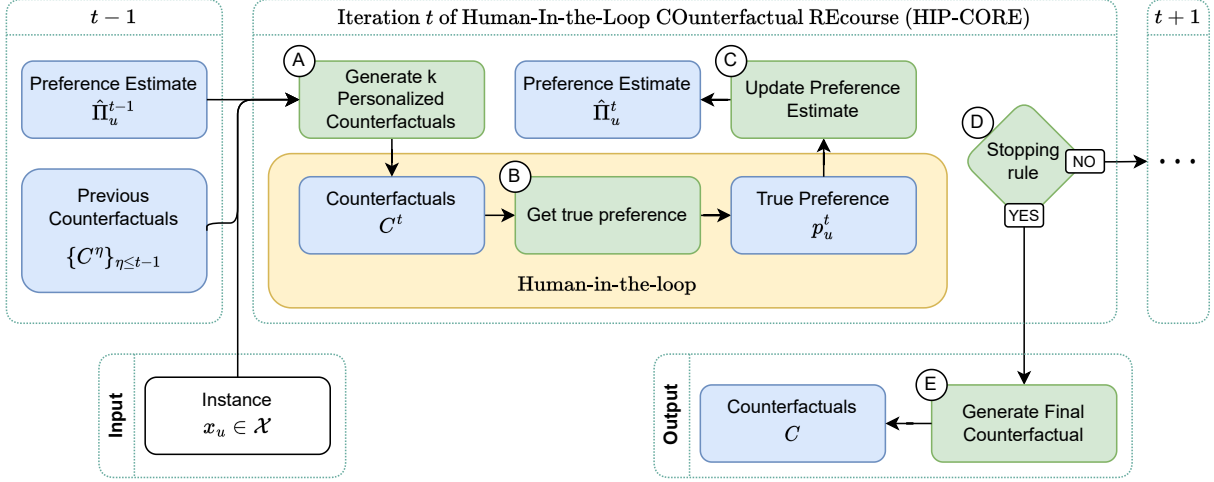


Figure 4.2: Iteration t of HIP-CORE.

Theorem 5.2. *If a user u has no intention to or can't change a feature i , their preference $\Pi_u(x'_i)$ can be modeled using a degenerate probability distribution π_u over x_i , such that: $\Pi_u(x'_i) = \begin{cases} 1 & \text{if } x'_i = x_i \\ 0 & \text{otherwise} \end{cases}$*

Proof. If a user u is unwilling to change a feature i , it is natural to assume that $\Pi_u(x'_i) = 0$ for all $x'_i \neq x_i$. Referring back to Definition 4.1.3.1, we can rewrite $\Pi_u(x'_i) = \frac{\pi_u(x'_i)}{\max_{x'_i} \pi_u(x'_i)}$. This leads to $\pi_u(x'_i) = 0$ for all $x'_i \neq x_i$. Since π_u is a probability distribution and must satisfy the constraint $\sum_{x'_i \in \mathcal{X}} \pi_u(x'_i) = 1$, it follows that $\pi_u(x_i) = 1$, and thus $\Pi_u(x_i) = 1$. This also agrees with Assumption 8, as $\Pi_u(x'_i)$ indeed attains its maximum value at $\Pi_u(x_i)$. \square

Theorem 5.3. *If a user u has no preference for changing feature i , their preference $\Pi_u(x'_i)$ can be modeled using a uniform probability distribution π_u : $\Pi_u(x'_i) = 1 \quad \forall x'_i \in \mathcal{X}_i$.*

Proof. If π_u is a continuous distribution, $\pi_u(x'_i) = \frac{1}{|\mathcal{X}_i|} \quad \forall x'_i \in \mathcal{X}_i$ (we are not considering the case when \mathcal{X}_i is an infinite set). Since that same value is also the maximum of π_u , we would get $\Pi_u(x'_i) = 1$. \square

Expanding to non-extreme cases, where preference is essentially estimated by definition, as there are no unknown parameters to estimate, we can assume that preference Π_u is, in fact, dependent on an unknown set of parameters $\vec{\theta}$. For instance, in the following section, we can define preferences for continuous features.

Theorem 5.4. *If a feature i is continuous, the preference $\Pi_u(x'_i)$ can be modeled using a normal distribution π_u with mean $\theta_1 = x_i$ and variance θ_2 , such that: $\Pi_u(x'_i) = e^{-\frac{1}{2} \left(\frac{x'_i - x_i}{\theta_2} \right)^2}$*

Proof. Referring back to Definition 4.1.3.1, we can write $\Pi_u(x'_i) = \frac{\pi_u(x'_i)}{\max_{x'_i} \pi_u(x'_i)}$. If π_u follows a normal distribution, it has a mean $\theta_1 = x_i$ according to Assumption 8. Thus, it can be expressed as $\pi_u(x'_i) = \frac{1}{\theta_2 \sqrt{2\pi}} e^{-\frac{1}{2} \left(\frac{x'_i - x_i}{\theta_2} \right)^2}$. Since the maximum is reached at $x'_i = x_i$, i.e., $\max_{x'_i} \pi_u(x'_i) = \pi_u(x_i) = \frac{1}{\theta_2 \sqrt{2\pi}}$, we obtain $\Pi_u(x'_i) = e^{-\frac{1}{2} \left(\frac{x'_i - x_i}{\theta_2} \right)^2}$. \square

Theorem 5.4 proves that it's not critical to estimate the position of preference, as it is always centered around the current value x_i . What matters, instead, is the variance θ_2 , which directly models the user's willingness to deviate from the current value x_i .

Theorem 5.5. *If a continuous feature i can only increase², the preference $\Pi_u(x'_i)$ can be modeled using an exponential distribution with rate θ , such that: $\Pi_u(x'_i) = \begin{cases} e^{-\theta(x'_i - x_i)} & \text{if } x'_i \geq x_i \\ 0 & \text{otherwise} \end{cases}$*

Proof. If π_u follows an exponential distribution, we can write: $\pi_u(x'_i) = \begin{cases} \theta e^{-\theta(x'_i - x_i)} & \text{if } x'_i \geq x_i \\ 0 & \text{otherwise} \end{cases}$

Noting that the maximum value $\pi_u(x'_i) = \pi_u(x_i) = \theta$ does not violate Assumption 8, we can write $\Pi_u(x'_i) = \frac{\pi_u(x'_i)}{\theta}$, following from Definition 4.1.3.1. \square

Theorem 5.6. *If a feature i is categorical with K categories, the preference $\Pi_u(x'_i)$ can be modeled using a categorical distribution π_u with parameters $\theta_1, \dots, \theta_K$, such that: $\Pi_u(x'_i) = \frac{\theta_k}{\theta_{x_i}} \quad \forall k \in \{1, \dots, K\}$.*

Proof. Given that $\pi_u(x'_i) = \theta_k \forall k \in \{1, \dots, K\}$, based on Assumption 8 we derive that $\max_{x'_i} \pi_u(x'_i) = \theta_{x_i}$. Consequently, we obtain $\Pi_u(x'_i) = \frac{\pi_u(x'_i)}{\theta_{x_i}}$, which attains value of 1 if $x'_i = x_i$, satisfying Assumption 7. \square

Preference Estimation

In the recourse setting, there is no access to Π_u , and we cannot invoke it at will. Furthermore, there could be a maximum number of feasible interaction to ask the user's preferences. Therefore, it is fundamental to be able to estimate Π_u .

The preference can be estimated by solving the following system of equations:

Definition 4.1.5.1. Given a user $u \in \mathcal{X}$, a set of preference values p_u and a set of counterfactuals C , preference can be estimated by solving the following system of equations in $\vec{\theta}$:

$$\hat{\Pi}_u(x^{(i)}|\vec{\theta}) = p_u^{(i)} \quad \forall i \in \{1, \dots, |C|\} \quad (4.3)$$

Solving this problem depends on both the quantity of generated counterfactuals $|C|$ for which true preferences p_u are available and the number of parameters $\vec{\theta}$ that comprise Π_u . These parameters are contingent on how the preference is defined, as exemplified in Theorems 5.2, 5.3, 5.4, 5.5 and 5.6. Although we defined Π_u as feature-independent, it does not simplify the problem because each $P(x'_i|x_i, u)$ can be represented by a nonlinear function. In our experiments, we solved this problem by minimizing the mean squared difference between $\Pi_u(x^{(i)}|\vec{\theta})$ and $p_u^{(i)}$ for all $i \in 1, \dots, |C|$ using the Powell Method [222] at each iteration t . We initialized the parameters at each iteration using the estimates from the previous iteration θ^{t-1} .

HIP-CORE Framework

Figure 4.2 provides a graphical schematization of the functioning of HIP-CORE, while its pseudocode is provided in Algorithm 1. At an high-level, at each iteration t , HIP-CORE refines the estimate of user

²The extension to non-increasing features is trivial.

preference $\hat{\Pi}_u^t$ with the human-in-the-loop true preference p_u^t , while generating more personalized counterfactual recourse C^t .

Algorithm 1 HIP-CORE

Require: a user identifier $u \in \mathcal{U}$, a user feature point $x \in \mathcal{X}$; a classifier f ; a number of counterfactual generated at each iteration $k \in \mathbb{N}^+$; a maximum number of iterations $T \in \mathcal{N}^+$.

Ensure: A personalized counterfactual recourse $\{x'\}$ and an estimation of user preference $\hat{\Pi}_u$

- 1: $\hat{\Pi}_u^0 \leftarrow g : \mathcal{X} \rightarrow 1$ s.t. $g(x) = \frac{1}{|\mathcal{X}|}$ if $x \in \mathcal{X}$ else 0 Initialize the estimate of user preferences;
 - 2: $C^0 \leftarrow \{\}$ Initialize counterfactuals' set
 - 3: $t \leftarrow 1$;
 - 4: **while** ($t \leq T$) **do**
 - 5: $C^t \leftarrow \text{get_c}(x, \hat{\Pi}_u^{t-1}, k, \{C^\eta\}_{\forall \eta < t})$ Generate counterfactuals;
 - 6: $p_u^t \leftarrow \{\Pi_u(x')\}_{\forall x' \in C^t}$ Ask user preference;
 - 7: $\hat{\Pi}_u^t \leftarrow \text{update_pref}(\hat{\Pi}_u^{t-1}, \{p_u^\eta\}_{\forall \eta \leq t}, \{C^\eta\}_{\forall \eta \leq t})$ Update preference estimation;
 - 8: $t \leftarrow t + 1$
 - 9: $C \leftarrow \text{get_c}(x, \hat{\Pi}_u^T, 1)$ Generate final counterfactual;
 - 10: **return** $C, \hat{\Pi}_u^T$
-

Initialization - The initialization of $\hat{\Pi}$ is defined as uniform over the entire set \mathcal{X} . However, if some data are available, it could be initialized as uniform over all data points in the dataset or only for those where the counterfactual is valid.

Iteration t - At each iteration t , the algorithm receives as input the original instance $x \in \mathcal{X}$, the set of counterfactuals generated in the previous iterations $\{C^\eta\}_{\forall \eta < t}$, and the estimated user preference $\hat{\Pi}^{t-1}$.

At each iteration t HIP-CORE performs the following steps:

(A) A set of $k \in \mathcal{N}^+$ personalized counterfactual recourse C^t are produced with get_c (check Section 4.1.5).

(B) C^t is provided to the user for a human-in-the-loop interaction and he expresses the preferences over the counterfactuals. At this stage, the true preferences $p_u^t = \{\Pi_u(x')\}_{\forall x' \in C^t}$ of the user are stored.

(C) The algorithm updates the estimate of the user preference $\hat{\Pi}_u^t$ given the new true preferences p_u^t with update_pref (check section 4.1.5).

(D) The stopping rule is a straightforward maximum number of iterations T . Alternatively, it may depend on other factors, such as whether the generated counterfactuals match those from the previous iteration.

(E) Once the stopping criteria are met, the final personalized counterfactual $C = \{x'\}$ is generated.

Limitations

Assumption 9 of feature-independence is often an oversimplification. If there are dependencies between features, modeling the joint distribution becomes more complex. For instance, the joint probability $\Pi_u(x)$ would not simply be the product of each $P(x'_i)$. Instead, you'd need to model the conditional probabilities such as $P(x'_i|x'_j) \quad \forall i, j$.

To model these dependencies, one might consider Bayesian Networks, where nodes represent features and directed edges indicate conditional dependencies, or multivariate distributions, such as

multivariate Gaussian for multiple continuous features.

Expressing a joint distribution with dependencies explicitly can be quite complex, especially for high-dimensional feature sets. Often, it requires specific modeling choices based on the nature and relationships of the features in question.

Nevertheless, HIP-CORE is more general and applies beyond the assumptions we have made in Sec. 4.1.5.

Table 4.1: Comparison of HIP-CORE and baseline model performance. The direction of arrows indicates what is considered the best performance: \uparrow/\downarrow denotes that higher/lower values are better. Best-performing values in each category are highlighted in **bold**.

Dataset	Model	Validity(\uparrow)	Preference(\uparrow)	Sparsity(\downarrow)	Proximity(\uparrow)	Personal Validity(\uparrow)
Adult Income	HIP-CORE	0.904	0.386 \pm 0.107	0.695 \pm 0.136	0.945 \pm 0.064	0.317 \pm 0.150
	Baseline	0.943	0.346 \pm 0.108	0.757 \pm 0.135	0.975 \pm 0.036	0.302 \pm 0.099
GiveMeSomeCredit	HIP-CORE	0.585	0.053 \pm 0.006	0.759 \pm 0.143	0.970 \pm 0.152	0.030 \pm 0.027
	Baseline	0.002	0.0 \pm 0.001	1.000 \pm 0.007	0.970 \pm 0.152	0.0 \pm 0.0
HELOC	HIP-CORE	0.515	0.070 \pm 0.043	0.880 \pm 0.086	0.702 \pm 0.254	0.037 \pm 0.050
	Baseline	0.342	0.031 \pm 0.047	0.925 \pm 0.117	0.761 \pm 0.261	0.005 \pm 0.008

Experimental setting

4.1.6 Experiments

To evaluate HIP-CORE, we define an experimental setting as follows. Each user $u \in \mathcal{U}$ is described by a user feature data $x_u \in \mathcal{D}$ and the true user-preference distribution Π_u is simulated as described in Section 4.1.5; more details can be found in the Supplementary Materials. To get the feature data \mathcal{D} , we used existing real-word datasets: Adult [26], GiveMeSomeCredit [58] and HELOC [126]. We used a black-box classifier f based on xgboost [48] and trained on an appropriate subset of the complete data.

To solve the personalized counterfactual recourse step of HIP-CORE, as defined in Problem 2, as well as the preference estimation step, as defined in Problem 4.1.5.1, we have chosen to employ the Powell’s method [222]. Furthermore, we have run a randomized search for the λ parametrization in Equation 4.2, to explore the trade-off between the different properties.

The experiment are performed for the tested dataset with two distinct setting: one user-agnostic (the baseline), i.e. $\lambda_{\Pi} = 0$, and one including the preference, i.e. $\lambda_{\Pi} > 0$, to highlight the importance of using preference in generating personalized recourse. The results are shown for the combination of λ parameters that achieves the maximum Personalized Validity.

Results

In Table 4.1, we report the main results of our experiments for the tested dataset for HIP-CORE with preference and a user-agnostic version. All metrics are improved by the HIP-CORE across all datasets, with the exception of the proximity, and validity on one dataset.

Sparsity value is decreased, meaning that on average less features are modified by HIP-CORE: we generated more concise counterfactual recourse using features that the user prefers. Proximity is slightly decreased compared to the baseline. However, given that proximity is a data-driven measure

that do not consider the subjective and potentially irrational user preference, we encourage the community to increase the relevance of the user preference with respect to the proximity.

Finally, the preference is substantially enhanced by HIP-CORE compared with the baseline. This underscores the importance of including the preference in counterfactual recourse generation process.

Discussion and Ethical Implications

In the new preference-based framework, traditional metrics like Sparsity and Proximity have undergone a significant transformation. Previously, they served as automatic methods to gauge a rational user’s preference. However, when applied in this new setting, they risk providing solutions that may not align with the user’s actual preferences. So, with the introduction of a more realistic modeling of user preference and Personal Validity, these metrics become outdated.

When considering the ethical implications of our work, several key aspects deserve attention.

- **Privacy and Data Handling:** Users have the option to keep their preferences confidential, but expressing preferences accurately is important for optimal recourse. Failing to provide preferences can affect preference estimation and recourse quality. The algorithm should prioritize data security, not retaining user data beyond creating recourse, to ensure user privacy.
- **The presence of bias or unfairness in the treatment of features within the model hinges on its design.** To enhance fairness, a null preference for specific features can be integrated, addressing potential bias or unfairness in the approach.

The broader issue of ethics in counterfactuals is multifaceted. However, we maintain that it falls beyond the scope of our current work. Our primary focus is on the development of a methodology rather than the creation of an operational product. The assurance of ethical practices ultimately hinges on the specifics of implementation.

In this study, we introduced HIP-CORE (Human-In-the-Loop Preference COUNTERfactual REcourse) to incorporate user preference in the generation of counterfactual recourse through a human-in-the-loop process. We have formalized the modeling of preference, positioning it as a fundamental property in the creation of personalized counterfactuals. Acknowledging that user preference is not known a priori, we have mathematically formalized the estimation of user preferences, establishing a foundation for new opportunities in this area.

In future works, we plan to further investigate the mathematical implication of the modeling and the estimation of the user preference in the counterfactual recourse setting. For instance, we want to provide a more comprehensive analysis of the preference estimate, considering more specific types of features, and exploring scenarios where the problem might have solutions, and of which type (unique or multiple solutions might exist). Furthermore, we intend to challenge the assumption of feature independence, delving into potential feature interactions. We will also explore modeling preference and, consequently, counterfactual recourse while considering the element of time.

In conclusion, we earnestly believe that this study underscores the paramount importance of considering users and their preferences when generating recourse. We hope this could serve as an encouragement for the counterfactual recourse community to adopt our proposed modeling approach and incorporate user preferences into the counterfactual recourse framework.

4.2 Deep Active Learning for Misinformation Detection Using Geometric Deep Learning

Human fact-checkers currently represent a key component of any semi-automatic misinformation detection pipeline. While current state-of-the-art systems are mostly based on geometric deep-learning models, these architectures still need human-labeled data to be trained and updated - due to shifting topic distributions and adversarial attacks. Most research on automatic misinformation detection, however, neither considers time budget constraints on the number of pieces of news that can be manually fact-checked, nor tries to reduce the burden of fact-checking on - mostly pro bono - annotators and journalists. The first contribution of this work is a thorough analysis of active learning (AL) strategies applied to Graph Neural Networks (GNN) for misinformation detection. Then, based on this analysis, we propose Deep Error Sampling (DES) - a new deep active learning architecture that, when coupled with uncertainty sampling, performs equally or better than the most common AL strategies and the only existing active learning procedure specifically targeting fake news detection. Overall, our experimental results on two benchmark datasets show that all AL strategies outperform random sampling, allowing – on average – to achieve a 2% increase in AUC for the same percentage of third-party fact-checked news and to save up to 25% of labeling effort for a desired level of classification performance. As for DES, while it does not always clearly outperform other strategies, it still reduces variance in the performance between rounds, resulting in a more reliable method. To the best of our knowledge, we are the first to comprehensively study active learning in the context of misinformation detection and to show its potential to reduce the burden of third-party fact-checking without compromising classification performance.

4.2.1 Introduction

Since the 2016 United States presidential elections, both the general public and the scientific community have become increasingly aware of the threat posed to democracies by the spread of online misinformation. Research on misinformation detection has then experienced significant momentum, with many websites and independent journalists starting to fact-check online news, and releasing new datasets on which automatic detection systems can be trained. Almost at the same time, research on graph neural networks (GNNs) started reaching remarkable results in node and graph classification [30, 115, 152, 194, 301]. GNNs are made up of several layers of interconnected nodes, where each node represents a vertex in the graph and each edge represents a connection between two vertices. The nodes in the GNN are able to communicate with one another through these edges, allowing the GNN to process and analyze the graph as a whole, rather than just individual nodes. This makes GNNs well-suited for tasks that require understanding the relationships and dependencies between different elements in the graph.

GNNs have enabled scientists to better model news diffusion patterns in social networks, thus moving away from simple text-based fake news detection pipelines. In a nutshell, state-of-the-art GNN-based misinformation detection methods try to classify graphs that represent URL cascades in

social networks. Despite the promising improvement in the performance of GNN-based architectures for fake news detection, in order to train these models, researchers still need high-quality third-party fact-checked news articles that are difficult and expensive to obtain. This problem is further amplified in large social networks and on the web, where the volume of news produced and spread daily makes extensive annotation virtually impossible. Indeed, manual data annotation consists of manually labeling and adding metadata to data, typically for the purpose of training machine learning models, and is a general pain point for most deep learning research due to its high costs - both in terms of human labour and time. In our case, while such scarcity of fake news data makes the problem of efficient annotation particularly urgent, research on misinformation detection under labeling constraints is still very scarce. In previous work, the need to reduce the human effort required to manually label news as fake or authentic has been largely ignored.

Active learning [158, 193] is a machine learning approach in which a model is able to interactively query the user (or some other information source) to obtain the desired output, rather than being solely trained on a fixed dataset. In active learning, the model initially starts with a small amount of labeled data and makes predictions on the rest of the data. The model then selects a subset of the data for which it is least confident in its predictions, and asks the user to label this data. The labeled data is then used to update the model, and the process is repeated until the model reaches a satisfactory level of performance. In this work, we then present the first in-depth analysis of active learning (AL) strategies for fake news detection. We also propose Deep Error Sampling (DES) - a new deep-learning method that, when used in conjunction with uncertainty sampling, performs better, on average, than the most common AL strategies, including the only proposed active learning principle specifically targeting fake news detection. All tested active learning strategies were applied to three state-of-the-art GNN-based misinformation classifiers. As for the datasets, we performed experiments on PolitiFact [264] and FbMultiLingMisinfo [22], two high-quality and human-labeled collections of real and fake news. While the former is smaller and only contains news written in English, the latter is more recent, larger, and composed of URLs pointing to news in several languages. Overall, compared to random sampling, the best AL strategies allow to achieve a 2% increase in AUC for the same percentage of third-party fact-checked news and to save up to 25% of labeling effort for a desired level of classification performance.

To sum up, our original contributions are the following:

- An in-depth analysis of active learning (AL) strategies in the context of automatic misinformation detection;
- We showed that, in the context of misinformation detection, active learning represents a viable and convenient strategy to increase the AUC classification metric by up to 5% and to reduce the cost of news labeling up to 25% for a given level of desired performance;
- Deep Error Sampling (DES), a new deep active learning architecture that, when coupled with uncertainty sampling, performs equally or better than the most common AL strategies and the only proposed active learning procedure specifically targeting fake news detection;
- In particular, while other active learning strategies allow to reach results similar to DES, overall Deep Error Sampling shows lower variance between rounds and can be considered a more robust method.

To the best of our knowledge, no previous deep active learning method has leveraged prediction errors as the main discriminative signal. As shown in the experimental section, its characteristics seem to match well with both uncertainty and diversity sampling, paving the way for new combinations of more robust active learning strategies.

4.2.2 Related Work

In this section, we first review the current state-of-the-art misinformation detection models that leverage geometric deep learning, we then go through the most common active learning strategies, with a focus on deep active learning, and finally, we briefly present recent finding on fake news benchmark datasets to justify our experimental choices.

Misinformation Detection Methods

Misinformation detection is not only challenging, but also necessary. As shown in a seminal work by Vosoughi et al. [305], in social networks fake news spreads faster and more extensively than high-quality information. Over the past five years, GNN-based methods have established themselves as the state-of-the-art approach in the fight against fake news. Unlike their predecessors, which were mostly content-based, these methods leverage the diffusion patterns of news in social networks as the main signal. These patterns are not merely features representing the spreading patterns of the news that are appended to content-based features to train traditional machine learning classifiers. Instead, the task is now formulated as a node [44, 177, 199, 238, 275, 337] or a graph classification task [116, 194, 278], using methods such as Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs) [173]. State-of-the-art methods use either node or graph embeddings obtained by training a geometric deep learning architecture on an appropriate graph. The most commonly used architectures include Graph Convolution Networks (GCN) [74, 177, 194], Bi-Directional Graph Convolution Networks (BiGCN) [30], Graph Attention Networks [74, 131, 236, 238], and GraphSAGE [74, 116]. Depending on the approach, these representations can be further combined with text-based features and/or with non-GNN-based embeddings that capture other aspects of fake news [177].

Convenient APIs offered by Twitter, which can be used for research purposes, have turned this platform into the *de facto* standard for testing and validating misinformation detection methods [177, 194, 278]. Typically, in graph-based representations used for misinformation detection, nodes correspond to either news articles [44, 199, 238, 263, 275, 341] or to users [44, 116, 177, 194, 199, 263, 275, 337, 338]. In other cases, content creators [44, 199, 238, 263, 338, 341] or article authors or sources are included as additional nodes [44, 199], and less often, nodes represent topics [238, 341] or comments [337]. Regarding edges, news articles can be directly connected to their authors [44, 199, 238, 263, 341], topic(s) [238, 341], or to users who post/share them [199]. Users, in turn, can be linked through their social graph, e.g., based on following or friendship relationships [44, 194, 199], re-posting activity [275], replies [278], or (posted) content similarity [275]. Moreover, users can be connected to their posts [263, 337], to an article through a stance score [44, 199], or to nodes representing their posted comments [337], which in turn are usually connected to their corresponding post [337]. As for news-posting URL hostnames/domains, an edge can be added every time two hostnames/domains link to each other [44, 199].

Finally, going more in depth into some of the most remarkable contributions, it is worth highlighting the following successful choices. Ren et al. [238] propose a novel hierarchical attention mechanism to perform node representation learning in heterogeneous information networks that effectively tackles fake news detection. They also use an active learning framework to enhance learning performance, especially when facing the paucity of labeled data. Yu et al. [337] aggregate multi-type information in a hierarchical manner and the information can reason over heterogeneous graph for the facticity of the news. Shu et al. [263] propose a tri-relationship embedding framework TriFN, which models publisher-news relations and user-news interactions simultaneously for fake news classification. The system is made up of 5 components, all based on some form of matrix decomposition and factorization. Finally, for each URL, Monti et al. [194] searched for all the related cascades and enriched their Twitter-based characterization (users and tweet data) by drawing edges among users according to Twitter’s social network.

Active Learning

Broadly speaking, AL refers to the iterative selection and labeling of samples to train a supervised classification model with the goal of reducing the number of labeled data points required to reach a desired performance. As extensively reviewed in Monarch [193] and Kumar and Gupta [158], the earliest and still most common AL strategies are variations of uncertainty sampling and diversity sampling. Uncertainty sampling prioritizes the items that the current model is most uncertain about, at the risk of selecting multiple similar, redundant samples. Diversity sampling counteracts this problem by exploiting the fact that data points are usually clustered in feature space, and prioritizes centroids and out-layers. In practice, a combination of uncertainty and diversity sampling generally outperforms random sampling, and can be adapted to work in an online setting [178] and/or with highly unbalanced classes [59, 155].

When complex deep learning architectures are deployed, however, standard AL strategies could under-perform due to the known problem of overconfidence of deep learning models. Indeed, the soft-max function is often used in the output layer of a neural network to convert the network’s output into a probability distribution. It does this by exponentiating the output of each unit in the output layer, normalizing the resulting values, and then mapping the exponentiated outputs to a probability distribution. It follows that, when the network has learned to make very confident predictions (i.e., the output of a unit is much larger than the output of the other units), the soft-max function will map these outputs to a very high probability. This can happen, for instance, when training data is very unbalanced or when the model is used on out-of-domain samples. For this reason, new AL strategies are specifically designed to work in the deep learning context [235]. This branch of research is sometimes referred to as deep active learning. Recently, some works have also specifically targeted active learning in graphs and graph neural networks. Madhawa and Murata [181] have studied the application of active learning on attributed graphs. They show that algorithms designed for other data types do not perform well on graphs. In Liu et al. [171], after showing that state-of-the-art AL algorithms do not properly work on attributed graphs, a new latent space clustering-based active learning method for node classification (LSCALE) is proposed. Finally, in Madhawa and Murata [181], a novel framework to address the challenge of active learning in large-scale imbalanced graph data (node classification) is presented.

As for active learning in misinformation detection, the scientific literature still lags behind -

with very few contributions. Ren et al. [238] use an active learning framework to enhance learning performance of their novel hierarchical attention mechanism. Bhattacharjee et al. [29], on the other end, propose a human-machine collaborative learning system to evaluate the veracity of a news content, with a limited amount of annotated data samples. In this work, we directly compare our Deep Error Sampling (DES) strategy against the active learning component of Ren et al. [238] - named here Deep Unseen Sampling (DUS). As for [29], we decided not to include this method in our analysis for two reasons: 1. the active learning component of the pipeline is very similar to Ren et al. [238], and 2. the whole workflow was optimized for a lexical-based fake news detector.

Fake News Datasets

The robustness of misinformation detection research depends on the quality of the data used to conduct experiments, but we find that fake news benchmark datasets are often small and contain biases that affect the results (few thousand not-randomly-sampled fact-checked URLs). A relatively large dataset coming from the fact-checking website gossipcop.com, and a smaller one sampled from politifact.com - both released as part of FakeNewsNet [264] - constitute two of the most commonly used benchmark datasets [265, 266]. While GossipCop still represents the largest fake news detection benchmark dataset, its real discriminative power has been recently put into question [22]. Indeed, GossipCop has proven to be exceptionally easy to classify and thus of limited utility to assess the discriminatory power of misinformation detection methods. For this reason, we decided not to include it in our experiments. Other common sources of annotated URLs or posts include BuzzFeed [355], Twitter [330] and Weibo [160, 173].

These datasets for benchmarking fake news detection have reliable labels, but tend to include news in a single language, and to be created following unknown selection criteria – see, e.g., a recent in-depth review of these datasets [77]. Moreover, they are usually quite easy to classify. Larger datasets, such as NELA, can be created by sampling news from notoriously reliable and unreliable sources using distant supervision [108, 202]. However, they are also noisy and biased since news articles are labeled as true or false according to their source, and are not individually fact-checked. Recently, a new multilingual benchmark dataset for misinformation detection was published [22]. This dataset comes from the recently published Facebook Privacy-Protected Full URLs Data Set [189], which comprises all 36 million URLs publicly shared on Facebook at least 100 times between January 2017 and July 2019, and includes fact-checking labels for 7334 of these URLs.

4.2.3 Problem Statement

We consider a collection U of unlabeled news items (news articles/URLs) that we want to categorize as real news or fake with the highest possible accuracy. Since human labeling is both expensive and time-consuming, we assume that we are allowed to annotate only b news pieces. In other words, only the subset $B \subset U$, with $|B| = b$, will be sent to annotators. The quantity b represents a budget of possible annotations. We can also define b as a fraction of the size of U . Furthermore, we assume that each annotation has a unit cost. Using this labeled news, we train an automatic misinformation detection system, which we will leverage, in turn, to annotate the remaining unlabeled news $U \setminus B$. The budget b cannot be too low because it would not allow training a good classifier, but it cannot be too large because, in most practical cases, it would be unfeasible to send each news item for

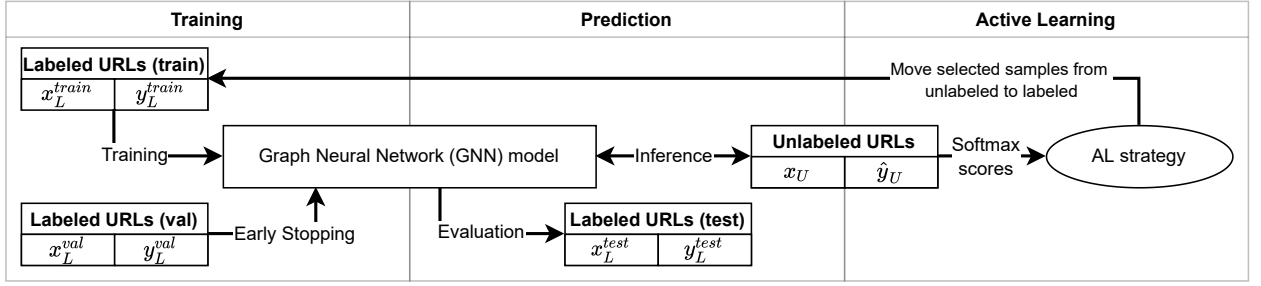


Figure 4.3: Pipeline for "shallow" Active Learning strategies. First the GNN model is trained on the training set of labeled URLs (x_L^{train}, y_L^{train}), using the validation set of labeled URLs (x_L^{val}, y_L^{val}) to stop the training. The model is then used to predict the label (\hat{y}_U) for the unlabeled set of URLs (x_U). This set is finally passed to the Active Learning Strategy to select the set of samples to be removed from it and added to the training set. While in a real case scenario there would not be any test set, since in our experiments we have the labels for all URLs, at every iteration we use x_L^{test}, y_L^{test} to measure the quality of the AL strategies - in a sort of ex-post analysis.

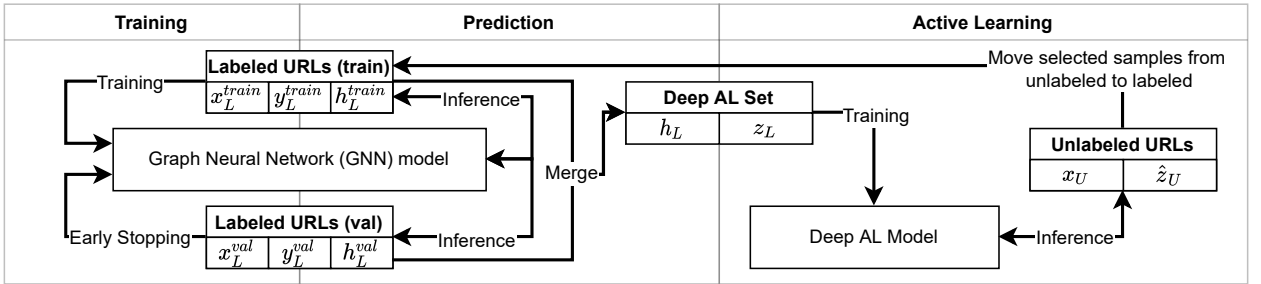


Figure 4.4: Pipeline for Deep Active Learning strategies. First the GNN model is trained on the training set of labeled URLs (x_L^{train}, y_L^{train}), using the validation set of labeled URLs (x_L^{val}, y_L^{val}) to stop the training. The labeled test set (x_L^{test}, y_L^{test}) is then used to evaluate the model's performance (omitted for graphical reasons). From the trained model, embeddings are extracted for the training h_L^{train} and test h_L^{test} set. These embeddings are used, together with a label z_L to train the Deep AL model. For our new AL technique called Deep Error Sampling, $z_L = 1$ for misclassified samples, while $z_L = 0$ for correctly classified samples. At last, this model predicts the labels \hat{z}_U for the unlabeled set of URLs (x_U), which are used to select the set of samples to be removed from it and added to the training set.

human review. Given the budget b , the question is: how can we efficiently and effectively select the B items to be fact-checked by professional journalists? This is precisely the question that active learning (AL) tries to answer in order to maximize the performance of the final model. The first step of any AL procedure is creating and annotating a validation set to guide the subsequent optimization steps. Following the literature [193], a fraction p_{test} of the initial dataset is selected uniformly at random to be used as the test set, and another percentage p_{val} of the remaining data is selected uniformly at random to form the validation set. Of course, the validation set must be human-labeled as well, and the $p_{val}(1 - p_{test})U$ samples will be subtracted from our labeling budget B . The AL strategy we use consists of a series of M iterations. At every iteration, new samples are identified, labeled, and added to the training set. Specifically, at each iteration, first we select k new URLs to annotate and add to the training set. Then, we train the classifier on all the URLs labeled so far. The validation set is used to assess the model performance and perform early stopping if its accuracy exceeds a pre-defined threshold. Iterations are executed until the annotation budget is exhausted. Most AL strategies require a somewhat reliable model to choose which samples to annotate - such a model is used by AL to find instances that bring more discriminative power to the current model. Since at the very beginning of the AL procedure, the training set is empty, and

thus a classification model cannot be reliably obtained, for the first M_{rnd} iterations, we randomly select the k URLs instead of relying on the chosen AL technique.

4.2.4 Active Learning Strategies

In this section, we first present standard and well-established AL strategies that only use the input and output of a classifier to select the next batch of samples to annotate. Then we introduce two deep-learning-based AL methods where the AL strategy itself is a deep neural network. As explained in more detail just below, Deep Unseen Sampling (DUS) is based on a recently proposed active learning procedure for misinformation detection (Ren et al. [238]), while Deep Error Sampling (DES) represents a new active learning strategy that we personally designed to overcome some limitations of current neural approaches to active learning.

Classical Active Learning Strategies

These Active Learning methods use the input and output of the classifier - or even the classifier itself - to decide which URLs to select. The overall structure of these types of methods is shown in figure 4.3.

Random sampling

Random sampling is the most intuitive baseline for the task at hand and represents the de-facto standard in the training of deep learning architectures. At each step, k samples are selected at random from the pool of unlabeled samples. Given that samples are independently picked, this method logically corresponds to selecting and labeling all the B URLs at once.

Uncertainty sampling

In uncertainty sampling, we use the most recently trained model to infer the labels of unlabeled samples. We assume that the last layer of a neural network-based classifier outputs soft-max scores for every class, and we use them to measure how confident the model is about its predictions. According to this principle, we will sample the k for which the model is most uncertain about and we will fact-check them in order to subsequently add them to the next-iteration training dataset. A known disadvantage of this methodology - when applied to deep learning models - is that usually deep learning architectures are overconfident of their predictions [235]. That is, they tend to predict soft-max scores very close to 0% or 100%.

Diversity sampling

Diversity sampling aims at avoiding the selection of very similar samples. The idea is that the model will not receive much help if it is trained with samples that are similar among each other. It is indeed much possible that - for a cluster of very similar samples - the model only needs a few of them to classify the whole cluster correctly. It is then important that the k samples represent different concepts, so that the model can generalise as much as possible. In practice, diversity sampling first clusters samples according to an algorithm like K-Means and then selects only a few examples from each cluster - for instance the centroid, a certain number of outliers and a certain number of random

samples, such that the total is always equal to k . In our work, we used diversity sampling as an additional step for filtering the samples selected with the other AL strategies. After identifying $3k$ samples with one of active learning method, we applied K-Means on the sample features to form k clusters and then we selected the most uncertain URL according to the AL strategy metric. Each sample was represented through its activation scores of the second to last layer of the classification model.

Deep Active Learning Strategies

Deep active learning refers to AL strategies that are specifically designed to work well with deep learning models. In this context, we will use deep active learning to group those pipelines where the AL strategy is itself a deep neural network. The Pipeline for this type of methods is shown in the figure 4.4. In order to train a deep neural network able to identify worth-annotating URLs, we first need to define a suitable training set and a learning objective. Our idea is to use the second-to-last layer activation scores h_L of the fake news classifier for both the training and validation sets as input to this Deep Neural Network. Concerning labels z_L , we experimented with two different DeepAL models. Deep Unseen Sampling (DUS) mimics what was done in the only paper on active learning for misinformation detection (Ren et al. [238]). While the original contribution embeds active learning as an additional feature of a more complex adversarial model for learning node classes on heterogeneous graphs - we decided to test the core idea behind their AL procedure, that is to use internal activation scores of the misinformation classifier to predict whether a sample was already labeled and part of the training set. Deep Error Sampling (DES), instead, is our proposed DeepAL strategy, where we try to predict whether a sample will be correctly classified, thus getting around the problem of soft-max overconfidence. For both methods, the network used is a fully-connected deep neural network. The specific parameters can be found in our anonymized Github repository³. Let's see the two techniques in more details.

Deep Unseen Sampling

Ren et al. [238] start from the assumption that it is good for the classifier to receive new samples other than those it has already seen. They therefore set the labels for the already labeled samples as 0, because the model has already seen them during training, and as 1 for the samples belonging to the validation set, because the model has not in fact seen them during its training. Since the training set of the classifier grows in time, at every iteration the number of samples taken from the validation set is equal to the current size of the training set of the misinformation classifier. Finally, each of the samples used to train the DeepAL architecture are represented through the second-to-last activation scores of the current misinformation detection model, i.e. that trained with the URLs labeled so far. At this point, using these labels as output and the embedding samples as input, we trained a feed-forward neural network to predict whether a URL has been already seen by the fake news classifier or not. In the end, the unlabeled data is given as input to the trained DeepAL architecture and the k samples with the higher prediction, i.e. those which the model predicts are more likely to be unseen by the classifier, are added to the training set.

³<https://anonymous.4open.science/r/Active-Learning-for-Misinformation-Detection-10CC>

Our method: Deep Error Sampling

This is the new method we propose in this paper. Always assuming that we want to train a Deep model that can select the best samples to send to fact-checking, and always constructing the network input from the samples' embeddings, we have chosen the labels differently this time. Our conjecture here is that we might try to predict in advance whether the classifier will mis-classify new samples. We pass the samples that we already have labelled, either training or validation, to the classifier and label 0 those that are classified correctly, and label 1 those that are classified incorrectly. On this set, we train our neural network, and then get the prediction on the unlabelled data. The k samples that the network thinks are most likely to have label 1, will be the ones where our fact-checking classifier is most likely to get it wrong, and it is our belief that they will be most useful for further training.

Mixed Strategies

As in many other areas, often the best result is obtained by aggregating different methodologies. Also here, as the various AL techniques are capable of capturing different information about the samples, it may be useful to combine their outputs. Specifically, we used a simple rank aggregation technique to merge the top- k samples received as output from 2 AL techniques.

4.2.5 Fake News Detection Classifiers

We experimented with three state-of-the-art GNN-based approaches for misinformation detection that work on news diffusion graphs.

- GCN [152] A simple GCN that uses an efficient layer-wise propagation rule based on a first-order approximation of spectral convolutions on graphs. It can learn hidden layer representations that encode both local graph structure and features of nodes.
- GAT [301] The use of multi-head graph attention makes this model computationally highly efficient, thus allowing it to deal with neighbourhoods of various sizes without depending on knowing the entire graph structure upfront.
- GraphSAGE [115] This model exploits inductive node embedding by making use of node features in order to generalise to unseen nodes.

Implementation-wise, we re-implemented in PyTorch Lightning the code, written in PyTorch, distributed by Dou et al. [74]⁴. Concerning the hyper-parameters, we used the values from the original papers as they performed well on both our datasets, as shown in Barnabò et al. [22]. The whole code of our project can be found on a GitHub repository⁵.

4.2.6 Datasets

We tested our pipeline on FbMultiLingMisinfo and Politifact, two publicly-available misinformation detection benchmarks. FbMultiLingMisinfo is a recently published multilingual collection of

⁴<http://github.com/safe-graph/GNN-FakeNews>

⁵<https://github.com/GiorgioBarnabo/Active-Learning-for-Misinformation-Detection>

Dataset	FbMultiLingMisinfo	PolitiFact
Fake News	4,034	157
True News	3,300	157
Total News	7,334	314
Twitter Posts	3,219,383	22,340
Twitter Users	1,240,592	14,873

Table 4.2: Statistics about FbMultiLingMisinfo and PolitiFact

fact-checked news, extracted from the Facebook Privacy-Protected Full URLs Data Set [189], and including diffusion cascades on Twitter for each news article [22].

This dataset includes any URL publicly shared on Facebook at least 100 times between January 2017 and July 2019.

It is particularly relevant because, to the best of our knowledge, 1. it is the only multilingual dataset for misinformation detection; 2. it is the second-largest benchmark dataset for misinformation detection fact-checked at the level of individual news articles (URLs); 3. all included URLs are highly impactful (shared at least 100 times on Facebook); 4. it was shown to be more complex than PolitiFact and GossipCop, the two most used benchmark datasets for misinformation detection [22].

We also experimented with PolitiFact, a widely used benchmark for fake news detection collected from a fact-checking website that focuses on political reporting [264]. Statistics about both datasets are shown in table 4.2.

The difference in the characteristics of the two datasets (one smaller and in only in English, the other multilingual) makes it possible to obtain information on the performance of the AL strategies proposed by us in two different scenarios.

Modeling the Diffusion Cascades of URLs Shared on Twitter

The models we experimented with take as input a graph representing each URL diffusion cascades. As in Dou et al. [74], given the sequence of tweets and retweets mentioning a URL, we built a graph as follows: a central node represents the news and there is an additional node for each tweet. All direct tweets are connected to the central node, while re-tweets are connected to the tweet they are re-tweeting. Finally, similarly to Dou et al. [74], we obtained the node features by encoding the user description with the paraphrase-multilingual-mpnet-base-v2 model from the Hugging Face multilingual sentence embedding model trained as in Reimers and Gurevych [234].

For the central node representing the URL, we used the news title embedding. Our choice of a multilingual model is due to the multilingual nature of the FbMultiLingMisinfo dataset. For the PolitiFact dataset, we used the diffusion graphs shared in [74], but we replaced the given node features with the multilingual sentence embeddings.

4.2.7 Experiments & Results

Experimental setting

We tested all the different AL strategies on GraphSAGE, GAT and GCN - three different state-of-the-art GNN-based misinformation classifiers [22]. We also tested all possible mixed strategies

<i>Results on the FbMultiLingMisinfo Dataset</i>					
AL strategy	Iterations				
metric: AUC	20 (3%)	40 (5,5%)	60 (8%)	80 (11%)	100 (13%)
GAT					
Random	0.71±0.9	0.76±0.10	0.82±0.7	0.84±0.04	0.85±0.05
Uncertainty	0.73±0.06	0.78±0.08	0.82±0.05	0.85±0.06	0.87±0.06
Uncertainty + Diversity	0.74±0.03	0.80±0.04	0.84±0.06	0.85±0.04	0.87±0.03
DUS	0.72±0.11	0.77±0.09	0.81±0.10	0.84±0.08	0.85±0.09
DUS + Diversity	0.71±0.09	0.76±0.09	0.80±0.08	0.84±0.09	0.85±0.07
DES*	0.73±0.05	0.78±0.06	0.82±0.06	0.85±0.03	0.87±0.02
DES* + Diversity	0.73±0.04	0.80±0.05	0.83±0.04	0.85±0.06	0.86±0.04
DES* + Uncertainty	0.74±0.02	0.80±0.02	0.84±0.03	0.86±0.04	0.87±0.02
GraphSAGE					
Random	0.74±0.08	0.82±0.07	0.85±0.10	0.86±0.09	0.87±0.07
Uncertainty	0.75±0.11	0.84±0.07	0.86±0.08	0.88±0.08	0.89±0.09
Uncertainty + Diversity	0.78±0.07	0.83±0.07	0.87±0.07	0.88±0.05	0.89±0.06
DUS	0.75±0.08	0.81±0.09	0.84±0.09	0.86±0.07	0.86±0.06
DUS + Diversity	0.76±0.08	0.81±0.05	0.85±0.05	0.87±0.04	0.87±0.07
DES*	0.77±0.05	0.84±0.07	0.86±0.04	0.88±0.03	0.89±0.04
DES* + Diversity	0.76±0.05	0.84±0.05	0.87±0.04	0.88±0.05	0.89±0.07
DES* + Uncertainty	0.77±0.06	0.84±0.05	0.87±0.03	0.88±0.04	0.89±0.03
GCN					
Random	0.74±0.08	0.79±0.06	0.82±0.09	0.83±0.10	0.85±0.06
Uncertainty	0.76±0.07	0.80±0.10	0.83±0.08	0.84±0.09	0.85±0.07
Uncertainty + Diversity	0.75±0.09	0.81±0.11	0.83±0.09	0.84±0.08	0.86±0.09
DUS	0.77±0.06	0.81±0.05	0.82±0.07	0.84±0.08	0.85±0.07
DUS + Diversity	0.76±0.07	0.80±0.06	0.82±0.05	0.84±0.07	0.85±0.06
DES*	0.77±0.07	0.81±0.06	0.82±0.05	0.85±0.06	0.87±0.04
DES* + Diversity	0.77±0.04	0.81±0.05	0.84±0.03	0.86±0.02	0.87±0.02
DES* + Uncertainty	0.75±0.05	0.80±0.04	0.83±0.04	0.85±0.03	0.87±0.01

Table 4.3: Results on FbMultiLingMisinfo. For each AL strategy, we show the AUC at key iterations. Under the number of iterations - in round brackets - we placed the percentage of the dataset that has been selected and used as training. In addition to that, we must also factor in the 10% validation set that is part of the final fact-checking budget used. With an asterisk we have marked our novel method. DUS = Deep Unseen Sampling, DES = Deep Error Sampling. Results are averaged over 5 runs and reported with their standard deviations.

by combining two sampling strategies as explained in section 4.2.4. The sampling strategies we show in the following results are only those that performed best. The experiment setting was as follows. For both Politifact and FbMultiLingMisinfo we set aside a random 10% of the URLs to use as validation sets. Validation sets are needed to perform early stopping and regularize the training throughout the active learning cycle. Since we assume the validation sets to be labeled as well, they must be subtracted to the total fact-checking budget. For FbMultiLingMisinfo, we set the number of AL iterations to 100, and select 10 URLs per iteration. For Politifact, given its reduced size, we opted for 20 iterations and 5 URLs per iteration. Regardless of the AL method, for FbMultiLingMisinfo the first $M_{rnd} = 5$ iterations always use random sampling, while for Politifact $M_{rnd} = 2$. All experiments were repeated 5 times and results were averaged. In addition, we applied a 3-step moving average on all the sequences of results to make the trends clearer.

<i>Results on the Politifact Dataset</i>					
AL strategy	Iterations				
metric: AUC	8 (12%)	11 (17%)	14 (22%)	17 (27%)	20 (31%)
GAT					
Random	0.83±0.12	0.85±0.07	0.89±0.09	0.88±0.08	0.89±0.07
Uncertainty	0.86±0.09	0.86±0.07	0.88±0.09	0.91±0.07	0.91±0.08
Uncertainty + Diversity	0.87±0.08	0.84±0.10	0.90±0.09	0.91±0.07	0.91±0.08
DUS	0.79±0.11	0.86±0.08	0.88±0.010	0.89±0.09	0.90±0.09
DUS + Diversity	0.76±0.10	0.86±0.09	0.88±0.11	0.91±0.08	0.91±0.07
DES*	0.83±0.06	0.89±0.05	0.90±0.07	0.91±0.03	0.92±0.04
DES* + Diversity	0.86±0.09	0.86±0.06	0.90±0.08	0.91±0.5	0.91±0.07
DES* + Uncertainty	0.84±0.05	0.86±0.04	0.88±0.07	0.90±0.05	0.91±0.03
GraphSAGE					
Random	0.85±0.8	0.85±0.10	0.90±0.09	0.90±0.09	0.90±0.11
Uncertainty	0.84±0.08	0.89±0.11	0.89±0.10	0.90±0.11	0.91±0.09
Uncertainty + Diversity	0.82±0.09	0.88±0.08	0.90±0.07	0.91±0.10	0.92±0.08
DUS	0.80±0.11	0.86±0.09	0.88±0.07	0.90±0.09	0.90±0.08
DUS + Diversity	0.78±0.7	0.87±0.10	0.88±0.09	0.91±0.08	0.91±0.08
DES*	0.88±0.4	0.89±0.05	0.89±0.06	0.91±0.07	0.91±0.06
DES* + Diversity	0.85±0.08	0.89±0.05	0.90±0.04	0.92±0.04	0.91±0.07
DES* + Uncertainty	0.87±0.06	0.89±0.03	0.90±0.05	0.91±0.04	0.92±0.04
GCN					
Random	0.87±0.09	0.90±0.09	0.91±0.10	0.89±0.08	0.89±0.09
Uncertainty	0.90±0.08	0.86±0.10	0.91±0.7	0.93±0.09	0.93±0.08
Uncertainty + Diversity	0.85±0.09	0.87±0.07	0.91±0.9	0.92±0.07	0.92±0.08
DUS	0.87±0.10	0.90±0.07	0.91±0.11	0.93±0.09	0.93±0.08
DUS + Diversity	0.86±0.07	0.90±0.07	0.90±0.08	0.93±0.09	0.93±0.10
DES*	0.87±0.10	0.87±0.09	0.92±0.07	0.93±0.07	0.93±0.07
DES* + Diversity	0.88±0.08	0.87±0.05	0.89±0.06	0.93±0.05	0.94±0.06
DES* + Uncertainty	0.88±0.06	0.89±0.05	0.91±0.05	0.93±0.04	0.92±0.04

Table 4.4: Results on PolitiFact. For each AL strategy, we show the AUC at key iterations. Under the number of iterations - in round brackets - we placed the percentage of the dataset that has been selected and used as training. In addition to that, we must also factor in the 10% validation set that is part of the final fact-checking budget used. With an asterisk we have marked our novel method. DUS = Deep Unseen Sampling, DES = Deep Error Sampling. Results are averaged over 5 runs and reported with their standard deviations.

Key findings

First and foremost, our analysis shows that active learning is a more efficient method for training GNN-based misinformation detection models. Indeed, as shown in tables 4.5 and 4.6 - results from experiments on both FbMultiLingMisinfo and PolitiFact indicate that all tested active learning strategies, except for Deep Unseen Sampling, outperform random sampling, allowing to reach a certain level of classification performance (AUC) with much less labeled data. For FbMultiLingMisinfo specifically, Deep Error Sampling+Uncertainty Sampling yields the best results on GAT and GraphSAGE, while for GCN Deep Error Sampling+Diversity Sampling works better. In all three cases, for lower value of AUC, Uncertainty Sampling and Deep Error Sampling seem to outperform other methods. This is due to the fact that the active learning process is at the very beginning and the

<i>FbMultiLingMisinfo.</i>									
<i>Numbers of iterations required to reach a desired level of AUC</i>									
AL strategy <i>metric: #iterations</i> <i>(lower is better)</i>	Expected average AUC								
	0.73	0.75	0.77	0.79	0.81	0.83	0.85	0.87	0.89
GAT									
Random	33	37	46	51	57	69	100	-	-
Uncertainty	20	27	36	44	54	66	80	100	-
Uncertainty + Diversity	17	25	34	38	44	57	78	96	-
DUS	24	35	40	53	59	76	94	-	-
DUS + Diversity	30	37	44	57	65	77	97	-	-
DES*	19	26	35	43	47	67	79	95	-
DES* + Diversity	18	26	32	37	44	58	79	-	-
DES* + Uncertainty	18	22	30	36	43	54	72	90	-
GraphSAGE									
Random	18	19	33	35	37	51	60	97	-
Uncertainty	17	20	30	33	37	39	53	69	93
Uncertainty + Diversity	9	14	18	35	38	40	52	59	90
DUS	16	20	36	38	40	57	68	-	-
DUS + Diversity	14	18	35	38	39	54	59	80	-
DES*	13	15	20	29	34	37	50	64	92
DES* + Diversity	15	17	23	26	33	37	46	57	90
DES* + Uncertainty	11	15	18	24	32	36	44	54	88
GCN									
Random	18	32	37	40	56	77	96	-	-
Uncertainty	15	18	23	36	49	59	97	-	-
Uncertainty + Diversity	16	20	26	32	38	58	92	-	-
DUS	11	15	20	33	40	83	93	-	-
DUS + Diversity	14	16	24	35	50	71	93	-	-
DES*	13	16	18	34	37	62	80	96	-
DES* + Diversity	13	15	19	30	35	53	72	88	-
DES* + Uncertainty	14	20	27	37	47	59	79	91	-

Table 4.5: Results on FbMultiLingMisinfo. For each AL strategy, we show how many iterations are needed to reach a desired level of expected/average AUC.

Deep Error Sampling architecture needs more data to be trained. Overall, the decrease in number of iterations required to reach a desired level of AUC is significant, with up to 50% less annotated URLs. As for PolitiFact, results reported in table 4.4 suggest similar trends, but it is harder to draw definitive conclusions given the small size of this benchmark dataset and the larger overlap among different active learning procedures. For both benchmark datasets, however, experiments highlight how active learning strategies could make the process of training GNN-based misinformation detection methods not only faster, but also lighter for annotators. Findings on FbMultiLingMisinfo are further confirmed when looking at figures 4.5, 4.6 and 4.7 - which show F1 Macro trends as the number of annotation rounds increases. For instance, the dotted line on figure 4.5 shows that Uncertainty Sampling + Diversity Sampling, Deep Error Sampling + Diversity Sampling, and Deep Error Sampling + Uncertainty Sampling reach an F1 Macro of 0.72 in just over 40 iterations, while the same result takes almost 100 iterations with Random Sampling or Deep Unseen Sampling +

<i>PolitiFact.</i>						
<i>Numbers of iterations required to reach a desired level of AUC</i>						
AL strategy <i>metric: #iterations</i> <i>(lower is better)</i>	Expected average AUC					
	0.83	0.86	0.88	0.90	0.92	0.94
GAT						
Random	8	13	17	-	-	-
Uncertainty	6	8	11	16	-	-
Uncertainty + Diversity	4	7	10	11	-	-
DUS	9	10	11	20	-	-
DUS + Diversity	10	12	14	16	-	-
DES*	8	9	10	11	20	-
DES* + Diversity	6	8	9	11	-	-
DES* + Uncertainty	5	9	11	17	-	-
GraphSAGE						
Random	6	7	10	11	-	-
Uncertainty	7	8	10	17	-	-
Uncertainty + Diversity	8	9	10	13	20	-
DUS	9	10	12	17	-	-
DUS + Diversity	9	11	13	16	-	-
DES*	5	7	8	16	-	-
DES* + Diversity	7	9	10	13	17	-
DES* + Uncertainty	4	7	9	11	20	-
GCN						
Random	5	7	10	13	-	-
Uncertainty	3	7	9	12	15	-
Uncertainty + Diversity	5	8	11	14	17	-
DUS	4	5	9	13	15	-
DUS + Diversity	6	8	10	11	14	-
DES*	5	7	9	13	14	-
DES* + Diversity	4	6	8	10	13	18
DES* + Uncertainty	4	5	8	12	14	-

Table 4.6: Results on PolitiFact. For each AL strategy, we show how many iterations are needed to reach a desired level of expected/average AUC.

Diversity Sampling. On average, when we use GAT to detect fake news in FbMultiLingMisinfo, choosing Uncertainty Sampling + Diversity Sampling, Deep Error Sampling + Diversity Sampling, or Deep Error Sampling + Uncertainty Sampling reduces the number of iterations needed to reach a desired level of F1 Macro by 25 to 40. A similar pattern can be seen in figures 4.7 and 4.6, although the average gap is narrower for GCN.

If we now change the point of observation and look at the performance of different methods given the number of iterations, differences might seem less remarkable. Tables 4.3 and 4.4 still show that all tested AL strategies except Deep Unseen Sampling outperform random sampling - sometimes to a significant extent. On average, however, the AUC is only 2% higher with little difference among Uncertainty Sampling + Diversity Sampling, Deep Error Sampling + Diversity Sampling, and Deep Error Sampling + Uncertainty Sampling. While 2% might seem low, it is worth mentioning that AUC is a demanding metric, and that - in a large news ecosystem like the web or a social network

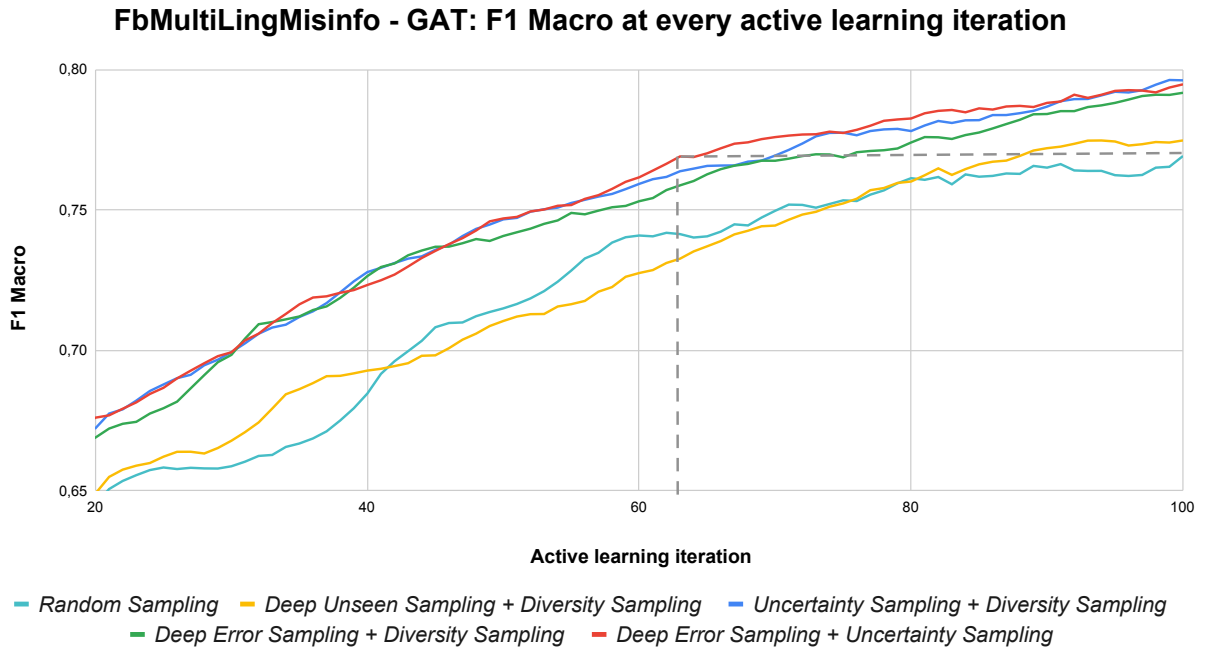


Figure 4.5: F1 Macro at each iteration for 5 AL strategies using GAT on FbMultiLingMisinfo. Deep Unseen + Diversity, Deep Error + Uncertainty and Uncertainty + Diversity all perform similarly and better than both Random and Deep Unseen + Diversity.

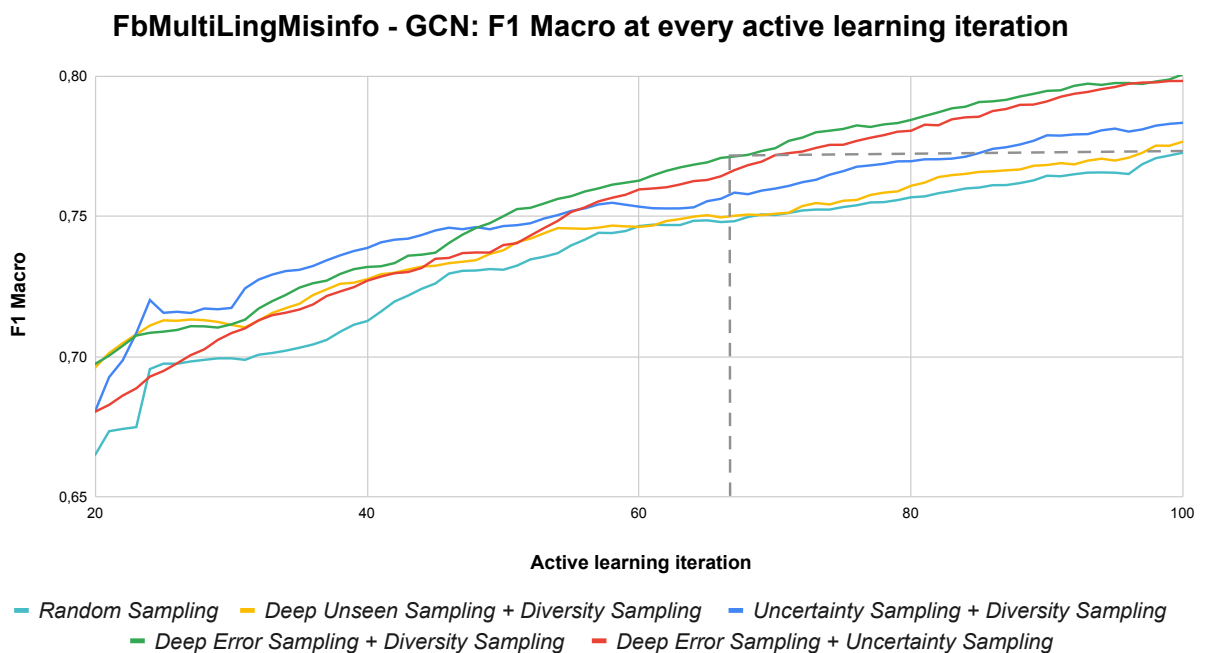


Figure 4.6: F1 Macro at each iteration for 5 AL strategies using GCN on FbMultiLingMisinfo. Deep Error + Diversity and Deep Error + Uncertainty outperform all the other methods.

- even small increases might lead to substantial improvements in the information quality inside the system. Let's now review the results more in depth, and for the two datasets separately.

On FbMultiLingMisinfo, for GAT, Uncertainty + Diversity and Deep Error Sampling + Uncer-

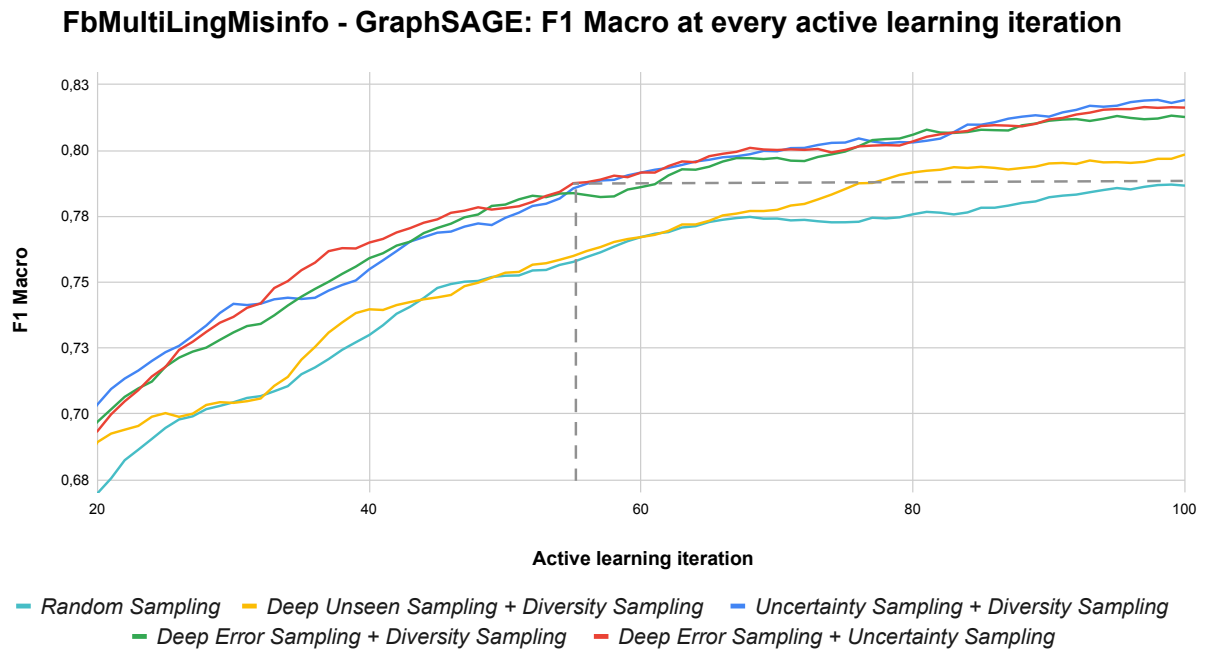


Figure 4.7: F1 Macro at each iteration for 5 AL strategies using GraphSAGE on FbMultiLingMisinfo. Deep Unseen + Diversity, Deep Error + Uncertainty and Uncertainty + Diversity all perform similarly and better than both Random and Deep Unseen + Diversity.

tainty performed equally good or better than any other sampling strategy, while for GraphSAGE also Deep Error Sampling + Diversity reached top performance. In general, regardless of the GNN used, random and Deep Unseen Sampling were always the two methods delivering the worst results - as well exemplified in figures 4.5, 4.6 and 4.7. Finally, when using GCN, Deep Error Sampling + Diversity showed the best performance overall for AUC; its performance in terms of F1 Macro are more clearly highlighted in figure 4.6 - with Deep Error Sampling + Uncertainty and Uncertainty + Diversity as second and third best performing methods respectively. On Politifact, results are more nuanced. Especially when using GCN as the base fake news classifier, no AL method clearly outperforms all the others. When using graphSAGE, Deep Error Sampling and Deep Error Sampling + Uncertainty start emerging as the top performing methods - in the first and second half of the process respectively. Finally it is worth noting that, when using GAT and starting from the 11th iteration, the Deep Error Sampling always reaches the best performance in terms of AUC. The most likely reason why no active learning strategy seems to prevail is that Politifact is too small and homogeneous to really make AL necessary. Overall, while in many cases other active learning strategies perform as well as our proposed Deep Error Sampling, for both FbMultiLingMisinfo and Politifact DES produces more stable outcomes - as measure by the lower variance in the results. In addition, when Deep Error Sampling is coupled with either Diversity or Uncertainty Sampling - result variance between rounds seems to further decrease. Our method thus adds to those already available with its own uniqueness and opens the way for new combinations of more robust active learning strategies.

To conclude, the most remarkable result of our enquiry on active learning for misinformation detection is what we show in figures 4.5, 4.6 and 4.7. The three best AL strategy require only between

45 and 65 iterations to reach the same F1 Macro that random sampling reaches at iteration 100. More generally, given a certain F1 Macro score on FbMultiLingMisinfo, the three plots also indicate the number of iterations needed to reach that level of performance with the three best and the two worst AL strategies for GAT, GCN and GraphSAGE respectively. In the worst cases, random and Deep Unseen Sampling require up to 50% more iterations than Deep Error Sampling + Uncertainty, Deep Error Sampling + Diversity and Uncertainty + Diversity to reach the same F1 Macro score - and the gap seems to increase as the performance of the model increases. These promising results pave the way for a great reduction in time and money spent for annotating online news - thus making the training of GNN-based fake news detectors more affordable.

4.2.8 Conclusion & Future Work

In this work we presented an in-depth analysis of active learning strategies in the context of automatic misinformation detection, we proposed a new deep active learning architecture that, when coupled with uncertainty sampling, performs equally or better than the most common AL strategies and the only proposed active learning procedure specifically targeting fake news detection. A key finding is that, in the context GNN-based models for misinformation detection, compared to random sampling AL allows – on average – to achieve a 2% increase in AUC for the same percentage of third-party fact-checked news and to save up to 25% of labeling effort for a desired level of classification performance. While this direction seems promising, more ablation studies are needed to find the optimal number of URLs that should be labeled at every AL iteration. Experiments on much larger datasets would also help gauging the feasibility of our proposed method in a real world scenario. More in general, while hard to do, it would also make sense to jointly optimize the hyper-parameters of both the misinformation classifier and of the Deep AL architecture. Finally, the Deep AL model itself could be made much more complex, possibly leading to much greater improvements.

4.2.9 Ethical Considerations

We acknowledge that automatic misinformation detection poses well-documented risks, including the marginalization of minority discourse through disparate false positive rates. At the same time, it also contributes to fighting misinformation campaigns that usually target marginalized groups, such as immigrants. The ethical considerations in this case affect all automated misinformation finding tools, and are not specific to our work, which uses well-established practices. The main subject of the work is in fact our Active Learning algorithm, the main purpose of which is to improve the performance of Misinformation Detection models. It is the use of the latter that can lead to ethical concerns and not our algorithm.

4.3 RRAML: Reinforced Retrieval Augmented Machine Learning

The emergence of large language models (LLMs) has revolutionized machine learning and related fields, showcasing remarkable abilities in comprehending, generating, and manipulating human language. However, their conventional usage through API-based text prompt submissions imposes certain limitations in terms of context constraints and external source availability. LLMs suffer from the problem of hallucinating text, and in the last year, several approaches have been devised to overcome this issue: adding an external Knowledge Base or an external memory consisting of embeddings stored and retrieved by vector databases. In all the current approaches, though, the main issues are: (i) they need to access an embedding model and then adapt it to the task they have to solve; (ii) in case they have to optimize the embedding model, they need to have access to the parameters of the LLM, which in many cases are “black boxes”. To address these challenges, we propose a novel framework called Reinforced Retrieval Augmented Machine Learning (RRAML). RRAML integrates the reasoning capabilities of LLMs with supporting information retrieved by a purpose-built retriever from a vast user-provided database. By leveraging recent advancements in reinforcement learning, our method effectively addresses several critical challenges. Firstly, it circumvents the need for accessing LLM gradients. Secondly, our method alleviates the burden of retraining LLMs for specific tasks, as it is often impractical or impossible due to restricted access to the model and the computational intensity involved. Additionally, we seamlessly link the retriever’s task with the reasoner, mitigating hallucinations and reducing irrelevant and potentially damaging retrieved documents. We believe that the research agenda outlined in this paper has the potential to profoundly impact the field of AI, democratizing access to and utilization of LLMs for a wide range of entities.

4.3.1 Introduction

The advent of Large Language Models (LLMs) has brought about a paradigm shift in machine learning and its related disciplines. LLMs [15, 37, 210, 248, 290] have exhibited unprecedented capabilities in understanding, generating, and manipulating the human language. Famously, ChatGPT [210] has entered the public space by reaching one million users in a matter of days. The way these models are used is through API that only allows submitting a textual prompt and getting back from the server the generated text. However, this causes an immediate limitation: all information must be passed through this context, and we know transformer-based models do not scale nicely. Even if they did, API costs are charged on the basis of their usage. Therefore, using long contexts would be expensive. Even if one had the resources to run their own LLM, the costs of training and of the hardware infrastructure, and the environmental impact should be considered. There is an impendent need, though, to accommodate the enormous power of those models to specific user needs by making sure that they could use the reasoning capabilities of LLMs, through in-context

learning [37] on their data.

A solution is to adopt a retrieval-augmented approach [163, 326]. In this setting, a retriever is used to filter out relevant information to be passed as context to the reasoner. This generates a new problem, however, namely that the retriever and the reasoner are not aligned [288, 289, 294]. In particular, the retriever might not be trained on the task of interest to the user. Moreover, the retriever might actually provide “dangerous” pieces of information to the reasoner, as proved in [247], leading to poor results and, more importantly, to hallucinations.

Ideally, one would have to fine-tune these models to account for these issues. Within this setting, fine-tuning the model for a given task is technically impossible. We asked ourselves: “*Is it still possible to use the API that gatekeeps those powerful LLMs on our data without the need for fine-tuning?*” We show that this question has a positive answer and in this paper, we propose a novel framework, Reinforced Retrieval Augmented Machine Learning (RRAML), in which we combine the reasoning capabilities of large foundational models enhanced by the provision of supporting relevant information provided by a retriever that searches them in a large database. In this setting, an efficient retriever model is tasked to search for relevant information in an arbitrarily large database of data provided by users. Once this set of relevant data has been retrieved, it is forwarded to the reasoner (a large foundational model such as ChatGPT, for instance) through its API to “reason” on the input and produce an adequate result. In particular, we plan to overcome current limitations, namely that the retriever’s task is detached from that of the reasoner, reducing in such a way the tendency of LLM to hallucinate and diminishing the number of damaging documents (as defined in [41, 246, 294]) returned by the retriever. The approach we devise in this research work exploits recent advances in reinforcement learning. Recently, in fact, reinforcement learning techniques like PPO [254] have been used to improve large foundational models with human feedback where the loss is non-differentiable. We propose to link the training phase of the retriever to the final task outcome by the use of a purposefully crafted reward model that depends either on human feedback or on the specific characteristics of the task data. The RL technique also offers the advantage of not requiring fine-tuning an LLM as a reasoner, which can be considered a black box in this setting, and exchanged freely.

Finally, we argue that the research agenda we lay out in this paper has the potential to hugely impact the field of AI and democratize the access and use of these large foundational models to a large set of entities.

4.3.2 Methodology

The system takes as input a task description, a query, and a database and gives as output the response generated by a reasoner. The overall system architecture, shown in Figure 4.8, consists of three main components: a Generative Language Model, a Retriever, and a Reasoner (typically an LLM).

More in detail, the Generative Language Model takes the *task description* and *query* as input and generates a prompt. The Retriever takes the *query* and the *database* as input and outputs a support set, which is then concatenated with the *query* and passed to the Reasoner.

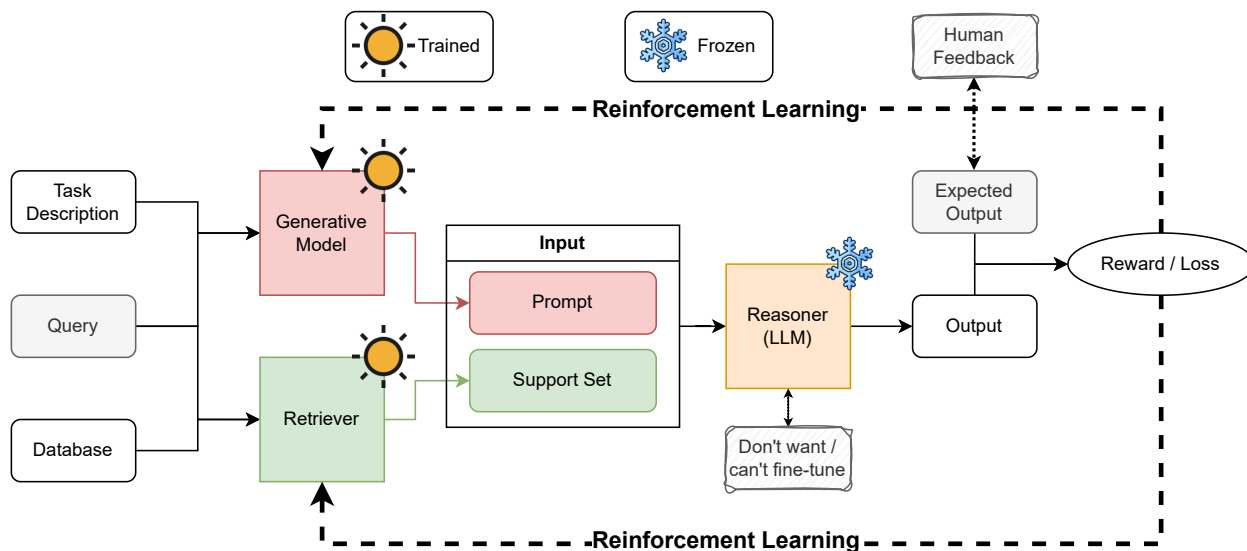


Figure 4.8: High-level design of the RRAML framework. On the left side, there are the three inputs: Task Description, user’s query, and a database that represents the external knowledge used to augment/update the reasoner. Then, we present the overall architecture flow with the Retriever, Generative Language Model, and Reasoner. Finally, how the reward is computed and propagated in the Generative Language Model and Retriever.

Data

The data is a critical component of the framework: the *task description* guides the generation of an appropriate prompt, the *query* represents the user request, and the *database* provides the data needed by the reasoner to perform the task.

Task Description The *task description* is a string that defines the nature of the task, possibly with expected results, that the user wants to perform. For example, if the user wants to generate a summarization of multiple news articles, a possible *task description* could be “News Summarization”. If the user wants to perform question answering on a vast document collection, the *task description* could be “Question Answering”.

Query The *query* represents the user’s need. The Retriever will operate on the *database* w.r.t to the user’s query, and the resulting data is input for the task. For example, if the user wants to summarize a collection of news articles, the *query* could be the topic the user is interested in. If the user wants to answer a specific *query*, this becomes the actual question.

Database The *database* is a collection of public or private data (or documents) that can be queried to provide relevant information to satisfy the user’s information needs. The database represents the knowledge needed by the Reasoner to perform the task. The data stored in the *database* will depend on the specific task and may include text, images, audio, and other data types (as in [294]). For example, if the user wants to summarize multiple news articles, the *database* could be an indexed collection of articles. If the user wants to perform Question Answering, the *database* may consist of facts related to a particular topic (as in [288, 289]).

Models

Generative Language Model The Generative Language Model component of the framework is responsible for generating textual instructions based on the input *Task Description* and *Query* that maximize the rewards w.r.t Reasoner. Specifically, it receives a string representing the task to be performed (*Task Description*) and a query (*Query*) that represents the user’s request. The Generative Language Model then generates a textual prompt that is relevant to the query and the task by performing automatic prompt engineering.

Retriever The Retriever component of the framework is responsible for retrieving relevant data from the Database based on the user’s query. We refer to the Retriever outputs as support set (as in [288, 289]). A support set is a subset of the data from the Database that either directly answers the given query or contributes to the final answer.

Prompt Aggregator This component is responsible for processing the input required by the *Reasoner*. In its simplest form, it just needs to concatenate the prompt generated by the Generative Language Model with the Support Set provided by the Retriever. However, in a more complex version, it may need to rework the prompt based on the number of support sets received to ensure that the LLM can provide a coherent response. For example, if the Retriever provides two support sets, the Prompt Aggregator may need to split the prompt into two parts and concatenate each part with one of the support sets.

Reasoner The Reasoner is responsible for generating the answer to the user’s query based on the final prompt generated by the Prompt Aggregator. The Reasoner can be a pre-trained model like GPT or a custom-trained model specific to the task at hand. The output of the LLM is a textual response, which can be further parsed to comply with the intended output.

Reinforcement Learning

The Reinforcement Learning (RL) part of the framework is responsible for fine-tuning the Generative Language Model (GLM) and Retriever based on the computed reward. The RL is a crucial part of RRAML, it will be used to constantly improve the GLM and Retriever. As mentioned earlier, the retriever will get a penalty if some of his recommendations will leads the Reasoner to a hallucinate, for example by adding damaging documents. The RL allows use to integrate and augment the signals in the training of these models, going beyond the data present in their training set, ensuring that they are aligned with the environment (i.e., the reasoner and the final task).

Reward The reward function can be defined based on the similarity between the generated output and the expected output and it can be estimated by training a Reward Model [254].

RL algorithm The specific RL method which can be used is Deep Q-Networks (DQN) [191], which is a model-free RL algorithm that learns to maximize the cumulative reward over time. DQN combines Q-Learning, which is a RL algorithm that learns the optimal action-value function, with a Deep Neural Network to approximate the action-value function. In the proposed framework, DQN is used to train the Generative Language Model and the Retriever to maximize the reward obtained

from user feedback. The update process is performed by backpropagating the reward signal through the neural networks using Stochastic Gradient Descent (SGD). The weights of the neural networks are updated in the direction that maximizes the expected reward, using the Q-Learning update rule. The update is performed iteratively until convergence, which is achieved when the expected reward stops improving.

Human-in-the-loop Human preferences can be incorporated into our ML system by allowing users to provide feedback on the system’s output. This feedback will be used to compute the reward for the RL algorithm and will help improve the performance of the overall system over time. We acknowledge that some tasks may not have a clear expected output or may require additional context that is not available in the input data. In these cases, we will leverage human-in-the-loop approaches to provide additional context and guidance to the system. For example, crowd-sourcing platforms or internal subject matter experts can be used to provide feedback on the system’s output and help train the model on more complex tasks.

4.3.3 Use Case Example

RRAML promises to be effective in many applications. Consider a situation where a company possesses a private database, which consists of factual information expressed in natural language, and they need to apply reasoning to this data. The volume of their data may exceed the context capacity of the LLM, and fine-tuning is not an option, for pricing/environmental impact or because the LLM is served by other company APIs. To tackle this challenge, RRAML uses its retriever to get only the relevant facts within the context, enabling the LLM to reason over them.

For instance, suppose a company has an employee list, projects that employees are currently or were previously assigned to, and performance evaluation grids with text-based feedback from superiors. The company might want to assign employees to a new project on a specific topic. To do so, it is necessary to input the information contained in these data to the LLM. However, due to capacity constraints, the entire data cannot fit within the context. Therefore, the retriever has to return a subset of this information, perhaps excluding data on projects from the distant past, employees who are already overburdened with multiple projects, or employees who have never worked on a project related to the same topic.

4.3.4 Related Work

Recent years have seen the emergence of large language models. Starting from the first Generative Pre/Training Model, better known as GPT [226], these kinds of large language models have rapidly improved. Even further, deep learning models have now reached multimodal capabilities beyond just images, with methods proficient on audio [24, 34, 70], video [167, 180], and 3D [52, 114, 293]. GPT-4 [211] is the most recent iteration, but in the meanwhile, many have rushed to propose their own version. Google has recently released BARD⁶, while Meta has proposed their own take on LLM with LLaMA [290]. The research community has also capitalized its effort by releasing several open

⁶<https://bard.google.com/>

source LLM of different sizes, like Bloom [248], Dolly⁷, and RWKV [217]. However, all these models fail to scale to a larger context size, either by excessive computational costs or by “losing it in the middle”, as shown in [172].

To address this context-length limitation, some have tried to incorporate external knowledge into LLMs [73, 96, 218]. In particular, in “Retrieval-enhanced machine learning” [339], authors have envisioned a framework in which retrieval systems can enhance the performance of a machine learning model. More recently, there have been attempts of jointly training retrieval models with LLMs [163, 347], notably, the line of research on neural databases, in which the authors tried to replace a traditional database with a neural framework removing the need for a schema [288, 289, 294]. However, all these works assume full access to the reasoner module, which is not the case for most users in practice.

To overcome this limitation, many have tried to craft systems that are able to deliver an optimized prompt that is input to the LLM. For instance, the research conducted by [176] demonstrated a substantial influence of the sequence in which prompts are presented on the ultimate performance of the task. Meanwhile, a study by Nie et al. [201] highlighted that the performance is susceptible to the arrangement of the examples in the prompt, prompt templates, and the in-context instances in the prompt. Lester et al. [162] suggested a method to enhance task performance by adding adjustable tokens during fine-tuning. LLM-AUGMENTER iteratively revises [218] to improve the model response.

All the works introduced above do not improve on the retriever, which is assumed fixed. In our work, we propose to finetune the retriever in conjunction with the reasoner to improve on results. Since the feedback is non-differentiable we resort to reinforcement learning. In particular, recent formulation such as Proximal Policy Optimization (PPO) [79] make use of a differentiable neural reward module to include and account for generally non-differentiable feedback, like in the case of reinforcement learning with human feedback (RLHF).

4.3.5 Conclusions

In conclusion, RRAML provides a promising framework for building intelligent interfaces to interact with large language models like GPT. By combining a generative language model with a retriever, this approach can effectively improve the performance of language models and help them understand user intents better.

However, this approach also comes with several challenges and uncertainties, such as the need for a large amount of training data, the potential for bias in the data and models, and the difficulty of balancing the trade-offs between generative and retrieval-based approaches.

Despite these challenges, RRAML holds great promise for creating more intelligent, natural, and effective interfaces for interacting with language models. We hope that this paper has provided a useful overview of this approach and its potential applications, and we look forward to further research and development in this exciting area.

⁷<https://github.com/databricks/dolly>

Chapter 5

Conclusions

Summary of Key Findings and Contributions

Throughout this research, the pursuit of trustworthy AI has led to significant findings and contributions. We have successfully addressed each of our research objectives:

Objective 1: Develop Explainable AI Methods and Components

We have conceptualized and developed architectural components that make AI systems more explainable by design. The introduction of innovative techniques and models, such as NEWRON and NEWRON+LEN, has enabled AI decisions to become interpretable and transparent. Furthermore, our exploration into concept-based explainability has opened new avenues for understanding AI operations, enhancing transparency, and expanding the range of interpretable concepts within AI systems.

Objective 2: Establish Trustworthiness Through Robust Loss Functions

Our research has resulted in the design and evaluation of robust loss functions. These functions serve as the bedrock for enhancing the trustworthiness of AI systems. By effectively mitigating vulnerabilities stemming from noisy labels and missing data, we bolster the reliability and fairness of AI decision-making.

Objective 3: Architect Trustworthy AI Auxiliary Frameworks

We have successfully designed auxiliary architectural frameworks that facilitate a spectrum of critical functions. They enable algorithmic recourse, empower the detection of misinformation, and enrich information retrieval through the integration of a neural database. This collective enhancement serves as a cornerstone, elevating the overall trustworthiness of AI systems, ensuring confidence in their deployment across diverse applications and domains.

Objective 4: Contribute to Trustworthy AI Research

Our research extends beyond the immediate objectives, offering valuable insights and solutions to the broader field of trustworthy AI. By conducting comprehensive empirical studies, evaluations, and experiments, we have not only enriched the understanding of trustworthy AI but have also provided

a valuable repository of practical guidelines and best practices. These contributions ensure that the principles of trustworthiness are seamlessly integrated into future AI technologies, fostering responsible AI development.

Future Research Directions in Trustworthy AI

Looking ahead, there are promising avenues for future research in the realm of trustworthy AI. The following are key areas where further exploration is warranted:

Advancing Interpretability and Transparency

One of the paramount objectives in Trustworthy AI is to continually enhance the methods and techniques for making AI systems even more interpretable and transparent to users. This includes the development of novel explainability tools and strategies that not only shed light on model decisions but also ensure the comprehensibility of the explanations to a broader audience. Future research should strive to bridge the gap between complex AI models and end-users, making AI's decision-making processes more accessible and comprehensible.

Resilience to Adversarial Threats and Biases

The landscape of AI is filled with adversarial attacks that challenge the trustworthiness of AI systems. Future research must focus on developing AI systems that can proactively defend against emerging adversarial threats. This encompasses the creation and implementation of mechanisms able to detect and mitigate adversarial attacks, ensuring that decision-making remains unaltered.

Deepening Fairness in AI

Fairness in AI is a subject of profound importance. Research in this area should be extended to ensure equitable decision-making and unbiased outcomes across a wide spectrum of applications. This includes developing fairness-aware machine learning techniques, investigating the root causes of bias, and formulating methods to rectify these biases. The future of AI must be one where AI systems are unwaveringly fair, impartial, and free from discriminatory behavior.

Trustworthiness Across Diverse Domains

The principles of trustworthiness need to be extended beyond the domains explored in this research. Future research should address the unique challenges and requirements of diverse domains such as healthcare, finance, and education. Each domain brings its own set of ethical, legal, and practical considerations, and research should account for these specifics to ensure that Trustworthy AI can be universally applied.

Closing Remarks

As we conclude our exploration of the architectural components of Trustworthy AI, it is essential to reflect on its significance and its broader implications for the field of artificial intelligence. Throughout this thesis, we have introduced innovative architectural elements and methodologies, providing

valuable insights into the development of AI systems prioritizing trustworthiness, explainability, and accountability. Collectively, our research has significantly advanced the understanding and practical implementation of these architectural components, thereby fortifying the trustworthiness of AI systems across a spectrum of domains. These contributions are valuable in promoting ethical AI practices, improving user trust, and addressing the multidimensional challenges posed by AI technologies.

However, it's crucial to acknowledge that these contributions represent just a step towards Trustworthy AI. The rapid pace of AI technological evolution, the continuously evolving ethical landscape, and the dynamic nature of societal values necessitate an ongoing commitment to research, development, and ethical reflection, ensuring that trustworthiness and societal values remain at the forefront of AI innovation.

In conclusion, we underscore the imperative of interdisciplinary collaboration, entailing the active involvement of ethicists, policymakers, social scientists, and affected communities in shaping the future of AI. As AI researchers and practitioners, it is our responsibility to prioritize ethical AI practices, transparency, fairness, and accountability in all aspects of AI development.

Bibliography

- [1] E. Abbe. Beiträge zur theorie des mikroskops und der mikroskopischen wahrnehmung. *Archiv für mikroskopische Anatomie*, 9(1):413–468, 1873.
- [2] A. Abutbul, G. Elidan, L. Katzir, and R. El-Yaniv. Dnf-net: A neural architecture for tabular data. *arXiv preprint arXiv:2006.06465*, 2020.
- [3] J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, and B. Kim. Sanity checks for saliency maps. *Advances in neural information processing systems*, 31, 2018.
- [4] G. Adomavicius and A. Tuzhilin. Context-aware recommender systems, 2010.
- [5] D. Alvarez Melis and T. Jaakkola. Towards robust interpretability with self-explaining neural networks. *Advances in neural information processing systems*, 31, 2018.
- [6] R. Andrews, J. Diederich, and A. B. Tickle. Survey and critique of techniques for extracting rules from trained artificial neural networks. *Knowledge-based systems*, 8(6):373–389, 1995.
- [7] A. Apicella, F. Donnarumma, F. Isgrò, and R. Prevete. A survey on modern trainable activation functions. *Neural Networks*, 2021.
- [8] S. Ö. Arik and T. Pfister. Tabnet: Attentive interpretable tabular learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 6679–6687, 2021.
- [9] A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins, et al. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, 58:82–115, 2020.
- [10] A. Artelt and B. Hammer. On the computation of counterfactual explanations - A survey. *CoRR*, abs/1911.07749, 2019. URL <http://arxiv.org/abs/1911.07749>.
- [11] M. G. Augasta and T. Kathirvalavakumar. Reverse engineering the neural networks for rule extraction in classification problems. *Neural processing letters*, 35(2):131–150, 2012.
- [12] M. G. Augasta and T. Kathirvalavakumar. Rule extraction from neural networks—a comparative study. In *International Conference on Pattern Recognition, Informatics and Medical Engineering (PRIME-2012)*, pages 404–408. IEEE, 2012.
- [13] S. Azzolin, A. Longa, P. Barbiero, P. Lio, and A. Passerini. Global explainability of gnns via logic combination of learned concepts. *The First Learning on Graphs Conference*, 2022.

-
- [14] A. Bacciu, F. Cuconasu, F. Siciliano, F. Silvestri, N. Tonello, and G. Trappolini. Rraml: Reinforced retrieval augmented machine learning. In *AIxIA 2023–Advances in Artificial Intelligence: XXIIInd International Conference of the Italian Association for Artificial Intelligence, AIxIA 2023, Rome, Italy, November 6 – 9, 2023, Discussion Track*, 2023.
- [15] A. Bacciu, G. Trappolini, A. Santilli, E. Rodolà, and F. Silvestri. Fauno: The italian large language model that will leave you senza parole! *arXiv preprint arXiv:2306.14457*, 2023.
- [16] M. Bajaj, L. Chu, Z. Y. Xue, J. Pei, L. Wang, P. C.-H. Lam, and Y. Zhang. Robust counterfactual explanations on graph neural networks. *Advances in Neural Information Processing Systems*, 34, 2021.
- [17] R. Baly, G. Da San Martino, J. Glass, and P. Nakov. We can detect your bias: Predicting the political ideology of news articles. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4982–4991, Online, Nov. 2020.
- [18] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *science*, 286(5439): 509–512, 1999.
- [19] R. Baraglia, F. Casheda, V. Carneiro, D. Fernandez, V. Formoso, R. Perego, and F. Silvestri. Search shortcuts: a new approach to the recommendation of queries, 2009.
- [20] P. Barbiero, G. Ciravegna, F. Giannini, P. Lió, M. Gori, and S. Melacci. Entropy-based logic explanations of neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 6046–6054, 2022.
- [21] P. Barbiero, G. Ciravegna, F. Giannini, P. Lió, M. Gori, and S. Melacci. Entropy-based logic explanations of neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 6046–6054, 2022.
- [22] G. Barnabò, F. Siciliano, C. Castillo, S. Leonardi, P. Nakov, G. Da San Martino, and F. Silvestri. Fbmultilingmisinfo: Challenging large-scale multilingual benchmark for misinformation detection. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2022.
- [23] G. Barnabò, F. Siciliano, C. Castillo, S. Leonardi, P. Nakov, G. Da San Martino, and F. Silvestri. Deep active learning for misinformation detection using geometric deep learning. *Online Social Networks and Media*, 33:100244, 2023.
- [24] G. Barnabò, G. Trappolini, L. Lastilla, C. Campagnano, A. Fan, F. Petroni, and F. Silvestri. Cycledrums: automatic drum arrangement for bass lines using cyclegan. *Discover Artificial Intelligence*, 3(1):4, 2023.
- [25] P. Battaglia, R. Pascanu, M. Lai, D. Jimenez Rezende, et al. Interaction networks for learning about objects, relations and physics. *Advances in neural information processing systems*, 29, 2016.
- [26] B. Becker and R. Kohavi. Adult. UCI Machine Learning Repository, 1996. DOI: <https://doi.org/10.24432/C5XW20>.

-
- [27] F. Betello, F. Siciliano, P. Mishra, and F. Silvestri. Investigating the robustness of sequential recommender systems against training data perturbations: an empirical study. *46th European Conference on Information Retrieval (ECIR) 2024*, 2024.
- [28] U. Bhatt, P. Ravikumar, et al. Building human-machine trust via interpretability. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 9919–9920, 2019.
- [29] S. D. Bhattacharjee, A. Talukder, and B. V. Balantrapu. Active learning based news veracity detection with feature weighting and deep-shallow fusion. In *2017 IEEE International Conference on Big Data (Big Data)*, pages 556–565. IEEE, 2017.
- [30] T. Bian, X. Xiao, T. Xu, P. Zhao, W. Huang, Y. Rong, and J. Huang. Rumor detection on social media with bi-directional graph convolutional networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 549–556, 2020.
- [31] G. Bologna and Y. Hayashi. A comparison study on rule extraction from neural network ensembles, boosted shallow trees, and svms. *Applied Computational Intelligence and Soft Computing*, 2018, 2018.
- [32] C. Bonferroni. Teoria statistica delle classi e calcolo delle probabilita. *Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commerciali di Firenze*, 8:3–62, 1936.
- [33] V. Borisov, T. Leemann, K. Sessler, J. Haug, M. Pawelczyk, and G. Kasneci. Deep neural networks and tabular data: A survey. *ArXiv*, abs/2110.01889, 2021.
- [34] Z. Borsos, R. Marinier, D. Vincent, E. Kharitonov, O. Pietquin, M. Sharifi, D. Roblek, O. Teboul, D. Grangier, M. Tagliasacchi, et al. Audioldm: a language modeling approach to audio generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.
- [35] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth, 1984.
- [36] I. Brillhante, J. A. Macedo, F. M. Nardini, R. Perego, and C. Renso. Tripbuilder: A tool for recommending sightseeing tours. In *Advances in Information Retrieval: 36th European Conference on IR Research, ECIR 2014, Amsterdam, The Netherlands, April 13-16, 2014. Proceedings 36*, pages 771–774. Springer, 2014.
- [37] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.
- [38] M. S. Bucarelli, L. Cassano, F. Siciliano, A. Mantrach, and F. Silvestri. Leveraging inter-rater agreement for classification in the presence of noisy labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3439–3448, 2023.
- [39] R. Burke, M. P. O’Mahony, and N. J. Hurley. Robust collaborative recommendation. In *Recommender systems handbook*, pages 961–995. Springer, 2015.

-
- [40] P. Cannarsa and T. D’Aprile. *Introduction to Measure Theory and Functional Analysis*. UNITEXT. Springer International Publishing, 2015. ISBN 9783319170183. URL <https://books.google.it/books?id=C1UergEACAAJ>.
- [41] D. Carmel, N. Cohen, A. Ingber, and E. Kravi. Ir evaluation and learning in the presence of forbidden documents. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 556–566, 2022.
- [42] D. Castelvechi. Can we open the black box of ai? *Nature News*, 538(7623):20, 2016.
- [43] M. Chakraborty, S. K. Biswas, and B. Purkayastha. Rule extraction from neural network trained using deep belief network and back propagation. *Knowledge and Information Systems*, 62(9):3753–3781, 2020.
- [44] S. Chandra, P. Mishra, H. Yannakoudakis, M. Nimishakavi, M. Saeidi, and E. Shutova. Graph-based modeling of online communities for fake news detection. *arXiv preprint arXiv:2008.06274*, 2020.
- [45] J. Chang, C. Gao, Y. Zheng, Y. Hui, Y. Niu, Y. Song, D. Jin, and Y. Li. Sequential recommendation with graph neural networks, 2021.
- [46] H. Chen, C. Gomez, C.-M. Huang, and M. Unberath. Explainable medical imaging ai needs human-centered design: guidelines and evidence from a systematic review. *npj Digital Medicine*, 5(1):156, 2022.
- [47] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.
- [48] T. Chen and C. Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD ’16*, pages 785–794, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4232-2. doi: 10.1145/2939672.2939785. URL <http://doi.acm.org/10.1145/2939672.2939785>.
- [49] Z. Chen, Y. Bei, and C. Rudin. Concept whitening for interpretable image recognition. *Nature Machine Intelligence*, 2(12):772–782, 2020.
- [50] Z. Chen, F. Silvestri, J. Wang, Y. Zhang, Z. Huang, H. Ahn, and G. Tolomei. Grease: Generate factual and counterfactual explanations for gnn-based recommendations. *arXiv preprint arXiv:2208.04222*, 2022.
- [51] Z. Chen, F. Silvestri, J. Wang, Y. Zhang, and G. Tolomei. The dark side of explanations: Poisoning recommender systems with counterfactual examples. *arXiv preprint arXiv:2305.00574*, 2023.
- [52] Z. Chen, G. Wang, and Z. Liu. Scenedreamer: Unbounded 3d scene generation from 2d image collections, 2023.
- [53] J. Y. Chin, Y. Chen, and G. Cong. The datasets dilemma: How much do we really know about recommendation datasets? In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pages 141–149, 2022.

- [54] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.
- [55] G. Ciravegna, P. Barbiero, F. Giannini, M. Gori, P. Lió, M. Maggini, and S. Melacci. Logic explained networks. *Artificial Intelligence*, 314:103822, 2023.
- [56] J. Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960. doi: 10.1177/001316446002000104. URL <https://doi.org/10.1177/001316446002000104>.
- [57] K. M. Collins, U. Bhatt, and A. Weller. Eliciting and learning with soft labels from every annotator, 2022. URL <https://arxiv.org/abs/2207.00810>.
- [58] W. C. Credit Fusion. Give me some credit, 2011. URL <https://kaggle.com/competitions/GiveMeSomeCredit>.
- [59] L. Cui, X. Tang, S. Katariya, N. Rao, P. Agrawal, K. Subbian, and D. Lee. Allie: Active learning on large-scale imbalanced graphs. In *Proceedings of the ACM Web Conference 2022*, pages 690–698, 2022.
- [60] L. Cuneo, M. Castello, S. Piazza, I. Nepita, I. Cainero, G. Tortarolo, L. Lanzañò, P. Bianchini, G. Vicidomini, and A. Diaspro. A deep learning-based method to spectrally separate overlapping fluorophores based on their fluorescence lifetime. *Nuovo Cimento C - Colloquia and Communications in Physics*, 46, 2023.
- [61] L. Cuneo, F. Siciliano, M. Castello, S. Piazza, F. Silvestri, and A. Diaspro. Explainable-by-design machine learning model for overlapping fluorophores separation based on fluorescence lifetime. In *Computational Intelligence Methods for Bioinformatics and Biostatistics: 18th International Meeting, CIBB 2023, Padova, Italy, September 6–8, 2023*, 2023.
- [62] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(2):303–314, 12 1989. doi: <https://doi.org/10.1007/BF02551274>.
- [63] F. Dalvi, N. Durrani, H. Sajjad, Y. Belinkov, A. Bau, and J. Glass. What is one grain of sand in the desert? analyzing individual neurons in deep nlp models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6309–6317, 2019.
- [64] F. Dalvi, A. Nortonsmith, A. Bau, Y. Belinkov, H. Sajjad, N. Durrani, and J. Glass. Neurox: A toolkit for analyzing individual neurons in neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9851–9852, 2019.
- [65] A. Darbari. Rule extraction from trained ann: A survey. 2000.
- [66] A. Davies, P. Veličković, L. Buesing, S. Blackwell, D. Zheng, N. Tomašev, R. Tanburn, P. Battaglia, C. Blundell, A. Juhász, et al. Advancing mathematics by guiding human intuition with ai. *Nature*, 600(7887):70–74, 2021.
- [67] A. P. Dawid and A. M. Skene. Maximum likelihood estimation of observer error-rates using the em algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):20–28, 1979. ISSN 00359254, 14679876. URL <http://www.jstor.org/stable/2346806>.

-
- [68] G. De Toni, P. Viappiani, B. Lepri, and A. Passerini. Generating personalized counterfactual interventions for algorithmic recourse by eliciting user preferences. *arXiv preprint arXiv:2205.13743*, 2022.
- [69] Y. Deldjoo, T. Di Noia, E. Di Sciascio, and F. A. Merra. How dataset characteristics affect the robustness of collaborative recommendation models. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, pages 951–960, 2020.
- [70] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever. Jukebox: A generative model for music. *arXiv preprint arXiv:2005.00341*, 2020.
- [71] A. Diaspro and P. Bianchini. Optical nanoscopy. *La Rivista del Nuovo Cimento*, 43(8):385–455, 2020.
- [72] M. A. Digman, V. R. Caiolfa, M. Zamai, and E. Gratton. The phasor approach to fluorescence lifetime imaging analysis. *Biophysical journal*, 94(2):L14–L16, 2008.
- [73] E. Dinan, S. Roller, K. Shuster, A. Fan, M. Auli, and J. Weston. Wizard of wikipedia: Knowledge-powered conversational agents. *arXiv preprint arXiv:1811.01241*, 2018.
- [74] Y. Dou, K. Shu, C. Xia, P. S. Yu, and L. Sun. *User Preference-Aware Fake News Detection*, page 2051–2055. Association for Computing Machinery, New York, NY, USA, 2021. ISBN 9781450380379. URL <https://doi.org/10.1145/3404835.3462990>.
- [75] D. Dua and C. Graff. UCI machine learning repository, 2017. URL <http://archive.ics.uci.edu/ml>.
- [76] J. M. Durán and K. R. Jongsma. Who is afraid of black box algorithms? on the epistemological and ethical basis of trust in medical ai. *Journal of Medical Ethics*, 47(5):329–335, 2021.
- [77] A. D’Ulizia, M. C. Caschera, F. Ferri, and P. Grifoni. Fake news detection: a survey of evaluation datasets. *PeerJ Computer Science*, 7:e518, 2021.
- [78] M. Egmont-Petersen, D. de Ridder, and H. Handels. Image processing with neural networks—a review. *Pattern recognition*, 35(10):2279–2301, 2002.
- [79] L. Engstrom, A. Ilyas, S. Santurkar, D. Tsipras, F. Janoos, L. Rudolph, and A. Madry. Implementation matters in deep policy gradients: A case study on ppo and trpo. *arXiv preprint arXiv:2005.12729*, 2020.
- [80] EUGDPR. Gdpr. general data protection regulation., 2017.
- [81] F. Fan, W. Cong, and G. Wang. A new type of neurons for machine learning. *International journal for numerical methods in biomedical engineering*, 34(2):e2920, 2018.
- [82] Z. Fan, Z. Liu, J. Zhang, Y. Xiong, L. Zheng, and P. S. Yu. Continuous-time sequential recommendation with temporal graph collaborative transformer, 2021.

-
- [83] L. Feng, S. Shu, Z. Lin, F. Lv, L. Li, and B. An. Can cross entropy loss be robust to label noise? In C. Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 2206–2212. International Joint Conferences on Artificial Intelligence Organization, 7 2020. doi: 10.24963/ijcai.2020/305. URL <https://doi.org/10.24963/ijcai.2020/305>. Main track.
- [84] M. Fey and J. E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.
- [85] J. Fiedler. Simple modifications to improve tabular neural networks. *arXiv preprint arXiv:2108.03214*, 2021.
- [86] J. Fleiss et al. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382, 1971.
- [87] J. Fonseca, A. Bell, C. Abrate, F. Bonchi, and J. Stoyanovich. Setting the right expectations: Algorithmic recourse over time. *arXiv preprint arXiv:2309.06969*, 2023.
- [88] E. W. Forgy. Cluster analysis of multivariate data: efficiency versus interpretability of classifications. *biometrics*, 21:768–769, 1965.
- [89] F. Fouss, S. Faulkner, M. Kolp, A. Pirotte, M. Saerens, et al. Web recommendation system based on a markov-chainmodel. In *ICEIS (4)*, pages 56–63, 2005.
- [90] F. Fouss, A. Pirotte, and M. Saerens. A novel way of computing similarities between nodes of a graph, with application to collaborative recommendation, 2005.
- [91] L. Fu. Rule generation from neural networks. *IEEE Transactions on Systems, Man, and Cybernetics*, 24(8):1114–1124, 1994.
- [92] Y. Gal and Z. Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016.
- [93] B. Ganter and R. Wille. *Formal concept analysis: mathematical foundations*. Springer Science & Business Media, 2012.
- [94] D. Georgiev, P. Barbiero, D. Kazhdan, P. Veličković, and P. Liò. Algorithmic concept-based explainable reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 6685–6693, 2022.
- [95] A. Ghazimatin, O. Balalau, R. Saha Roy, and G. Weikum. Prince: Provider-side interpretability with counterfactual explanations in recommender systems. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 196–204, 2020.
- [96] M. Ghazvininejad, C. Brockett, M.-W. Chang, B. Dolan, J. Gao, W.-t. Yih, and M. Galley. A knowledge-grounded neural conversation model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

-
- [97] A. Ghorbani, A. Abid, and J. Zou. Interpretation of neural networks is fragile. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3681–3688, 2019.
- [98] A. Ghorbani, J. Wexler, J. Y. Zou, and B. Kim. Towards automatic concept-based explanations. *Advances in Neural Information Processing Systems*, 32, 2019.
- [99] A. Ghosh, N. Manwani, and P. Sastry. Making risk minimization tolerant to label noise. *Neurocomputing*, 160:93–107, 2015. ISSN 0925-2312. doi: <https://doi.org/10.1016/j.neucom.2014.09.081>. URL <https://www.sciencedirect.com/science/article/pii/S0925231215001204>.
- [100] A. Ghosh, H. Kumar, and P. S. Sastry. Robust loss functions under label noise for deep neural networks. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI’17, page 1919–1925. AAAI Press, 2017.
- [101] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR, 2017.
- [102] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
- [103] Y. Gorishniy, I. Rubachev, V. Khruikov, and A. Babenko. Revisiting deep learning models for tabular data. *Advances in Neural Information Processing Systems*, 34:18932–18943, 2021.
- [104] M. Goyani and N. Chaurasiya. A review of movie recommendation system: Limitations, survey and challenges, 2020.
- [105] G. Grani, M. Gentili, F. Siciliano, D. Albano, V. Zilioli, S. Morelli, E. Puxeddu, M. C. Zatelli, I. Gagliardi, A. Piovesan, et al. A data-driven approach to refine predictions of differentiated thyroid cancer outcomes: a prospective multicenter study. *The Journal of Clinical Endocrinology & Metabolism*, page dgad075, 2023.
- [106] F. Greco, A. Polli, F. Siciliano, et al. Leveraging deep learning models to assess the temporal validity of emotional text mining procedures. In *JADT 2022 Proceedings: 16th International Conference on Statistical Analysis of Textual Data*, volume 2, pages 475–481, 2022.
- [107] L. Grinsztajn, E. Oyallon, and G. Varoquaux. Why do tree-based models still outperform deep learning on tabular data? *arXiv preprint arXiv:2207.08815*, 2022.
- [108] M. Gruppi, B. D. Horne, and S. Adah. Nela-gt-2020: A large multi-labelled news dataset for the study of misinformation in news articles. *arXiv preprint arXiv:2102.04567*, 2021.
- [109] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, et al. Recent advances in convolutional neural networks. *Pattern recognition*, 77:354–377, 2018.
- [110] Z. Guo, M. Schlichtkrull, and A. Vlachos. A survey on automated fact-checking. *arXiv preprint arXiv:2108.11896*, 2021.

-
- [111] M. M. Gupta, I. Bukovsky, N. Homma, A. M. Solo, and Z.-G. Hou. Fundamentals of higher order neural networks for modeling and simulation. In *Artificial Higher Order Neural Networks for Modeling and Simulation*, pages 103–133. IGI Global, 2013.
- [112] S. Gupta and A. Gupta. Dealing with noise problem in machine learning data-sets: A systematic review. *Procedia Computer Science*, 161:466–474, 2019.
- [113] T. Hailesilassie. Rule extraction algorithm for deep neural networks: A review. *arXiv preprint arXiv:1610.05267*, 2016.
- [114] O. Halimi, I. Imanuel, O. Litany, G. Trappolini, E. Rodolà, L. Guibas, and R. Kimmel. Towards precise completion of deformable shapes. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIV 16*, pages 359–377. Springer, 2020.
- [115] W. L. Hamilton, R. Ying, and J. Leskovec. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 1025–1035, 2017.
- [116] Y. Han, S. Karunasekera, and C. Leckie. Graph neural networks with continual learning for fake news detection from social media. *arXiv preprint arXiv:2007.03316*, 2020.
- [117] F. M. Harper and J. A. Konstan. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4):1–19, 2015.
- [118] F. M. Harper and J. A. Konstan. The movielens datasets: History and context, dec 2015. ISSN 2160-6455. URL <https://doi.org/10.1145/2827872>.
- [119] Y. Hayashi, R. Setiono, and A. Azcarraga. Neural network training and rule extraction with augmented discretized input. *Neurocomputing*, 207:610–622, 2016.
- [120] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [121] D. Hendrycks, M. Mazeika, D. Wilson, and K. Gimpel. Using trusted data to train deep networks on labels corrupted by severe noise. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL <https://proceedings.neurips.cc/paper/2018/file/ad554d8c3b06d6b97ee76a2448bd7913-Paper.pdf>.
- [122] B. Heo, M. Lee, S. Yun, and J. Y. Choi. Knowledge transfer via distillation of activation boundaries formed by hidden neurons. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3779–3787, 2019.
- [123] B. Hidasi and A. Karatzoglou. Recurrent neural networks with top-k gains for session-based recommendations. In *Proceedings of the 27th ACM international conference on information and knowledge management*, pages 843–852, 2018.

- [124] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*, 2015.
- [125] B. Hidasi, M. Quadrana, A. Karatzoglou, and D. Tikk. Parallel recurrent neural network architectures for feature-rich session-based recommendations, 2016.
- [126] S. Holter, O. Gomez, and E. Bertini. FICO Explainable Machine Learning Challenge. FICO COmmunity, 2018. DOI: <https://community.fico.com/s/explainable-machine-learning-challenge>.
- [127] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, USA, 2nd edition, 2012. ISBN 0521548233.
- [128] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989. ISSN 0893-6080. doi: [https://doi.org/10.1016/0893-6080\(89\)90020-8](https://doi.org/10.1016/0893-6080(89)90020-8). URL <https://www.sciencedirect.com/science/article/pii/0893608089900208>.
- [129] E. R. Hruschka and N. F. Ebecken. Extracting rules from multilayer perceptrons in classification problems: A clustering-based approach. *Neurocomputing*, 70(1-3):384–397, 2006.
- [130] M. Huai, J. Liu, C. Miao, L. Yao, and A. Zhang. Towards automating model explanations with certified robustness guarantees. 2022.
- [131] Q. Huang, J. Yu, J. Wu, and B. Wang. Heterogeneous graph attention networks for early detection of rumors on twitter. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2020.
- [132] D. R. Hush. Classification with neural networks: a performance analysis. In *Proceedings of the IEEE international conference on systems engineering*, pages 277–280. Dayton Ohio, USA, 1989.
- [133] H. Hwangbo, Y. S. Kim, and K. J. Cha. Recommendation system development for fashion retail e-commerce, 2018.
- [134] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and < 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016.
- [135] S.-Y. Ihm, S.-E. Lee, Y.-H. Park, A. Nasridinov, M. Kim, and S.-H. Park. A technique of recursive reliability-based missing data imputation for collaborative filtering. *Applied Sciences*, 11(8):3719, 2021.
- [136] P. Jaccard. The distribution of the flora in the alpine zone. 1. *New phytologist*, 11(2):37–50, 1912.
- [137] H. Jacobsson. Rule extraction from recurrent neural networks: Ataxonomy and review. *Neural Computation*, 17(6):1223–1263, 2005.

- [138] Y. Ji, A. Sun, J. Zhang, and C. Li. A critical study on data leakage in recommender system offline evaluation, 2023.
- [139] A. Kadra, M. Lindauer, F. Hutter, and J. Grabocka. Well-tuned simple nets excel on tabular datasets. *Advances in neural information processing systems*, 34:23928–23941, 2021.
- [140] S. Kamruzzaman, M. Islam, et al. Extraction of symbolic rules from artificial neural networks. *arXiv preprint arXiv:1009.4570*, 2010.
- [141] W.-C. Kang and J. McAuley. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, pages 197–206. IEEE, 2018.
- [142] A.-H. Karimi, G. Barthe, B. Schölkopf, and I. Valera. A survey of algorithmic recourse: definitions, formulations, solutions, and prospects. *arXiv preprint arXiv:2010.04050*, 2020.
- [143] A.-H. Karimi, B. Schölkopf, and I. Valera. Algorithmic recourse: from counterfactual explanations to interventions. In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 353–362, 2021.
- [144] D. Kazhdan, B. Dimanov, M. Jamnik, and P. Liò. Meme: generating rnn model explanations via model extraction. *arXiv preprint arXiv:2012.06954*, 2020.
- [145] D. Kazhdan, B. Dimanov, M. Jamnik, P. Liò, and A. Weller. Now you see me (cme): concept-based model extraction. *arXiv preprint arXiv:2010.13233*, 2020.
- [146] A. H. Khasahmadi, K. Hassani, P. Moradi, L. Lee, and Q. Morris. Memory-based graph networks. In *International Conference on Learning Representations*, 2020.
- [147] A. Khetan, Z. C. Lipton, and A. Anandkumar. Learning from noisy singly-labeled data, 2017. URL <https://arxiv.org/abs/1712.04577>.
- [148] U. Khurana and S. Galhotra. Semantic concept annotation for tabular data. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 844–853, 2021.
- [149] B. Kim, M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, F. Viegas, and R. Sayres. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, page 2668–2677. PMLR, Jul 2018.
- [150] B. Kim, M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, F. Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pages 2668–2677. PMLR, 2018.
- [151] P.-J. Kindermans, S. Hooker, J. Adebayo, M. Alber, K. T. Schütt, S. Dähne, D. Erhan, and B. Kim. The (un) reliability of saliency methods. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pages 267–280. Springer, 2019.

- [152] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.
- [153] P. W. Koh, T. Nguyen, Y. S. Tang, S. Mussmann, E. Pierson, B. Kim, and P. Liang. Concept bottleneck models. In H. D. III and A. Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5338–5348. PMLR, 13–18 Jul 2020. URL <https://proceedings.mlr.press/v119/koh20a.html>.
- [154] P. W. Koh, T. Nguyen, Y. S. Tang, S. Mussmann, E. Pierson, B. Kim, and P. Liang. Concept bottleneck models. In *International Conference on Machine Learning*, pages 5338–5348. PMLR, 2020.
- [155] Ł. Korycki, A. Cano, and B. Krawczyk. Active learning with abstaining classifiers for imbalanced drifting data streams. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 2334–2343. IEEE, 2019.
- [156] R. V. Kulkarni and G. K. Venayagamoorthy. Generalized neuron: Feedforward and recurrent architectures. *Neural networks*, 22(7):1011–1017, 2009.
- [157] S. Kulkarni and S. F. Rodd. Context aware recommendation systems: A review of the state of the art techniques. *Computer Science Review*, 37:100255, 2020.
- [158] P. Kumar and A. Gupta. Active learning query strategies for classification, regression, and clustering: a survey. *Journal of Computer Science and Technology*, 35(4):913–945, 2020.
- [159] L. Lanzanò, I. Coto Hernández, M. Castello, E. Gratton, A. Diaspro, and G. Vicidomini. Encoding and decoding spatio-temporal information for super-resolution microscopy. *Nature communications*, 6(1):6701, 2015.
- [160] A. Lao, C. Shi, and Y. Yang. Rumor detection with field of linear and non-linear propagation. In *Proceedings of the Web Conference 2021*, pages 3178–3187, 2021.
- [161] T. Laugel, A. Jeyasothy, M.-J. Lesot, C. Marsala, and M. Detyniecki. Achieving diversity in counterfactual explanations: a review and discussion. In *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency*, pages 1859–1869, 2023.
- [162] B. Lester, R. Al-Rfou, and N. Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021.
- [163] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474, 2020.
- [164] J. Li, P. Ren, Z. Chen, Z. Ren, T. Lian, and J. Ma. Neural attentive session-based recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1419–1428, 2017.

- [165] J. Li, Y. Wang, and J. McAuley. Time interval aware self-attention for sequential recommendation. In *Proceedings of the 13th international conference on web search and data mining*, pages 322–330, 2020.
- [166] Y. Li, W. Ma, C. Chen, M. Zhang, Y. Liu, S. Ma, and Y. Yang. A survey on dropout methods and experimental verification in recommendation. *arXiv preprint arXiv:2204.02027*, 2022.
- [167] Y. Li, C.-Y. Wu, H. Fan, K. Mangalam, B. Xiong, J. Malik, and C. Feichtenhofer. Mvitv2: Improved multiscale vision transformers for classification and detection, 2022.
- [168] D. Liang, R. G. Krishnan, M. D. Hoffman, and T. Jebara. Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 world wide web conference*, pages 689–698, 2018.
- [169] Z. C. Lipton, J. Berkowitz, and C. Elkan. A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019*, 2015.
- [170] D. Liu, Y. Sun, X. Zhao, G. Zhang, and R. Liu. Adversarial training for session-based item recommendations. In *2020 IEEE 9th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, volume 9, pages 1162–1168. IEEE, 2020.
- [171] J. Liu, Y. Wang, B. Hooi, R. Yang, and X. Xiao. Active learning for node classification: The additional learning ability from unlabelled nodes. *arXiv preprint arXiv:2012.07065*, 2020.
- [172] N. F. Liu, K. Lin, J. Hewitt, A. Paranjape, M. Bevilacqua, F. Petroni, and P. Liang. Lost in the middle: How language models use long contexts. *arXiv preprint arXiv:2307.03172*, 2023.
- [173] Y. Liu and Y.-F. B. Wu. Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks. In *Thirty-second AAAI conference on artificial intelligence*, 2018.
- [174] S. Lo Piano. Ethical principles in machine learning and artificial intelligence: cases from the field and possible ways forward. *Humanities and Social Sciences Communications*, 7(1):1–7, 2020.
- [175] H. Lu, R. Setiono, and H. Liu. Neurorule: A connectionist approach to data mining. *arXiv preprint arXiv:1701.01358*, 2017.
- [176] Y. Lu, M. Bartolo, A. Moore, S. Riedel, and P. Stenetorp. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. *arXiv preprint arXiv:2104.08786*, 2021.
- [177] Y.-J. Lu and C.-T. Li. Gcan: Graph-aware co-attention networks for explainable fake news detection on social media. *arXiv preprint arXiv:2004.11648*, 2020.
- [178] E. Lughofer. On-line active learning: A new paradigm to improve practical useability of data stream modeling methods. *Information Sciences*, 415:356–376, 2017.
- [179] D. Luo, W. Cheng, D. Xu, W. Yu, B. Zong, H. Chen, and X. Zhang. Parameterized explainer for graph neural network. *Advances in neural information processing systems*, 33:19620–19631, 2020.

-
- [180] Z. Luo, D. Chen, Y. Zhang, Y. Huang, L. Wang, Y. Shen, D. Zhao, J. Zhou, and T. Tan. Videofusion: Decomposed diffusion models for high-quality video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023.
- [181] K. Madhawa and T. Murata. Active learning for node classification: an evaluation. *Entropy*, 22(10):1164, 2020.
- [182] L. C. Magister, D. Kazhdan, V. Singh, and P. Liò. Gcexplainer: Human-in-the-loop concept-based explanations for graph neural networks. *arXiv preprint arXiv:2107.11889*, 2021.
- [183] L. C. Magister, P. Barbiero, D. Kazhdan, F. Siciliano, G. Ciravegna, F. Silvestri, M. Jamnik, and P. Liò. Concept distillation in graph neural networks. In *World Conference on Explainable Artificial Intelligence*, pages 233–255. Springer, 2023.
- [184] L. R. Medsker and L. Jain. Recurrent neural networks. *Design and Applications*, 5:64–67, 2001.
- [185] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan. A survey on bias and fairness in machine learning. *ACM computing surveys (CSUR)*, 54(6):1–35, 2021.
- [186] R. Meir and T. Zhang. Generalization error bounds for bayesian mixture algorithms. *J. Mach. Learn. Res.*, 4:839–860, dec 2003. ISSN 1532-4435.
- [187] A. Menon, B. V. Rooyen, C. S. Ong, and B. Williamson. Learning from corrupted binary labels via class-probability estimation. In F. Bach and D. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 125–134, Lille, France, 07–09 Jul 2015. PMLR. URL <https://proceedings.mlr.press/v37/menon15.html>.
- [188] A. K. Menon, A. S. Rawat, S. J. Reddi, and S. Kumar. Can gradient clipping mitigate label noise? In *International Conference on Learning Representations*, 2020. URL <https://openreview.net/forum?id=rk1B76EKPr>.
- [189] S. Messing, C. DeGregorio, B. Hillenbrand, G. King, S. Mahanti, Z. Mukerjee, C. Nayak, N. Persily, B. State, and A. Wilkins. Facebook Privacy-Protected Full URLs Data Set, 2020. URL <https://doi.org/10.7910/DVN/TDOAPG>.
- [190] G. A. Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological review*, 63:81–97, 1956.
- [191] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.
- [192] M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of Machine Learning*. The MIT Press, 2012. ISBN 026201825X.
- [193] R. M. Monarch. *Human-in-the-Loop Machine Learning: Active learning and annotation for human-centered AI*. Simon and Schuster, 2021.

- [194] F. Monti, F. Frasca, D. Eynard, D. Mannion, and M. M. Bronstein. Fake news detection on social media using geometric deep learning. *arXiv preprint arXiv:1902.06673*, 2019.
- [195] C. Morris, N. M. Kriege, F. Bause, K. Kersting, P. Mutzel, and M. Neumann. Tudataset: A collection of benchmark datasets for learning with graphs. *arXiv preprint arXiv:2007.08663*, 2020.
- [196] R. K. Mothilal, A. Sharma, and C. Tan. Explaining machine learning classifiers through diverse counterfactual explanations. *CoRR*, abs/1905.07697, 2019. URL <http://arxiv.org/abs/1905.07697>.
- [197] W.-J. Nam, S. Gur, J. Choi, L. Wolf, and S.-W. Lee. Relative attributing propagation: Interpreting the comparative contributions of individual units in deep neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 2501–2508, 2020.
- [198] N. Natarajan, I. S. Dhillon, P. K. Ravikumar, and A. Tewari. Learning with noisy labels. In C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013. URL <https://proceedings.neurips.cc/paper/2013/file/3871bd64012152bfb53fdf04b401193f-Paper.pdf>.
- [199] V.-H. Nguyen, K. Sugiyama, P. Nakov, and M.-Y. Kan. Fang: Leveraging social context for fake news detection using graph representation. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 1165–1174, 2020.
- [200] J. Ni, J. Li, and J. McAuley. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 188–197, 2019.
- [201] F. Nie, M. Chen, Z. Zhang, and X. Cheng. Improving few-shot performance of language models via nearest neighbor calibration. *arXiv preprint arXiv:2212.02216*, 2022.
- [202] J. Nørregaard, B. D. Horne, and S. Adalı. Nela-gt-2018: A large multi-labelled news dataset for the study of misinformation in news articles. In *Proceedings of the international AAAI conference on web and social media*, volume 13, pages 630–638, 2019.
- [203] I. Nunes and D. Jannach. A systematic review and taxonomy of explanations in decision support and recommender systems. *User Modeling and User-Adapted Interaction*, 27(3):393–444, 2017.
- [204] K. Odajima, Y. Hayashi, G. Tianxia, and R. Setiono. Greedy rule generation from discrete data and its use in neural network rule extraction. *Neural Networks*, 21(7):1020–1028, 2008.
- [205] S. Oh and S. Kumar. Robustness of deep recommendation systems to untargeted interaction perturbations. *arXiv preprint arXiv:2201.12686*, 2022.

- [206] S. Oh, B. Ustun, J. McAuley, and S. Kumar. Rank list sensitivity of recommender systems to interaction perturbations. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 1584–1594, 2022.
- [207] S. J. Oh, B. Schiele, and M. Fritz. Towards reverse-engineering black-box neural networks. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pages 121–144. Springer, 2019.
- [208] M. O’Mahony, N. Hurley, N. Kushmerick, and G. Silvestre. Collaborative recommendation: A robustness analysis. *ACM Transactions on Internet Technology (TOIT)*, 4(4):344–377, 2004.
- [209] K. Ong, S.-C. Haw, and K.-W. Ng. Deep learning based-recommendation system: An overview on models, datasets, evaluation metrics, and future trends. In *Proceedings of the 2019 2nd International Conference on Computational Intelligence and Intelligent Systems*, pages 6–11, 2019.
- [210] OpenAI. Chatgpt: A large-scale language model for conversational ai. *OpenAI Blog*, November 2022.
- [211] OpenAI. Gpt-4 technical report, 2023.
- [212] A. Pal, C. Eksombatchai, Y. Zhou, B. Zhao, C. Rosenberg, and J. Leskovec. Pinnersage: Multi-modal user embedding framework for recommendations at pinterest. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2311–2320, 2020.
- [213] C. Panigutti, R. Hamon, I. Hupont, D. Fernandez Llorca, D. Fano Yela, H. Junklewitz, S. Scalzo, G. Mazzini, I. Sanchez, J. Soler Garrido, et al. The role of explainable ai in the context of the ai act. In *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency*, pages 1139–1150, 2023.
- [214] G. Patrini, A. Rozza, A. Krishna Menon, R. Nock, and L. Qu. Making deep neural networks robust to label noise: A loss correction approach. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1944–1952, 2017.
- [215] M. Pawelczyk, T. Leemann, A. Biega, and G. Kasneci. On the trade-off between actionable explanations and the right to be forgotten. *arXiv preprint arXiv:2208.14137*, 2022.
- [216] V. Pendyala and J. Choi. Concept-based explanations for tabular data. 2022. doi: 10.48550/ARXIV.2209.05690.
- [217] B. PENG. RWKV-LM, Aug. 2021. URL <https://github.com/BlinkDL/RWKV-LM>.
- [218] B. Peng, M. Galley, P. He, H. Cheng, Y. Xie, Y. Hu, Q. Huang, L. Liden, Z. Yu, W. Chen, et al. Check your facts and try again: Improving large language models with external knowledge and automated feedback. *arXiv preprint arXiv:2302.12813*, 2023.
- [219] J. Peterson, R. Battleday, T. Griffiths, and O. Russakovsky. Human uncertainty makes classification more robust. In *Proceedings - 2019 International Conference on Computer Vision*,

-
- ICCV 2019*, Proceedings of the IEEE International Conference on Computer Vision, pages 9616–9625, United States, Oct. 2019. Institute of Electrical and Electronics Engineers Inc. doi: 10.1109/ICCV.2019.00971.
- [220] A. Petrov and C. Macdonald. Effective and efficient training for sequential recommendation using recency sampling. In *Proceedings of the 16th ACM Conference on Recommender Systems*, pages 81–91, 2022.
- [221] J. E. Potter. Matrix quadratic solutions. *SIAM Journal on Applied Mathematics*, 14(3):496–501, 1966. ISSN 00361399. URL <http://www.jstor.org/stable/2946224>.
- [222] M. J. D. Powell. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *The Computer Journal*, 7(2):155–162, 01 1964. ISSN 0010-4620. doi: 10.1093/comjnl/7.2.155. URL <https://doi.org/10.1093/comjnl/7.2.155>.
- [223] A. Purificato, G. Cassarà, P. Liò, and F. Silvestri. Sheaf neural networks for graph-based recommender systems. *arXiv preprint arXiv:2304.09097*, 2023.
- [224] A. Purpura, G. Silvello, and G. A. Susto. Learning to rank from relevance judgments distributions, 2022. URL <https://arxiv.org/abs/2202.06337>.
- [225] M. Quadrana, P. Cremonesi, and D. Jannach. Sequence-aware recommender systems. *ACM Computing Surveys (CSUR)*, 51(4):1–36, 2018.
- [226] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, et al. Improving language understanding by generative pre-training. *OpenAI Blog*, 2018.
- [227] A. Ramponi and B. Plank. Neural unsupervised domain adaptation in nlp—a survey. *arXiv preprint arXiv:2006.00632*, 2020.
- [228] G. Ras, M. van Gerven, and P. Haselager. Explanation methods in deep learning: Users, values, concerns and challenges. In *Explainable and Interpretable Models in Computer Vision and Machine Learning*, pages 19–36. Springer, 2018.
- [229] P. Rasouli and I. C. Yu. CARE: coherent actionable recourse based on sound counterfactual explanations. *CoRR*, abs/2108.08197, 2021. URL <https://arxiv.org/abs/2108.08197>.
- [230] K. Rawal and H. Lakkaraju. Beyond individualized recourse: Interpretable and interactive summaries of actionable recourses. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 12187–12198. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/8ee7730e97c67473a424ccfeff49ab20-Paper.pdf.
- [231] V. C. Raykar, S. Yu, L. H. Zhao, G. H. Valadez, C. Florin, L. Bogoni, and L. Moy. Learning from crowds. *Journal of Machine Learning Research*, 11(43):1297–1322, 2010. URL <http://jmlr.org/papers/v11/raykar10a.html>.
- [232] L. Rayleigh. Xxxi. investigations in optics, with special reference to the spectroscope. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 8(49):261–274, 1879.

- [233] P. Regulation. Regulation (eu) 2016/679 of the european parliament and of the council. *Regulation (eu)*, 679:2016, 2016.
- [234] N. Reimers and I. Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. URL <http://arxiv.org/abs/1908.10084>.
- [235] P. Ren, Y. Xiao, X. Chang, P.-Y. Huang, Z. Li, B. B. Gupta, X. Chen, and X. Wang. A survey of deep active learning. *ACM Computing Surveys (CSUR)*, 54(9):1–40, 2021.
- [236] Y. Ren and J. Zhang. Fake news detection on news-oriented heterogeneous information networks through hierarchical graph attention. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2021.
- [237] Y. Ren, M. Tomko, F. D. Salim, J. Chan, C. L. Clarke, and M. Sanderson. A location-query-browse graph for contextual recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 30(2):204–218, 2017.
- [238] Y. Ren, B. Wang, J. Zhang, and Y. Chang. Adversarial active learning based heterogeneous graph neural network for fake news detection. In *2020 IEEE International Conference on Data Mining (ICDM)*, pages 452–461. IEEE, 2020.
- [239] F. Ricci, L. Rokach, and B. Shapira. Introduction to recommender systems handbook. In *Recommender systems handbook*, pages 1–35. Springer, 2011.
- [240] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, pages 65–386, 1958.
- [241] C. Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019.
- [242] C. Rudin, C. Chen, Z. Chen, H. Huang, L. Semenova, and C. Zhong. Interpretable machine learning: Fundamental principles and 10 grand challenges. *arXiv preprint arXiv:2103.11251*, 2021.
- [243] W. Rudin. *Principles of mathematical analysis*. International series in pure and applied mathematics. McGraw-Hill, New York, 1976.
- [244] W. Rudin. *Real and Complex Analysis*. McGraw-Hill, New York, 1987.
- [245] E. W. Saad and D. C. Wunsch II. Neural network explanation using inversion. *Neural networks*, 20(1):78–93, 2007.
- [246] A. Sauchuk, J. Thorne, A. Halevy, N. Tonello, and F. Silvestri. On the role of relevance in natural language processing tasks. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1785–1789, 2022.

- [247] A. Sauchuk, J. Thorne, A. Y. Halevy, N. Tonello, and F. Silvestri. On the role of relevance in natural language processing tasks. In E. Amigó, P. Castells, J. Gonzalo, B. Carterette, J. S. Culpepper, and G. Kazai, editors, *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*, pages 1785–1789. ACM, 2022. doi: 10.1145/3477495.3532034. URL <https://doi.org/10.1145/3477495.3532034>.
- [248] T. L. Scao, A. Fan, C. Akiki, E. Pavlick, S. Ilić, D. Hesslow, R. Castagné, A. S. Luccioni, F. Yvon, M. Gallé, et al. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*, 2022.
- [249] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008.
- [250] J. B. Schafer, J. A. Konstan, and J. Riedl. E-commerce recommendation applications, 2001.
- [251] M. Schedl, P. Knees, B. McFee, D. Bogdanov, and M. Kaminskas. Music recommender systems, 2015.
- [252] M. Schedl, H. Zamani, C.-W. Chen, Y. Deldjoo, and M. Elahi. Current challenges and visions in music recommender systems research, 2018.
- [253] J. Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61: 85–117, 2015. ISSN 0893-6080. doi: <https://doi.org/10.1016/j.neunet.2014.09.003>. URL <https://www.sciencedirect.com/science/article/pii/S0893608014002135>.
- [254] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
- [255] R. Setiono and H. Liu. Understanding neural networks via rule extraction. In *IJCAI*, volume 1, pages 480–485. Citeseer, 1995.
- [256] R. Setiono and H. Liu. Symbolic representation of neural networks. *Computer*, 29(3):71–77, 1996.
- [257] R. Setiono and H. Liu. Neurolinear: From neural networks to oblique decision rules. *Neuro-computing*, 17(1):1–24, 1997.
- [258] R. Setiono, B. Baesens, and C. Mues. Recursive neural network rule extraction for data with mixed attributes. *IEEE transactions on neural networks*, 19(2):299–307, 2008.
- [259] R. Setiono, A. Azcarraga, and Y. Hayashi. Mofn rule extraction from neural networks trained with augmented discretized input. In *2014 International Joint Conference on Neural Networks (IJCNN)*, pages 1079–1086. IEEE, 2014.
- [260] G. Shani, D. Heckerman, R. I. Brafman, and C. Boutilier. An mdp-based recommender system. *Journal of Machine Learning Research*, 6(9), 2005.
- [261] M. W. Shen. Trust in ai: Interpretability is not necessary or sufficient, while black-box interaction is necessary and sufficient. *arXiv preprint arXiv:2202.05302*, 2022.

-
- [262] J. Sherman and W. Morrison. Abstracts of Papers. *The Annals of Mathematical Statistics*, 20(4):620 – 624, 1949. doi: 10.1214/aoms/1177729959. URL <https://doi.org/10.1214/aoms/1177729959>.
- [263] K. Shu, S. Wang, and H. Liu. Beyond news contents: The role of social context for fake news detection. In *Proceedings of the twelfth ACM international conference on web search and data mining*, pages 312–320, 2019.
- [264] K. Shu, D. Mahudeswaran, S. Wang, D. Lee, and H. Liu. Fakenewsnet: A data repository with news content, social context, and spatiotemporal information for studying fake news on social media. *Big data*, 8(3):171–188, 2020.
- [265] K. Shu, D. Mahudeswaran, S. Wang, and H. Liu. Hierarchical propagation networks for fake news detection: Investigation and exploitation. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 14, pages 626–637, 2020.
- [266] K. Shu, G. Zheng, Y. Li, S. Mukherjee, A. H. Awadallah, S. Ruston, and H. Liu. Leveraging multi-source weak social supervision for early detection of fake news. *arXiv preprint arXiv:2004.01732*, 2020.
- [267] R. Shwartz-Ziv and A. Armon. Tabular data: Deep learning is not all you need. *Information Fusion*, 81:84–90, 2022.
- [268] F. Siciliano, G. Consolini, R. Tozzi, M. Gentili, F. Giannattasio, and P. De Michelis. Forecasting sym-h index: A comparison between long short-term memory and convolutional neural networks. *Space Weather*, 19(2):e2020SW002589, 2021.
- [269] F. Siciliano, M. S. Bucarelli, G. Tolomei, and F. Silvestri. Newron: a new generalization of the artificial neuron to enhance the interpretability of neural networks. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 01–17. IEEE, 2022.
- [270] F. Siciliano, C. Abrate, F. Bonchi, and F. Silvestri. Human-in-the-loop personalized counterfactual recourse. In *Submitted to International Conference on Artificial Intelligence and Statistics (AISTATS) 2024*, 2023.
- [271] F. Siciliano, A. Bacciu, N. Tonello, and F. Silvestri. Integrating item relevance in training loss for sequential recommender systems. In *Proceedings of the 17th ACM Conference on Recommender Systems, RecSys '23*, page 1114–1119, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400702419. doi: 10.1145/3604915.3610643. URL <https://doi.org/10.1145/3604915.3610643>.
- [272] F. Siciliano, S. Lagziel, and G. Gamzu, Iftah Tolomei. Robust training of sequential recommender systems with missing input data. In *Submitted to Information Processing and Management*, 2023.
- [273] F. Siciliano, L. C. Magister, M. S. Bucarelli, P. Barbiero, F. Silvestri, and P. Lio. Explaining neural networks using a ruleset based on interpretable concepts. In *Submitted to EPJ Data Science*, 2023.

-
- [274] F. Siciliano, L. Maiano, L. Papa, F. Baccini, I. Amerini, and F. Silvestri. Adversarial data poisoning for fake news detection: How to make a model misclassify a target news without modifying it. In *ECML-PKDD Deep Learning and Multimedia Forensics Workshop*, 2023.
- [275] A. Silva, Y. Han, L. Luo, S. Karunasekera, and C. Leckie. Propagation2vec: Embedding partial propagation networks for explainable fake news early detection. *Information Processing & Management*, 58(5):102618, 2021.
- [276] S. Sirur, J. R. Nurse, and H. Webb. Are we there yet? understanding the challenges faced in complying with the general data protection regulation (gdpr). In *Proceedings of the 2nd International Workshop on Multimedia Privacy and Security*, pages 88–95, 2018.
- [277] J. Sokolić, R. Giryes, G. Sapiro, and M. R. Rodrigues. Robust large margin deep neural networks. *IEEE Transactions on Signal Processing*, 65(16):4265–4280, 2017.
- [278] C. Song, K. Shu, and B. Wu. Temporally evolving graph neural network for fake news detection. *Information Processing & Management*, 58(6):102712, 2021.
- [279] H. Song, M. Kim, D. Park, Y. Shin, and J.-G. Lee. Learning from noisy labels with deep neural networks: A survey, 2020. URL <https://arxiv.org/abs/2007.08199>.
- [280] E. M. Stein and R. Shakarchi. *Real analysis: measure theory, integration, and Hilbert spaces*. Princeton lectures in analysis. Princeton Univ. Press, Princeton, NJ, 2005.
- [281] J. M. Stokes, K. Yang, K. Swanson, W. Jin, A. Cubillos-Ruiz, N. M. Donghia, C. R. MacNair, S. French, L. A. Carfrae, Z. Bloom-Ackermann, et al. A deep learning approach to antibiotic discovery. *Cell*, 180(4):688–702, 2020.
- [282] A. Sun. From counter-intuitive observations to a fresh look at recommender system, 2022.
- [283] F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, and P. Jiang. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 1441–1450, 2019.
- [284] S. Sun, Z. Cao, H. Zhu, and J. Zhao. A survey of optimization methods from a machine learning perspective. *IEEE transactions on cybernetics*, 50(8):3668–3681, 2019.
- [285] J. Tan, S. Xu, Y. Ge, Y. Li, X. Chen, and Y. Zhang. Counterfactual explainable recommendation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 1784–1793, 2021.
- [286] J. Tang and K. Wang. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the eleventh ACM international conference on web search and data mining*, pages 565–573, 2018.
- [287] J. Tang, Y. Drori, D. Chang, M. Sathiamoorthy, J. Gilmer, L. Wei, X. Yi, L. Hong, and E. H. Chi. Improving training stability for multitask ranking models in recommender systems. *arXiv preprint arXiv:2302.09178*, 2023.

-
- [288] J. Thorne, M. Yazdani, M. Saeidi, F. Silvestri, S. Riedel, and A. Halevy. Database reasoning over text. *arXiv preprint arXiv:2106.01074*, 2021.
- [289] J. Thorne, M. Yazdani, M. Saeidi, F. Silvestri, S. Riedel, and A. Halevy. From natural language processing to neural databases. In *Proceedings of the VLDB Endowment*, volume 14, pages 1033–1039. VLDB Endowment, 2021.
- [290] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample. Llama: Open and efficient foundation language models, 2023.
- [291] K. H. Tran, A. Ghazimatin, and R. Saha Roy. Counterfactual explanations for neural recommenders. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1627–1631, 2021.
- [292] T. N. T. Tran, A. Felfernig, C. Trattner, and A. Holzinger. Recommender systems in the healthcare domain: state-of-the-art and research issues. *Journal of Intelligent Information Systems*, 57(1):171–201, 2021.
- [293] G. Trappolini, L. Cosmo, L. Moschella, R. Marin, S. Melzi, and E. Rodolà. Shape registration in the time of transformers. *Advances in Neural Information Processing Systems*, 34:5731–5744, 2021.
- [294] G. Trappolini, A. Santilli, E. Rodolà, A. Halevy, and F. Silvestri. Multimodal neural databases. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, page 2619–2628, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9781450394086. doi: 10.1145/3539618.3591930. URL <https://doi.org/10.1145/3539618.3591930>.
- [295] N. Tsopze, E. Mephu-Nguifo, and G. Tindo. Towards a generalization of decompositional approach of rule extraction from multilayer artificial neural network. In *The 2011 International Joint Conference on Neural Networks*, pages 1562–1569. IEEE, 2011.
- [296] H. Tsukimoto. Extracting rules from trained neural networks. *IEEE Transactions on Neural networks*, 11(2):377–389, 2000.
- [297] A. Uma, T. Fornaciari, D. Hovy, S. Paun, B. Plank, and M. Poesio. A case for soft loss functions. *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, 8(1):173–177, Oct. 2020. URL <https://ojs.aaai.org/index.php/HCOMP/article/view/7478>.
- [298] E. Union. Proposal for a regulation of the european parliament and of the council laying down harmonised rules on artificial intelligence (artificial intelligence act) and amending certain union legislative acts. *COM/2021/206final*, 2021.
- [299] E. Union. Proposal for a directive of the european parliament and of the council on adapting non-contractual civil liability rules to artificial intelligence (ai liability directive). *COM/2022/496 final*, 2022.

-
- [300] V. N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 1998.
- [301] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.
- [302] S. Verma, V. Boonsanong, M. Hoang, K. E. Hines, J. P. Dickerson, and C. Shah. Counterfactual explanations and algorithmic recourses for machine learning: A review. *arXiv preprint arXiv:2010.10596*, 2020.
- [303] J. Vinagre, A. M. Jorge, and J. Gama. Online bagging for recommender systems. *Expert Systems*, 35(4):e12303, 2018.
- [304] P. Voigt and A. Von dem Bussche. The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, 10(3152676):10–5555, 2017.
- [305] S. Vosoughi, D. Roy, and S. Aral. The spread of true and false news online. *Science*, 359(6380):1146–1151, 2018.
- [306] M. Vu and M. T. Thai. Pgm-explainer: Probabilistic graphical model explanations for graph neural networks. *Advances in neural information processing systems*, 33:12225–12235, 2020.
- [307] S. Wachter, B. Mittelstadt, and C. Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harv. JL & Tech.*, 31:841, 2017.
- [308] M. Y. Wang. Deep graph library: Towards efficient and scalable deep learning on graphs. In *ICLR workshop on representation learning on graphs and manifolds*, 2019.
- [309] S. Wang, L. Hu, Y. Wang, L. Cao, Q. Z. Sheng, and M. Orgun. Sequential recommender systems: challenges, progress and prospects, 2019.
- [310] W. Wang, F. Feng, X. He, L. Nie, and T.-S. Chua. Denoising implicit feedback for recommendation, 2021.
- [311] X. Wang, Y. Chen, J. Yang, L. Wu, Z. Wu, and X. Xie. A reinforcement learning framework for explainable recommendation. In *2018 IEEE international conference on data mining (ICDM)*, pages 587–596. IEEE, 2018.
- [312] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua. Neural graph collaborative filtering, 2019.
- [313] Y. Wang, X. Ma, Z. Chen, Y. Luo, J. Yi, and J. Bailey. Symmetric cross entropy for robust learning with noisy labels. In *IEEE International Conference on Computer Vision*, 2019.
- [314] Z. Wang, J. Zhang, H. Xu, X. Chen, Y. Zhang, W. X. Zhao, and J.-R. Wen. Counterfactual data-augmented sequential recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 347–356, 2021.
- [315] Z. J. Wang, J. W. Vaughan, R. Caruana, and D. H. Chau. Gam coach: Towards interactive and user-centered algorithmic recourse. In *Proceedings of the 2023 CHI Conference on Human*

-
- Factors in Computing Systems*. ACM, apr 2023. doi: 10.1145/3544548.3580816. URL <https://doi.org/10.1145/3544548.3580816>.
- [316] D. S. Watson and L. Floridi. The explanation game: a formal framework for interpretable machine learning. In *Ethics, Governance, and Policies in Artificial Intelligence*, pages 185–219. Springer, 2021.
- [317] J. Wei, Z. Zhu, T. Luo, E. Amid, A. Kumar, and Y. Liu. To aggregate or not? learning with separate noisy labels, 2022. URL <https://arxiv.org/abs/2206.07181>.
- [318] D. M. West. *The Future of Work: Robots, AI, and Automation*. Brookings Institution Press, 2018. ISBN 9780815732938. URL <http://www.jstor.org/stable/10.7864/j.ctt1vjqp2g>.
- [319] L. Wheeden, Richard and A. Zygmund. *Measure and integral: An introduction to real analysis*. CRC Press, 2015. Second ediction.
- [320] F. Wilcoxon. Individual comparisons by ranking methods. In *Breakthroughs in Statistics: Methodology and Distribution*, pages 196–202. Springer, 1992.
- [321] Q. Wu, Y. Liu, C. Miao, B. Zhao, Y. Zhao, and L. Guan. Pd-gan: Adversarial learning for personalized diversity-promoting recommendation. In *IJCAI*, volume 19, pages 3870–3876, 2019.
- [322] X. Wu, L. Xiao, Y. Sun, J. Zhang, T. Ma, and L. He. A survey of human-in-the-loop for machine learning. *CoRR*, abs/2108.00941, 2021. URL <https://arxiv.org/abs/2108.00941>.
- [323] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1): 4–24, 2020.
- [324] W. Xia, L. He, J. Gu, and K. He. Effective collaborative filtering approaches based on missing data imputation. In *2009 Fifth International Joint Conference on INC, IMS and IDC*, pages 534–537. IEEE, 2009.
- [325] X. Xia, T. Liu, N. Wang, B. Han, C. Gong, G. Niu, and M. Sugiyama. Are anchor points really indispensable in label-noise learning? In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL <https://proceedings.neurips.cc/paper/2019/file/9308b0d6e5898366a4a986bc33f3d3e7-Paper.pdf>.
- [326] Z. Xie, S. Singh, J. McAuley, and B. P. Majumder. Factual and informative review generation for explainable recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 13816–13824, 2023.
- [327] P. Yadav, P. Hase, and M. Bansal. Low-cost algorithmic recourse for users with uncertain cost functions. *CoRR*, abs/2111.01235, 2021. URL <https://arxiv.org/abs/2111.01235>.
- [328] D. Yang, D. Zhang, V. W. Zheng, and Z. Yu. Modeling user activity preference by leveraging user spatial temporal characteristics in lbsns, 2014.

- [329] L. Yang, Y. Zheng, X. Cai, H. Dai, D. Mu, L. Guo, and T. Dai. A lstm based model for personalized context-aware citation recommendation. *IEEE access*, 6:59618–59627, 2018.
- [330] X. Yang, Y. Lyu, T. Tian, Y. Liu, Y. Liu, and X. Zhang. Rumor detection on social media with graph structured adversarial learning. In *IJCAI*, pages 1417–1423, 2020.
- [331] Y. Yang, I. G. Morillo, and T. M. Hospedales. Deep neural decision trees. *arXiv preprint arXiv:1806.06988*, 2018.
- [332] Y. Yang, R. Khanna, Y. Yu, A. Gholami, K. Keutzer, J. E. Gonzalez, K. Ramchandran, and M. W. Mahoney. Boundary thickness and robustness in learning models. *Advances in Neural Information Processing Systems*, 33:6223–6234, 2020.
- [333] Y. Yao, T. Liu, B. Han, M. Gong, J. Deng, G. Niu, and M. Sugiyama. Dual t: Reducing estimation error for transition matrix in label-noise learning. In *Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS’20*, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.
- [334] C.-K. Yeh, B. Kim, S. Arik, C.-L. Li, T. Pfister, and P. Ravikumar. On completeness-aware concept-based explanations in deep neural networks. *Advances in Neural Information Processing Systems*, 33:20554–20565, 2020.
- [335] J. Yetukuri, I. Hardy, and Y. Liu. Towards user guided actionable recourse. In *Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society*, pages 742–751, 2023.
- [336] Z. Ying, D. Bourgeois, J. You, M. Zitnik, and J. Leskovec. Gnnexplainer: Generating explanations for graph neural networks. *Advances in neural information processing systems*, 32, 2019.
- [337] J. Yu, Q. Huang, X. Zhou, and Y. Sha. Iarnet: An information aggregating and reasoning network over heterogeneous graph for fake news detection. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–9. IEEE, 2020.
- [338] C. Yuan, Q. Ma, W. Zhou, J. Han, and S. Hu. Early detection of fake news by utilizing the credibility of news, publishers, and users based on weakly supervised learning. *arXiv preprint arXiv:2012.04233*, 2020.
- [339] H. Zamani, F. Diaz, M. Dehghani, D. Metzler, and M. Bendersky. Retrieval-enhanced machine learning. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2875–2886, 2022.
- [340] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*, 2017. URL <https://openreview.net/forum?id=Sy8gdB9xx>.
- [341] J. Zhang, B. Dong, and S. Y. Philip. Fakedetector: Effective fake news detection with deep diffusive neural network. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*, pages 1826–1829. IEEE, 2020.

- [342] M. Zhang, J. Lee, and S. Agarwal. Learning from noisy labels with no change to the training process. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 12468–12478. PMLR, 18–24 Jul 2021. URL <https://proceedings.mlr.press/v139/zhang21k.html>.
- [343] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.
- [344] S. Zhang, L. Yao, A. Sun, and Y. Tay. Deep learning based recommender system: A survey and new perspectives, 2019.
- [345] S. Zhang, D. Yao, Z. Zhao, T.-S. Chua, and F. Wu. Causerec: Counterfactual user sequence synthesis for sequential recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 367–377, 2021.
- [346] Y. Zhang, X. Chen, et al. Explainable recommendation: A survey and new perspectives. *Foundations and Trends® in Information Retrieval*, 14(1):1–101, 2020.
- [347] Y. Zhang, S. Sun, X. Gao, Y. Fang, C. Brockett, M. Galley, J. Gao, and B. Dolan. Retgen: A joint framework for retrieval and grounded text generation modeling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11739–11747, 2022.
- [348] Z. Zhang and M. R. Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems, NIPS’18*, page 8792–8802, Red Hook, NY, USA, 2018. Curran Associates Inc.
- [349] Z. Zhang, Q. Liu, H. Wang, C. Lu, and C. Lee. Protggn: Towards self-explaining graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 9127–9135, 2022.
- [350] W. X. Zhao, Z. Lin, Z. Feng, P. Wang, and J.-R. Wen. A revisiting study of appropriate offline evaluation for top-n recommendation algorithms, 2022.
- [351] B. Zhou, Y. Sun, D. Bau, and A. Torralba. Interpretable basis decomposition for visual explanation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 119–134, 2018.
- [352] C. Zhou, J. Bai, J. Song, X. Liu, Z. Zhao, X. Chen, and J. Gao. Atrank: An attention-based user behavior modeling framework for recommendation, 2018.
- [353] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun. Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81, 2020.
- [354] J. Zhou, A. H. Gandomi, F. Chen, and A. Holzinger. Evaluating the quality of machine learning explanations: A survey on methods and metrics. *Electronics*, 10(5):593, 2021.

- [355] X. Zhou and R. Zafarani. Network-based fake news detection: A pattern-driven approach. *ACM SIGKDD explorations newsletter*, 21(2):48–60, 2019.
- [356] Z. Zhu, Y. Song, and Y. Liu. Clusterability as an alternative to anchor points when learning with noisy labels, 2021. URL <https://arxiv.org/abs/2102.05291>.
- [357] Z. Zhu, J. Wang, and Y. Liu. Beyond images: Label noise transition matrix estimation for tasks with lower-quality features. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 27633–27653. PMLR, 17–23 Jul 2022. URL <https://proceedings.mlr.press/v162/zhu22k.html>.
- [358] J. R. Zilke, E. Loza Mencía, and F. Janssen. Deepred–rule extraction from deep neural networks. In *International conference on discovery science*, pages 457–473. Springer, 2016.

Appendix A

NEWRON- Supplementary Materials

1.0.1 Universality Theorems

This is the appendix to the Universality section in the main article. In this section, we shall prove the mathematical results concerning the universal approximation properties of our IAN model. In particular, we restrict ourselves to some specific cases. We consider the cases where the processing function is the Heaviside function, a continuous sigmoidal function, or the rescaled product of hyperbolic tangents.

Heaviside IAN

Theorem 5.1. *The finite sums of the form*

$$\psi(x) = \sum_{j=1}^N \alpha_j H(w_j \sum_{i=1}^n H(w_{ij}(x_i - b_{ij})) - b_j) \quad (\text{A.1})$$

with $N \in \mathbb{N}$ and $w_{ij}, w_j, \alpha_j, b_{ij}, b_j \in \mathbb{R}$ are dense in $L^p(A, \mu)$ for $1 \leq p < \infty$, for any $A \in \mathcal{B}(\mathbb{R}^n)$ (\mathcal{B} denote the Borel σ -algebra) and μ a Radon measure on $(A, \mathcal{B}(A))$.

In other words given, $g \in L^p(A, \mu)$ and $\epsilon > 0$ there is a sum $\psi(x)$ of the above form for which

$$\|\psi - g\|_p^p = \int_{\mathbb{R}^n} |\psi(x) - g(x)|^p d\mu(x) < \epsilon.$$

To prove that a neural network defined as in equation (A.1) is a universal approximator in L^p , for $1 \leq p < \infty$ we exploit that step functions are dense in L^p and that our network can generate step functions.

Proposition 1.0.1.1. Let \mathcal{R} be the set of the rectangles in \mathbb{R}^n of the form

$$R = \prod_{k=1}^n [a_k, b_k) \quad a_k, b_k \in \mathbb{R}, \quad a_k < b_k$$

We denote by \mathcal{F} the vector space on \mathbb{R} generated by $\mathbf{1}_R, R \in \mathcal{R}$ i.e.

$$\mathcal{F} = \left\{ \sum_{i=1}^m \alpha_i \mathbf{1}_{R_i} \mid m \in \mathbb{N}, \alpha_i \in \mathbb{R}, R_i \in \mathcal{R} \right\} \quad (\text{A.2})$$

If $A \in \mathcal{B}(\mathbb{R}^n)$, then the set

$$\mathcal{F}|_A = \left\{ \sum_{i=1}^m \alpha_i \mathbb{1}_{R_i \cap A} \mid m \in \mathbb{N}, \alpha_i \in \mathbb{R}, R_i \in \mathcal{R} \right\} \quad (\text{A.3})$$

$\mathcal{F}|_A$ is dense in $L^p(A, \mu)$ for $1 \leq p < \infty$, with μ a Radon measure on $(A, \mathcal{B}(A))$.

Proof. See chapter 3, L^p Spaces, in [40]. □

Lemma 5.2. Given $\rho(x) \in \mathcal{F}$, with \mathcal{F} defined as in equation (A.2), there exists a finite sum $\psi(x)$ of the form (A.1) such that $\rho(x) = \psi(x) \forall x \in \mathbb{R}^n$.

Proof. To prove that a neural network described as in equation (A.1) can generate step functions we proceed in two steps. First, we show how we can obtain the indicator functions of orthants from the first layer of the network. Then we show how, starting from these, we can obtain the step functions.

An *orthant* is the analogue in n -dimensional Euclidean space of a quadrant in \mathbb{R}^2 or an octant in \mathbb{R}^3 . We denote by *translated orthant* an orthant with origin in a point different from the origin of the Euclidean space O . Let A be a point in the n -dimensional Euclidean space, and let us consider the intersection of n mutually orthogonal half-spaces intersecting in A . By independent selections of half-space signs with respect to A (i.e. to the right or left of A) 2^n orthants are formed.

Now we shall see how to obtain translated orthant with origin in a point A of coordinates (a_1, a_2, \dots, a_n) from the first layer of the network i.e. $\sum_{i=1}^n H(w_i(x_i - b_i))$.

For this purpose we can take $w_i = 1 \quad \forall i \in \{1, \dots, n\}$.

The output of $\sum_{i=1}^n H(x_i - b_i) \in \{0, \dots, n\}$ and depends on how many of the n Heaviside functions are activated. We obtain the translated orthant with origin in A by choosing $b_i = a_i \quad \forall i \in \{1, \dots, n\}$. In fact,

$$H(x_i - a_i) = \begin{cases} 0 & \text{if } x_i < a_i \\ 1 & \text{if } x_i \geq a_i. \end{cases}$$

The i -th Heaviside is active in the half-space $x_i \geq a_i$ delimited by the hyperplane $x_i = a_i$ that is orthogonal to the i -th axis. Therefore, the Euclidian space \mathbb{R}^n is divided in 2^n regions according to which value the function $\sum_{i=1}^n H(x_i - a_i)$ takes in each region. See Figure A.1 for an example in \mathbb{R}^2 .

There is only one region in which the output of the sum is n , which corresponds to the orthant in which the condition $x_i \geq a_i \forall i = 1, \dots, n$ holds. We denote it as *positive orthant* (the red colored orthant in the example shown in Figure A.1).

Going back to equation (A.1), let us now consider the Heaviside function applied after the sum. As before, we can choose $w_j = 1$. If we take $b_j > n - 1$, the value of the output is 0 for each of the 2^n orthants except for the positive orthant. This way, we get the indicator function of the positive orthant.

The indicator function of a rectangle in \mathcal{R} can be obtained as a linear combination of the indicator function of the positive orthants centered in the vertices of the rectangle. See Figure A.2 for an example of the procedure in \mathbb{R}^2 .

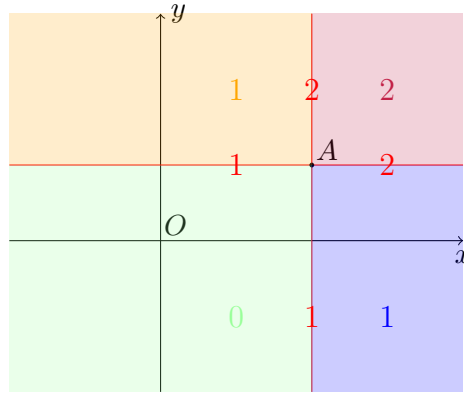


Figure A.1: Partition of \mathbb{R}^2 , according to output of the function $H(x_1 - a_1) + H(x_2 - a_2)$. A is a point of coordinates (a_1, a_2) .

In general, the procedure involves considering a linear combination of indicator functions of positive orthants centered in the vertices of the rectangle in such a way that opposite values are assigned to the orthants corresponding to adjacent vertices.

For example, suppose we want to obtain the indicator function of the right-closed left-open square $[0, 1]^2$ in \mathbb{R}^2 (see the illustration in Figure A.2). Denoting by $\mathbb{1}_{(x_P, y_P)_\leftarrow}$ the indicator function of the positive orthant centered in the point of coordinates (x_P, y_P) , we can write:

$$\mathbb{1}_{[0,1]^2} = \mathbb{1}_{(0,0)_\leftarrow} - \mathbb{1}_{(1,0)_\leftarrow} - \mathbb{1}_{(0,1)_\leftarrow} + \mathbb{1}_{(1,1)_\leftarrow}.$$

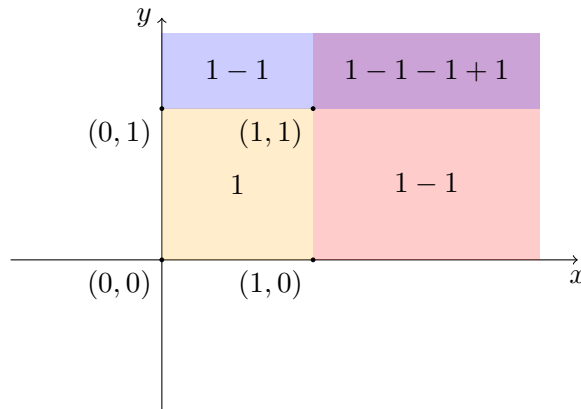


Figure A.2: How to obtain the indicator function on the square $[0, 1]^2$ from the linear combination of four indicator functions of positive orthants centered in the vertices of $[0, 1]^2$. $\mathbb{1}_{[0,1]^2} = \mathbb{1}_{(0,0)_\leftarrow} - \mathbb{1}_{(1,0)_\leftarrow} - \mathbb{1}_{(0,1)_\leftarrow} + \mathbb{1}_{(1,1)_\leftarrow}$. The numbers in the orthants shows the sum of the indicator functions that are active in that orthant. For instance if $x = (x_1, x_2)$ belongs to the blue part of the plane, i.e. it is true that $0 < x_1 < 1$ and $x_2 > 1$, we have that $\mathbb{1}_{(0,0)_\leftarrow}(x) - \mathbb{1}_{(1,0)_\leftarrow}(x) - \mathbb{1}_{(0,1)_\leftarrow}(x) + \mathbb{1}_{(1,1)_\leftarrow}(x) = 1 - 0 - 1 + 0 = 1 - 1$.

Now suppose we want the linear combination of the indicator functions of K rectangles with coefficients $\alpha_1, \dots, \alpha_K$. With suitably chosen coefficients the indicator function of a rectangle can be written as

$$\sum_{l=1}^{2^n} (-1)^l H(w_{jl} \sum_{i=1}^n H(w_{ij}(x_i - b_{ij})) - b_{jl})$$

that replacing $H(w_{jl} \sum_{i=1}^n H(w_{ij}(x_i - b_{ij})) - b_{jl})$ by H_l , to abbreviate the notation becomes

$$\sum_{l=1}^{2^n} (-1)^l H_l.$$

The linear combination of the indicator functions of K rectangles with coefficients $\alpha_1, \dots, \alpha_K$ can be derived as

$$\sum_{k=1}^K \alpha_k \sum_{l=1}^{2^n} (-1)^l H_{lk}. \quad (\text{A.4})$$

The summation (A.4) can be written as a single sum, defining a sequence $\beta_j = (-1)^j \alpha_m$ with $m = \lceil \frac{j}{2^n} \rceil$ for $j = 1, \dots, 2^n K$. Thus (A.4) becomes

$$\sum_{j=1}^{N=2^n K} \beta_j H_j$$

that is an equation of form (A.1). We have therefore shown that for every step function ρ in \mathcal{F} there are $N \in \mathbb{N}$ and $\alpha_j, w_{ij}, b_{ij}, b_j, w_j \in \mathbb{R}$ such that the sum in equation (A.1) is equal to ρ . \square

Proof of Theorem 5.1. The theorem follows immediately from Lemma 5.2 and Proposition 1.0.1.1. \square

Remark 2. In Lemma 5.2 we proved that a network defined as in equation (A.1) can represent functions belonging to set \mathcal{F} defined as in equation (A.2). Note that if the input is a set A we can obtain functions belonging to set $\mathcal{F}|_A$. For instance, suppose $x \in [0, 1]^n$, we can obtain indicator functions of other kinds of sets. If we choose $w_{ij} = 1$ and $b_{ij} < 0 \forall i, j$ and if we choose the weights of the second layer so that they don't operate any transformation, we can obtain the indicator function of $[0, 1]^n$. By a suitable choice of parameters, (A.1) may also become the indicator functions of any hyperplane $x_i = 0$ or $x_i = 1$ for $i \in \{1, \dots, n\}$. Furthermore we can obtain any rectangle of dimension $n - 1$ that belongs to an hyperplane of the form $x_i = 1$ or $x_i = 0$.

We have proven in Lemma 5.2 that a network formulated as in equation (A.1) can represent step functions. By this property and by Proposition 1.0.1.2 we shall show that it can approximate Lebesgue measurable functions on any finite space, for example the unit n -dimensional cube $[0, 1]^n$.

We denote by I_n the closed n -dimensional cube $[0, 1]^n$. We denote by M^n the set of measurable functions with respect to Lebesgue measure m , on I_n , with the metric d_m defined as follows. Let be $f, g \in M^n$,

$$d_m(f, g) = \inf\{\epsilon > 0 : m\{x : |f(x) - g(x)| > \epsilon\} < \epsilon\}$$

We remark that d_m -convergence is equivalent to convergence in measure (see Lemma 2.1 in [128]).

Theorem 5.3. *The finite sums of the form (A.1) with $N \in \mathbb{N}$ and $w_{ij}, w_j, \alpha_j, b_{ij}, b_j \in \mathbb{R}$ are d_m -dense in M^n . M^n is the set of Lebesgue measurable functions on I_n .*

This means that, given g measurable with respect to the Lebesgue measure m on I_n , and given an $\epsilon > 0$, there is a sum ψ of the form (A.1) such that $d_m(\psi, g) < \epsilon$.

Proposition 1.0.1.2. Suppose f is measurable on \mathbb{R}^n . Then there exists a sequence of step functions $\{\rho_k\}_{k=1}^{\infty}$ that converges pointwise to $f(x)$ for almost every x .

Proof. See Theorem 4.3 p. 32 in [280]. □

Proof of Theorem 5.3. Given any measurable function, by Proposition 1.0.1.2 there exists a sequence of step functions that converge to it pointwise. By Lemma 5.2 we have that equation (A.1) can generate step functions. Now $m(I_n) = 1$ and for a finite measure space pointwise convergence implies convergence in measure, this concludes the proof. □

Remark 3. Notice that for Theorem 5.3 we don't need the fact that I_n , is a closed set. The proof only requires that the domain is a bounded set (so that its Lebesgue measure is finite). The compactness of I_n will be necessary for the next theorem.

We notice furthermore that if the function we want to approximate is in L^p we can obtain the convergence in measure from Theorem 5.1. Indeed from Chebyshev's inequality it follows that convergence in L^p implies convergence in measure.

Theorem 5.4. Given $g \in C(I_n)$ and given $\epsilon > 0$ there is a sum $\psi(x)$ of the form (A.1) such that

$$|\psi(x) - g(x)| < \epsilon \quad \forall x \in I_n.$$

Proof. Let g be a continuous function from I_n to \mathbb{R} , by the compactness of I_n follows that g is also uniformly continuous (see Theorem 4.19 p. 91 in [243]). In other words, for any $\epsilon > 0$, there exists $\delta > 0$ such that for every $x, x' \in [0, 1]^n$ such that $\|x - x'\|_{\infty} < \delta$ it is true that $|g(x) - g(x')| < \epsilon$. To prove the statement of Theorem 5.4, let $\epsilon > 0$ be given, and let $\delta > 0$ be chosen according to the definition of uniform continuity.

As we have already seen in Lemma 5.2 the neural network described in (A.1) can generate step functions with support on right-open left-closed n -dimensional rectangles and on $(n-1)$ -dimensional rectangles that belongs to an hyperplane of equation $x_i = 0$ or $x_i = 1$ for some $i \in \{1, \dots, n\}$ as seen in Remark 2. There exists a partition of $[0, 1]^n$, (R_1, \dots, R_N) , consisting of right-open left-closed n -dimensional rectangles and of $(n-1)$ -dimensional rectangles that belongs to an hyperplane of equation $x_i = 0$ or $x_i = 1$ for some $i \in \{1, \dots, n\}$, such that all side lengths are no greater than δ . An example of a set of rectangles with this property is the set of right-open left-closed cubes of side length $\frac{1}{m}$, $\tilde{m} > \lceil \frac{1}{\delta} \rceil$ with the $(n-1)$ -dimensional rectangles with the same side length which we need to cover all the boundary of $[0, 1]^n$ not covered by the right-open left-closed rectangles.

Suppose that for all $j \in \{1, \dots, N\}$ we choose $x_j \in R_j$, and we set $\alpha_j = g(x_j)$. If $x \in [0, 1]^n$ there is j so that $x \in R_j$, hence x satisfies $\|x - x_j\|_{\infty} \leq \delta$, and consequentially $|g(x) - g(x_j)| \leq \epsilon$. Therefore the step function $h = \sum_{j=1}^N \alpha_j \mathbb{1}_{R_j}$ satisfies

$$\begin{aligned} & \sup_{x \in I_n} |h(x) - g(x)| = \\ &= \sup_{j \in \{1, \dots, N\}} \sup_{x \in R_j} |h(x) - g(x)| = \\ &= \sup_{j \in \{1, \dots, N\}} \sup_{x \in R_j} |\alpha_j - g(x)| \leq \epsilon \end{aligned}$$

□

Sigmoid IAN

Definition 1.0.1.3. A function $\sigma : \mathbb{R} \rightarrow [0, 1]$ is called sigmoidal if

$$\lim_{x \rightarrow -\infty} \sigma(x) = 0, \quad \lim_{x \rightarrow +\infty} \sigma(x) = 1$$

Theorem 5.5. Let σ be a continuous sigmoidal function. Then the finite sums of the form:

$$\psi(x) = \sum_{j=1}^N \alpha_j \sigma(w_j (\sum_{i=1}^n \sigma(w_{ij}(x_i - b_{ij})) - b_j)) \quad (\text{A.5})$$

with $w_{ij}, \alpha_j, b_{ij}, b_j, w_j \in \mathbb{R}$ and $N \in \mathbb{N}$ are dense in $C(I_n)$.

In other words, given a $g \in C(I_n)$ and given $\epsilon > 0$ there is a sum $\psi(x)$ of the form (A.5) such that

$$|\psi(x) - g(x)| < \epsilon \quad \forall x \in I_n.$$

Proof. Since σ is a continuous function, it follows that the set U of functions of the form (A.5) with $\alpha_j, w_{ij}, b_{ij}, w_j, b_j \in \mathbb{R}$ and $N \in \mathbb{N}$ is a linear subspace of $C(I_n)$. We claim that the closure of U is all of $C(I_n)$.

Assume that U is not dense in $C(I_n)$, let S be the closure of U , $S \neq C(I_n)$. By the Hahn-Banach theorem (see p. 104 of [244]) there is a bounded linear functional on $C(I_n)$, call it L , with the property that $L \neq 0$ but $L(S) = L(U) = 0$.

By the Riesz Representation Theorem (see p. 40 of [244]), the bounded linear functional L , is of the form

$$L(f) = \int_{I_n} f(x) d\mu$$

for some signed regular Borel measures μ such that $\mu(K) < \infty$ for every compact set $K \subset I_n$ (i.e. μ is a Radon measure). Hence,

$$\int_{I_n} h(x) d\mu = 0, \forall h \in U. \quad (\text{A.6})$$

We shall prove that (A.6) implies $\mu = 0$, which contradicts the hypothesis $L \neq 0$.

Using the definition of U , equation (A.6) can also be written as

$$\sum_{j=1}^N \alpha_j \int_{I_n} \sigma(w_j (\sum_{i=1}^n \sigma(w_{ij}(x_i - b_{ij})) - b_j)) d\mu = 0,$$

for any choice of $\alpha_j, w_{ij}, w_j, b_{ij}, b_j \in \mathbb{R}$ and $N \in \mathbb{N}$.

Note that for any $w, x, b \in \mathbb{R}$ we have that the continuous functions

$$\sigma_\lambda(w(x - b)) = \sigma(\lambda w(x - b) + \phi)$$

converge pointwise to the unit step function as λ goes to infinity, i.e.

$$\lim_{\lambda \rightarrow \infty} \sigma_\lambda(w(x - b)) = \gamma(w(x - b))$$

with

$$\gamma(y) = \begin{cases} 1 & \text{if } y > 0 \\ \sigma(\phi) & \text{if } y = 0 \\ 0 & \text{if } y < 0 \end{cases}$$

By hypothesis is true that for all λ_1, λ_2 in \mathbb{R}

$$\int_{I_n} \sigma_{\lambda_2}(w_j(\sum_{i=1}^n \sigma_{\lambda_1}(w_{ij}(x_i - b_{ij})) - b_j)) d\mu = 0.$$

It follows that for all λ_2 :

$$\lim_{\lambda_1 \rightarrow \infty} \int_{I_n} \sigma_{\lambda_2}(w_j(\sum_{i=1}^n \sigma_{\lambda_1}(w_{ij}(x_i - b_{ij})) - b_j)) d\mu = 0.$$

Now applying the Dominated Convergence Theorem (see Theorem 11.32 p 321 of [243]) and the fact that σ is continuous:

$$\begin{aligned} \int_{I_n} \lim_{\lambda_1 \rightarrow \infty} \sigma_{\lambda_2}(w_j(\sum_{i=1}^n \sigma_{\lambda_1}(w_{ij}(x_i - b_{ij})) - b_j)) d\mu = \\ \int_{I_n} \sigma_{\lambda_2}(w_j(\sum_{i=1}^n \gamma(w_{ij}(x_i - b_{ij})) - b_j)) d\mu. \end{aligned}$$

Again, by Dominated Convergence Theorem we have:

$$\begin{aligned} \lim_{\lambda_2 \rightarrow \infty} \int_{I_n} \sigma_{\lambda_2}(w_j(\sum_{i=1}^n \gamma(w_{ij}(x_i - b_{ij})) - b_j)) d\mu = \\ \int_{I_n} \gamma(w_j(\sum_{i=1}^n \gamma(w_{ij}(x_i - b_{ij})) - b_j)) d\mu. \end{aligned}$$

Hence we have obtained that $\forall \alpha_j, w_{ij}, b_{ij}, w_j, b_j \in \mathbb{R}$ and $\forall N \in \mathbb{N}$

$$\int_{I_n} \sum_{j=1}^N \alpha_j \gamma(w_j(\sum_{i=1}^n \gamma(w_{ij}(x_i - b_{ij})) - b_j)) d\mu = 0.$$

The function γ is very similar to the Heaviside function H , the only difference is that $H(0) = 1$ while $\gamma(0) = \sigma(\phi)$. Let R_i denote an open rectangle, $\partial_a R_i$ its left boundary (i.e. the boundary of a left-closed right-open rectangle) and $\partial_b R_i$ its right boundary (i.e. the boundary of a right-closed left-open rectangle). Repeating the construction seen in Lemma 5.2 to obtain rectangles, with the difference that here γ takes value $\sigma(\phi)$ on the boundaries, we get that

$$\sigma(\phi)\mu(\partial_a R_i) + (1 - \sigma(\phi))\mu(\partial_b R_i) + \mu(R_i) = 0$$

for every open rectangle R_i . Taking $\phi \rightarrow \infty$, implies

$$\mu(\partial_a R_i) + \mu(R_i) = 0 \quad \forall \text{ open rectangle } R_i.$$

Every open subset A of I_n , can be written as a countable union of disjoint partly open cubes (see Theorem 1.11 p.8 of [319]). Thus, from the fact that the measure is σ -additive we have that for every open subset A of I_n , $\mu(A) = 0$. Furthermore $\mu(I_n) = 0$. To obtain I_n from

$$\sum_{j=1}^N \alpha_j \gamma(w_j (\sum_{i=1}^n \gamma(w_{ij}(x_i - b_{ij})) - b_j))$$

it is sufficient to choose the parameters so that $w_{ij}(x_i - b_{ij}) > 0 \forall x_i \in [0, 1]$ and so that w_j, b_j maintains the condition on the input.

Hence, $\mu(A^C) = \mu(I_n) - \mu(A) = 0$. It follows that for all compact set K of I_n , $\mu(K) = 0$.

From the regularity of the measure, it follows that μ is the null measure. □

tanh-prod IAN

Theorem 5.6. *The finite sums of the form*

$$\begin{aligned} \psi(x) &= \sum_{j=1}^N \frac{\alpha_j}{2} \left[\prod_{l=1}^{M_j} \tanh(w_{jl}(z_j(x) - b_{jl})) + 1 \right] \\ z_j(x) &= \sum_{i=1}^n \frac{1}{2} \left[\prod_{k=1}^{m_i} \tanh(w_{ijk}(x_i - b_{ijk})) + 1 \right] \end{aligned} \tag{A.7}$$

with $w_{jl}, w_{ijk}, \alpha_j, b_{jl}, b_{ijk} \in \mathbb{R}$ and $M_j, N, m_i \in \mathbb{N}$, are dense in $C(I_n)$.

In other words given $g \in C(I_n)$ and given $\epsilon > 0$ there is a sum $\psi(x)$ defined as above such that

$$|\psi(x) - g(x)| < \epsilon \quad \forall x \in I_n.$$

Since \tanh is a continuous function, it follows that the family of functions defined by equation (A.7) is a linear subspace of $C(I_n)$. To prove that it is dense in $C(I_n)$ we will use the same argument we used for the continuous sigmoidal functions.

This is, called U the set of functions of the form (A.7), we assume that U is not dense in $C(I_n)$. Thus, by the Hahn-Banach theorem there exists a not null bounded linear functional on $C(I_n)$ with the property that it is zero on the closure of U . By the Riesz Representation Theorem, the bounded linear functional can be represented by a Radon measures. Then using the definition of U we will see that this measure must be the zero measure, hence the functional associated with it is null contradicting the hypothesis.

We define

$$h_\lambda(x) = \frac{1}{2} \left[\prod_{k=1}^m \tanh(\lambda(w_k(x - b_k)) + \phi) + 1 \right]. \tag{A.8}$$

To proceed with the proof as in the case of the proof for continuous sigmoidal functions, we need only to understand to what converges the function

$$\psi_{\lambda_2, \lambda_1}(x) = \sum_{j=1}^N \frac{\alpha_j}{2} h_{j\lambda_2} \left(\sum_{i=1}^n h_{i\lambda_1}(x) \right) \tag{A.9}$$

when λ_1 and λ_2 tend to infinity, and $h_{i\lambda}$ indicates the processing function related to input i .

Once we have shown that for some choice of the parameters they converge pointwise to step function we can use the same argument we used in the proof of Theorem 5.5.

The first step is therefore to study the limit of equation (A.9). Let us focus of the multiplication of tanh in the first layer, given by equation (A.8).

The pointwise limit of $h_\lambda(x)$ for $\lambda \rightarrow \infty$ depends on the sign of the limit of the product of tanh, that in turn depends on the sign of $w_k(x - b_k)$ for $k \in \{1, \dots, m\}$.

Remark 4. *We remark that for $x \in [0, 1]$, from the limit of equation (A.8) we can obtain the indicator functions of set of the form $x > b$ or $x < b$ for any $b \in \mathbb{R}$. We just have to choose the parameters in such a way that only one of the tanh in the multiplication is relevant. Let us define $Z = \{k \in \{1, \dots, m\} : w_k(x - b_k) > 0 \quad \forall x \in [0, 1]\}$. If $|Z| = m - 1$, i.e. there is only one $i \in \{1, \dots, m\}$ so that its weight are significant it holds that*

$$\lim_{\lambda \rightarrow \infty} h_\lambda(x) = v(x) = \begin{cases} 1 & \text{if } w_i(x - b_i) > 0 \\ \sigma(2\phi) & \text{if } w_i(x - b_i) = 0 \\ 0 & \text{if } w_i(x - b_i) < 0 \end{cases}$$

taking into account that $\sigma(2\phi) = \frac{1}{2} (\tanh(\phi) + 1)$.

Proof of Theorem 5.6. Considering Remark 4, the proof of Theorem 5.6 is analogous to that of Theorem 5.5. □

1.0.2 Experimental settings

All code was written in Python Programming Language. In particular, the following libraries were used for the algorithms: tensorflow for neural networks, scikit-learn for Logistic Regression, Decision Trees and Gradient Boosting Decision Trees.

A small exploration was made to determine the best structure of the neural network for each dataset. We used a breadth-first search algorithm defined as follows. We started with a network with just one neuron, we trained it and evaluated its performance. At each step, we can double the number of neurons in each layer except the output one or increase the depth of the network by adding a layer with one neuron. For each new configuration, we build a new structure based on it, initialize it and train it. If the difference between the accuracy achieved by the new structure and that of the previous step is lower than 1%, then a patience parameter is reduced by 1. The patience parameter is initialized as 5 and is passed down from a parent node to its spawned children, so that each node has its own instance of it. When patience reach 0, that configuration will not spawn new ones.

Before the neural network initialization, a random seed was set in order to reproduce the same results. As for the initialization of IAN, the weights w are initialised using the glorot uniform. For the biases b of the first layer a uniform between the minimum and the maximum of each feature was used, while for the following layers a uniform between the minimum and the maximum possible output from the neurons of the previous layer was used.

For the network training, Adam with a learning rate equal to 0.1 was used as optimization algorithm. The loss used is the binary or categorical crossentropy, depending on the number of classes in the dataset. In the calculation of the loss, the weight of each class is also taken into account, which is inversely proportional to the number of samples of that class in the training set. The maximum number of epochs for training has been fixed at 10000. To stop the training, an early stopping method was used based on the loss calculated on the train. The patience of early stopping is 250 epochs, with the variation that in these epochs the loss must decrease by at least 0.01. Not using a validation dataset may have led to overfitting of some structures, so in future work we may evaluate the performance when using early stopping based on a validation loss. The batch size was fixed at 128 and the training was performed on CPU or GPU depending on which was faster considering the amount of data. The Heaviside was trained as if its derivative was the same as the sigmoid.

For Decision Trees (DT) and Gradient Boosting Decision Trees (GBDT), an optimisation of the hyperparameters was carried out, in particular for `min_samples_split` (between 2 and 40) and `min_samples_leaf` (between 1 and 20). For GBDT, 1000 estimators were used, while for DT the `class_weight` parameter was set. For the rest of the parameters, we kept the default values of the python sklearn library.

1.0.3 Datasets

19 out of 23 datasets are publicly available, either on the UCI Machine Learning Repository website or on the Kaggle website. Here we present a full list of the datasets used, correlated with their shortened and full-length name, and the corresponding webpage where the description and data can be found.

Short name	Full-length name	Webpage
adult	Adult	<UCI_MLR_URL>/adult
australian	Statlog (Australian Credit Approval)	<UCI_MLR_URL>/statlog+(australian+credit+approval)
b-c-w	Breast Cancer Wisconsin	<UCI_MLR_URL>/Breast+Cancer+Wisconsin+(Diagnostic)
car	Car Evaluation	<UCI_MLR_URL>/car+evaluation
cleveland	Heart Disease	<UCI_MLR_URL>/heart+disease
crx	Credit Approval	<UCI_MLR_URL>/credit+approval
diabetes	Diabetes	https://www.kaggle.com/uciml/pima-indians-diabetes-database
german	Statlog (German Credit Data)	<UCI_MLR_URL>/statlog+(german+credit+data)
glass	Glass Identification	<UCI_MLR_URL>/glass+identification
haberman	Haberman's Survival	<UCI_MLR_URL>/haberman%27s+survival
heart	Statlog (Heart)	<UCI_MLR_URL>/statlog+(heart)
hepatitis	Hepatitis	<UCI_MLR_URL>/hepatitis
image	Statlog (Image Segmentation)	<UCI_MLR_URL>/Statlog+(Image+Segmentation)
ionosphere	Ionosphere	<UCI_MLR_URL>/ionosphere
iris	Iris	<UCI_MLR_URL>/iris
monks-1	MONK's Problems	<UCI_MLR_URL>/MONK%27s+Problems
monks-2	MONK's Problems	<UCI_MLR_URL>/MONK%27s+Problems
monks-3	MONK's Problems	<UCI_MLR_URL>/MONK%27s+Problems
sonar	Connectionist Bench	<UCI_MLR_URL>/Connectionist+Bench+(Sonar,+Mines+vs.+Rocks)

Table A.1: Publicly available datasets, with the short name used in in our work, their full-length name and the webpage where data and description can be found. The UCI_MLR_URL is the following: <https://archive.ics.uci.edu/ml/datasets/>

The 4 synthetic datasets of our own creation are composed of 1000 samples with 2 variables generated as random uniforms between -1 and 1 and an equation dividing the space into 2 classes.

The 4 equations used are:

- bisector: $x_1 > x_2$
- xor: $x_1 > 0 \wedge x_2 > 0$
- parabola: $x_2 < 2x_1^2 - \frac{1}{2}$
- circle $x_1^2 + x_2^2 < \frac{1}{2}$

These datasets are also represented in Figure A.3.

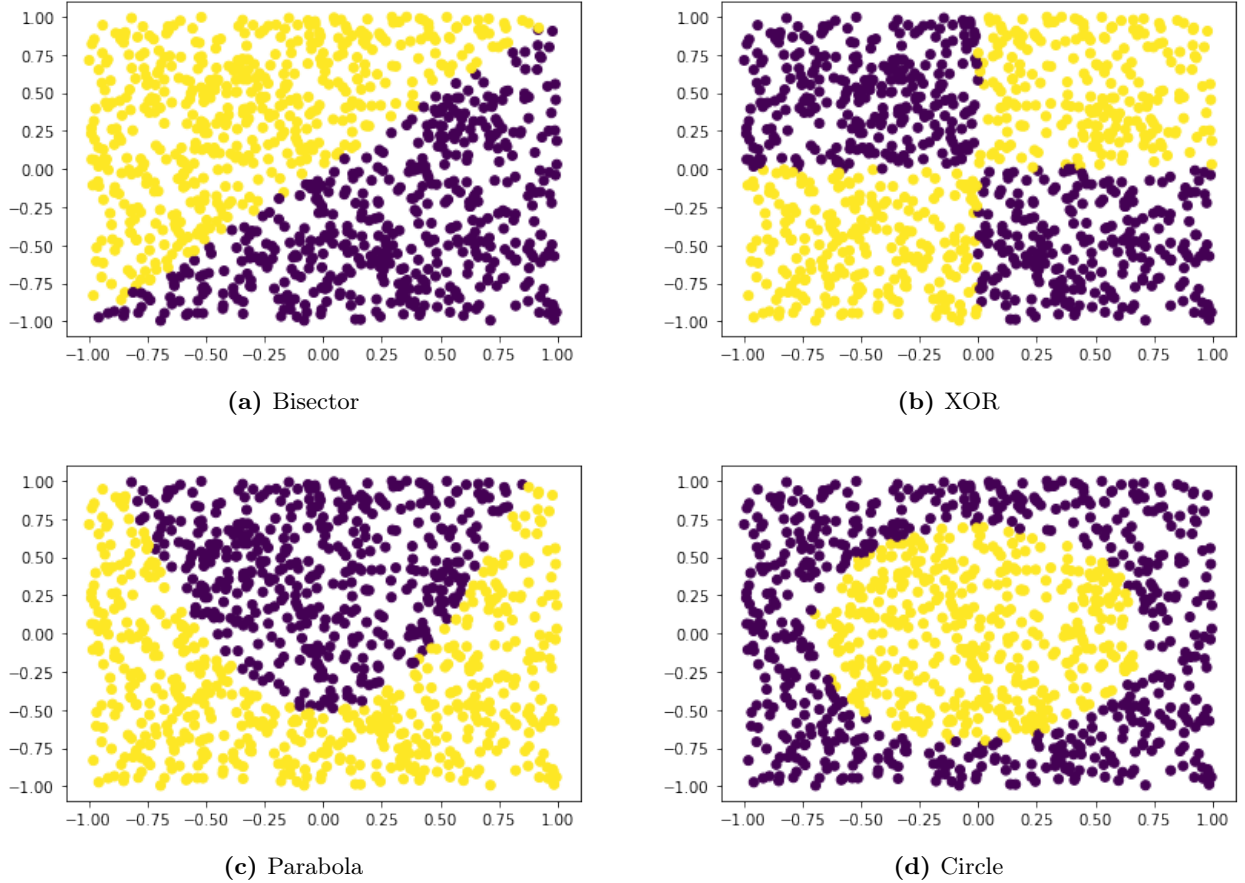


Figure A.3: The synthetically generated datasets we used to assess the soundness of our methodology.

1.0.4 Examples

Heart dataset - Heaviside IAN

The Statlog Heart dataset is composed of 270 samples and 13 variables of medical relevance. The dependent variable is whether or not the patient suffers from heart disease. In Figure A.4 you can find the network based on Heaviside IAN trained on the heart dataset. Only the inputs with a relevant contribution to the output are shown. From now on, we will indicate with $R_{k,j,i}$ the rule related to the processing function corresponding to the i -th input, of the j -th neuron, of the k -th layer. From the first neuron of the first layer we can easily retrieve the following rules: $R_{1,1,1} = x_1 \leq 54.29$, $R_{1,1,3} = x_3 \leq 3.44$, $R_{1,1,4} = x_4 \leq 123.99$, $R_{1,1,5} = x_5 \geq 369,01$, $R_{1,1,9} = x_9 \leq$

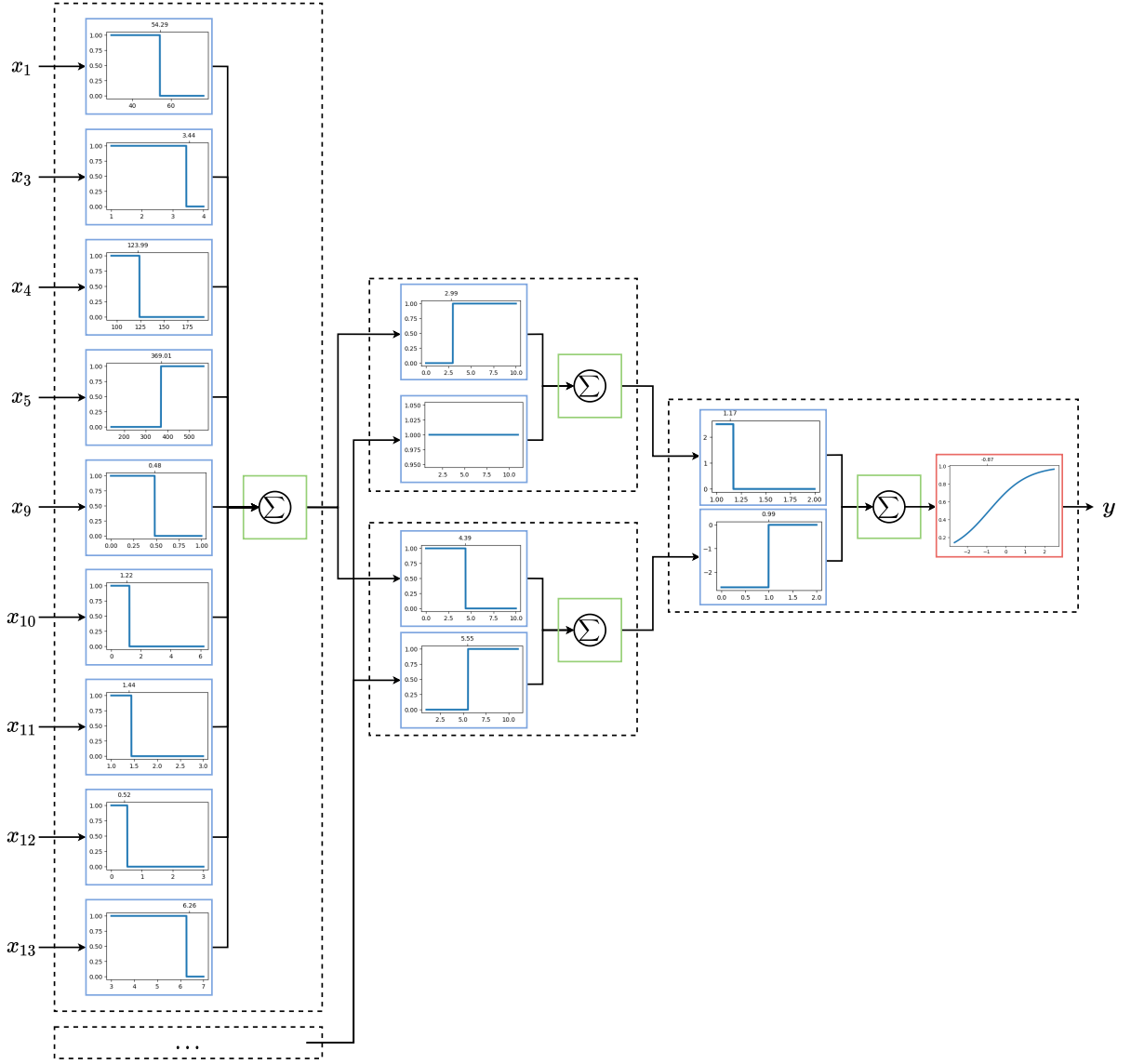


Figure A.4: The Heaviside IAN Network trained on the heart dataset. The Figure follows the color convention used for NEWRON.

$0.48, R_{1,1,10} = x_{10} \leq 1.22, R_{1,1,11} = x_{11} \leq 1.44, R_{1,1,12} = x_{12} \leq 0.52, R_{1,1,13} = x_{13} \leq 6.26$. The second neuron of the first layer is not shown for lack of space, but its obtained rules are $R_{1,2,2} = x_2 \geq 0.79, R_{1,2,3} = x_3 \geq 3.59, R_{1,2,4} = x_4 \geq 99.95, R_{1,2,5} = x_5 \geq 253.97, R_{1,2,8} = x_8 \leq 97.48, R_{1,2,9} = x_9 \leq 0.04, R_{1,2,10} = x_{10} \geq 2.56, R_{1,2,11} = x_{11} \geq 1.53, R_{1,2,12} = x_{12} \geq 0.52, R_{1,2,13} = x_{13} \geq 5.47$. Moreover, input x_7 gives always 1, so this must be taken into consideration in the next layer.

Moving on to the second layer, we can see in the first neuron that the second input is irrelevant, since the Heaviside is constant. The first processing function activates if it receives an input that is greater or equal to 2.99. Given that the input can only be an integer, we need at least 3 of the rules obtained for the first neuron of the first layer to be true: $R_{2,1,1} = 3 - of - \{R_{1,1,i}\}$. Following the same line of reasoning, in the second neuron of the second layer we see that we get $R_{2,2,1} = 5 - of - \{\neg R_{1,1,i}\}$ and $R_{2,2,2} = 5 - of - \{R_{1,2,i}\}$ (5 and not 6 because of x_7 processing function).

In the last layer, the first processing function has an activation of around 2.5 if it receives

an input that's less than 1.17. This can happen only if $R_{2,1,1}$ does not activate, so we can say: $R_{3,1,1} = \neg R_{2,1,1} = 7 - of - \{\neg R_{1,1,i}\}$. The second processing function gives a value of around -2.5 only if it gets an input less than 0.99, so only if the second neuron of the second layer does not activate. This means that $R_{2,2,1}$ and $R_{2,2,2}$ must be both false at the same time, so we get $R_{3,1,2} = \neg R_{2,2,1} \wedge \neg R_{2,2,2} = 5 - of - \{R_{1,1,i}\} \wedge 6 - of - \{\neg R_{1,2,i}\}$. Now there are 4 cases for the sum, i.e. the combinations of the 2 activations: $\{0+0, 2.5+0, 0-2.5, 2.5-2.5\} = \{-2.5, 0, 2.5\}$. Given that both have around the same value for the α parameter, the set is reduced to two cases. Looking at the processing function, we can see that is increasing with respect to the input, so since α_1 is positive, we can say that rule $R_{3,1,1}$ is correlated to class 1, while rule $R_{3,1,2}$, having a negative α_2 , has an opposite correlation. Looking at its values, we can see that for both 0 and 2.5 inputs, the activation function gives an output greater than 0.5. If we consider this as a threshold, we can say that only for an input of -2.5 we get class 0 as prediction. This happens only if $R_{3,1,2}$ is true and $R_{3,1,1}$ is false. Summarizing we get $R_0 = R_{3,1,2} \wedge \neg R_{3,1,1} = 5 - of - \{R_{1,1,i}\} \wedge 6 - of - \{\neg R_{1,2,i}\} \wedge 3 - of - \{R_{1,1,i}\} = 5 - of - \{R_{1,1,i}\} \wedge 6 - of - \{\neg R_{1,2,i}\}$, so that we can say "if R_0 then predicted class is 0, otherwise is 1".

Although we are not competent to analyse the above results from a medical perspective, it is interesting to note for example that the variables x_1 and x_4 , representing age and resting blood pressure respectively, are positively correlated with the presence of a heart problem.

Xor - sigmoid IAN

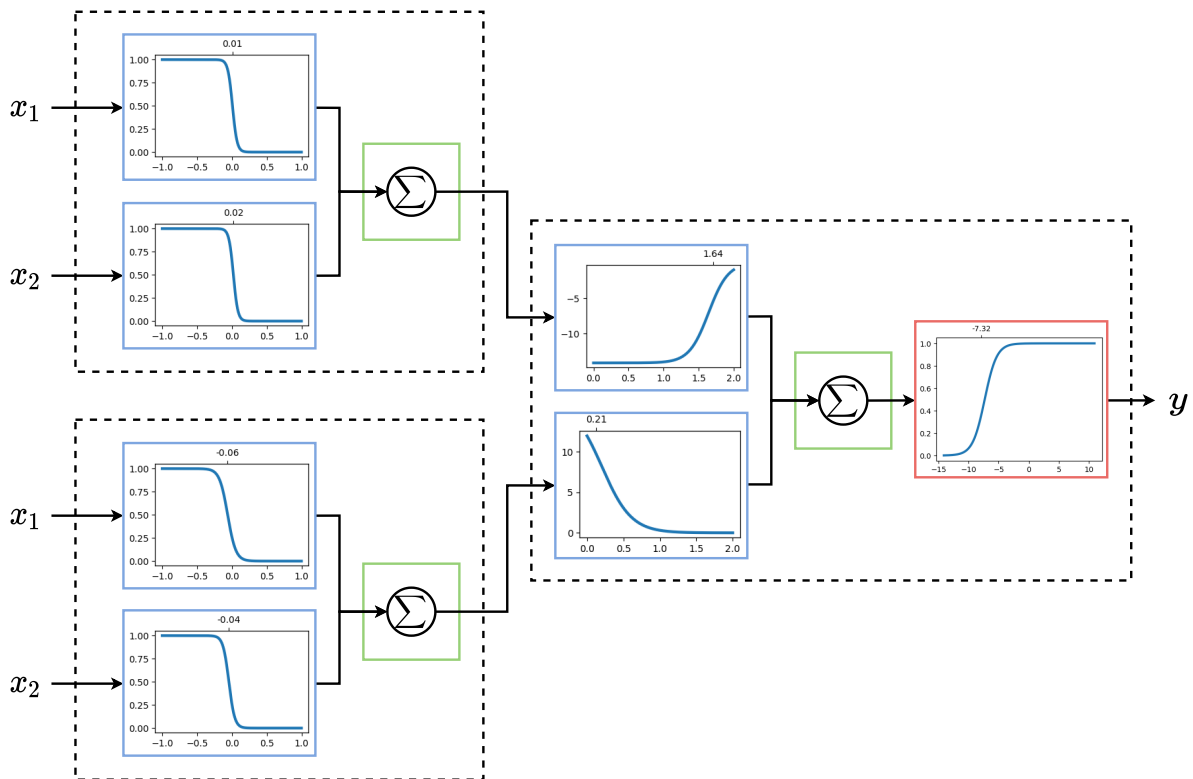


Figure A.5: The sigmoid IAN Network trained on the xor dataset. The Figure follows the color convention used for NEWRON.

Our custom xor dataset divides the 2D plane in quadrants, with the opposites having the same

label.

The network based on sigmoid IAN trained on xor dataset is represented in Figure A.5. As we can see, all the processing functions of the first layer converged to nearly the same shape: a steep inverted sigmoid centered in 0. Therefore, we can say the rules obtained are $R_{1,1,1} = R_{1,2,1} = x_1 \leq 0$ and $R_{1,1,2} = R_{1,2,2} = x_2 \leq 0$. In the last layer, the first processing function has a value of about -15 for inputs in $[0, 1]$, then it starts growing slowly to reach almost 0 for an input of 2. This tells us that it doesn't have an activation if both rules of the first neuron are true, so if $x_1 \leq 0 \wedge x_2 \leq 0$. On the other hand, the second processing function has no activation if its input greater than 1, that happens for example if we have a clear activation from at least one of the inputs in the second neuron of the first layer. So looking at it the opposite way, we need both those rules to be false ($x_1 > 0 \wedge x_2 > 0$) to have an activation of 12.5. The activation function is increasing with respect to the input, and to get a clear class 1 prediction, we need the input to be at least -5 . Considering if the processing functions could give only $\{-15, 0\}$ and $\{12.5, 0\}$ values, just in the case we got -15 from the first one and 0 from the second one it would give us a clear class 0 prediction. This happens only if $\neg(x_1 \leq 0 \wedge x_2 \leq 0) = x_1 > 0 \vee x_2 > 0$ and $\neg(x_1 > 0 \wedge x_2 > 0) = x_1 \leq 0 \vee x_2 \leq 0$, that can be summarised $(x_1 > 0 \vee x_2 > 0) \wedge (x_1 \leq 0 \vee x_2 \leq 0) = (x_1 > 0 \wedge x_2 \leq 0) \wedge (x_1 \leq 0 \vee x_2 > 0)$. Since this rule describes the opposite to xor, for class 1 we get the exclusive or logical operation.

Iris dataset - tanh-prod IAN

A dataset widely used as a benchmark in the field of machine learning is the Iris dataset. This contains 150 samples, divided into 3 classes (setosa, versicolor and virginica) each representing a type of plant, while the 4 attributes represent in order sepal length and width and petal length and width.

In Figure A.6 you can see the final composition of the network generated with the tanh-prod2 IAN neuron.

Considering the first neuron of the first layer, we see that it generates the following fuzzy rules: $R_{1,1,2} = x_2 > 3.08$ (sepal width), $R_{1,1,3} = x_3 < 5.14$ (petal length) and $R_{1,1,4} = x_4 < 1.74$ (petal width). For the first attribute (sepal length) it does not generate a clear rule, but forms a bell shape, reaching a maximum of 0.5. This tells us that x_1 is less relevant than the other attributes, since, unlike the other processing functions, it does not reach 1. The second neuron has an inverse linear activation for the first attribute, starting at 0.7 and reaching almost 0. The second attribute also has a peculiar activation, with an inverse bell around 2.8 and a minimum value of 0.4. The third and fourth attributes have clearer activations, such as $R_{1,2,3} = x_3 < 2.51$ and $R_{1,2,4} = x_4 < 1.45$.

The fact that petal length and width are the ones with the clearest activations and with those specific thresholds are in line with what has previously been identified on the Iris dataset by other algorithms.

We denote by $y_{k,j}$ the output of the j -th neuron of the k -th layer. Moving on to the second layer, the first neuron generates the rules "if $y_{1,1} < 1.83$ " and "if $y_{1,2} < 2.66$ ", while the second one generates "if $y_{2,1} > 2.08$ " and "if $y_{2,2} > 2.22$ ". Combined with what we know about the previous layer, we can deduce the following: $y_{1,1}$ is less than 1.83 only if the sum of the input activation functions is less than 1.83, which only happens if no more than one of the last three rules is activated ($0 + 1 + 0 < 1.83$), while the first one, even taking its maximum value, is discriminative only when the input of one of the other rules is close to the decision threshold ($0.5 + 1 + 0 + 0 < 1.83$, while

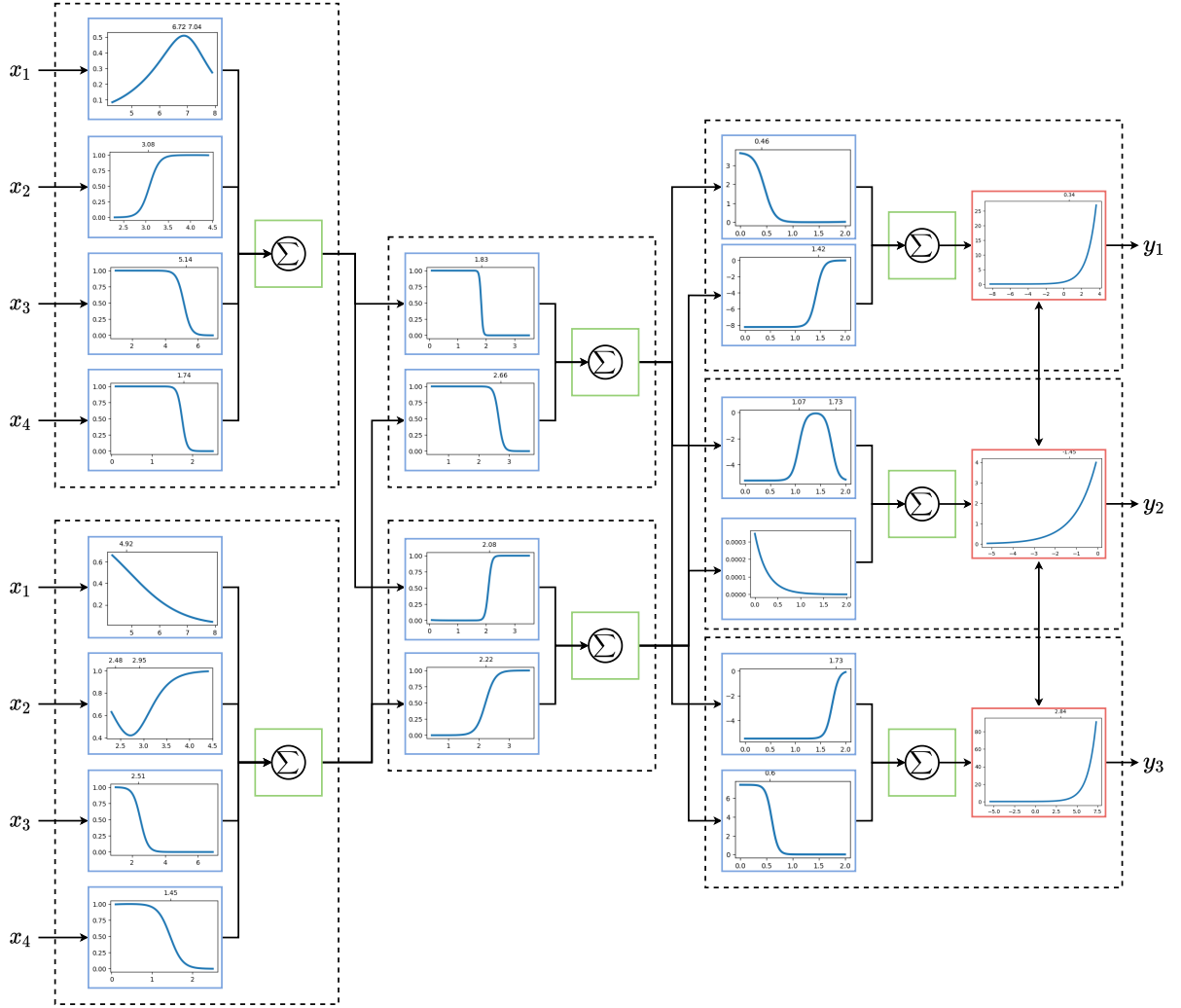


Figure A.6: The tanh-prod IAN Network trained on the iris dataset. The Figure follows the color convention used for NEWRON.

$0.5 + 1 + 0.5 + 0 > 1.83$). For $y_{1,2} < 2.66$, there are more cases. We can divide the second processing function of the second neuron of the first layer in two intervals: one for which $x_2 < 3.2$ and the other when $x_2 \geq 3.2$. In the first interval, the processing function gives a value that is less than 0.66, greater in the second one. With this, we can say that $y_{1,2} < 2.66$ even if $R_{1,2,3}$ and $R_{1,2,4}$ activates, if $x_2 < 3.2$ and x_1 is near its maximum.

In the second neuron of the second layer, the first processing function is nearly the exact opposite to that of the other neuron; we need at least two of $R_{1,1,2}$, $R_{1,1,3}$ or $R_{1,1,4}$ to be true, while $R_{1,1,1}$ still doesn't have much effect. The second processing function gives us $y_{1,2} > 2.22$. Considering that the minimum for the processing function related to x_2 is 0.4, we may need both rules $R_{1,2,3}$ and $R_{1,2,4}$ to be true to exceed the threshold, or just one of them active and x_1 to take on a low value and x_2 to be a high value.

For the last layer, remember that in this case since there are more than 2 classes, a softmax function is used to calculate the output probability, hence the arrows in the figure that join the layers of the last layer.

For the first output neuron, in order to obtain a clear activation, we need the first input to be less than 0.46 and the second greater than 1.42. This is because the α_i are 3 and -8 , and the

output activation function starts to have an activation for values greater than -2 . This means that the first neuron of the second layer should hardly activate at all, while the other should activate almost completely. Considering the thresholds for $y_{1,1}$ and $y_{1,2}$, we need the first to be greater than 2.08 and the other to be greater than 2.66. So $R_{3,1,1} = 2 - of - \{x_2 > 3.08, x_3 < 5.14, x_4 < 1.74\}$. For $R_{3,1,2}$ is more tricky to get a clear decision rule, but we can say that we may need both $R_{1,2,3}$ and $R_{1,2,4}$ to be true and $x_2 \geq 3.2$. If $x_2 < 3.2$, we need x_1 to not be near its maximum value. If just one of those two rules is true, we need $x_2 < 3.2$ and x_1 near 4, or $x_2 > 3.2$ but with a (nearly) direct correlation with x_1 , such that the more x_1 increases, the same does x_2 .

In the second output neuron, the second processing function is negligible, while the first one forms a bell shape between 1 and 2. This means that it basically captures when $y_{2,1}$ has a value of approximately 1.5, so when the decision is not clear. This is what gives this neuron maximum activation.

In the third and last output layer, since the first processing function has a negative α parameter and the activation function is increasing with respect to the input, we want it to output 0, and this requires maximum activation for the first neuron of the second layer. Regarding the second processing function, we want it to output 8, so we need nearly no activation from the second neuron of the second layer. So we need the first neuron of the first layer to output a value lower than 1.83 and the second neuron to output a value lower than 2.22. This means that no more than one rule $R_{1,1,i}$ needs to be active and at most two rules of $R_{1,2,i}$ need to be true.

We can conclude by saying that both neurons of the first layer are positively correlated with class 1, while they are negatively correlated with class 3. This means that low values of x_3 and x_4 , or high values of x_2 increase the probability of a sample to belong to class 1, while x_1 has almost no effect. For class 2, what we can say is that it correlates with a non-maximum activation of both neurons of the first layer, meaning that it captures those cases in which the prediction of one of the other classes is uncertain.

Appendix B

Explaining Neural Networks Using a Ruleset Based on Interpretable Concepts

2.0.1 Experimental Setup

Implementation

All code is written in Python 3 Programming Language. In particular, the following libraries are used for the algorithms: pytorch for neural networks, scikit-learn for Logistic Regression, Decision Trees and Gradient Boosted Decision Trees.

All the experiments have been run on a machine with this configuration: AMD EPYC 7373 Processor, 64GB RAM and NVIDIA GeForce RTX A4000 GPU.

Hyperparameters

Randomized search was used to determine the best set of hyperparameters for each classifier. Instead of using a predefined number of combinations, we used a fixed execution time, to make the comparison between classifiers more fair. For each dataset, for each classifier, the hyperparameters' search ran for 30 minutes. A list of all possible hyper parameter values for each algorithm follows:

IAN-LEN:

- number of concepts: 8, 16, 32, 64
- number of neurons: 8, 16, 32, 64
- number of layers: 1, 2
- IAN learning rate: 1.0e-2, 1.0e-3
- LEN learning rate: 1.0e-2, 1.0e-3

Neural Network:

- number of neurons: 1, 2, 4, 8, 16, 32, 64, 128
- number of layers: 1, 2, 3, 4, 5

- learning rate: 1.0e-2, 1.0e-3, 1.0e-4
- activation function: Tanh, LeakyReLU

Logistic Regression:

- penalty: l1, l2, elasticnet, none
- C: 1.0e+2, 5.0e+1, 1.0e+1, 5.0e+0, 1.0e+0, 5.0e-1, 1.0e-1, 5.0e-2, 1.0e-2
- solver: saga
- class_weight: balanced
- l1_ratio: 0.5

Decision Tree:

- criterion: gini, entropy
- splitter: best, random
- max_depth: 3, 5, 10, 20, 50, 100
- min_samples_split: 2, 0.01, 0.05, 0.1
- min_samples_leaf: 1, 0.01, 0.05, 0.1
- max_features: 1.0, 0.5, sqrt, log2
- class_weight: balanced

Gradient Boosted Decision Trees:

- learning_rate: 5.0e-1, 1.0e-1, 5.0e-2, 1.0e-2, 1.0e-3
- n_estimators: 10, 50, 100
- subsample: 0.5, 0.75, 1.0
- min_samples_split: 2, 0.01, 0.05, 0.1
- min_samples_leaf: 1, 0.01, 0.05, 0.1
- max_depth: 3, 5, 10, 20, 50, 100
- max_features: 1.0, 0.5, sqrt, log2

For Logistic Regression, Decision Trees and Gradient Boosted Decision Trees, it was used the name of the parameters and values as described by the Python scikit-learn library.

For both IAN-LEN and Neural Network, the number of maximum epochs was set to 2000. Early stopping with a patience of 100 epochs was used, monitoring the validation loss. At training end, the best model according to validation loss was selected. Data was divided in batches of 256 samples. The optimizer used for training is Adam. After selecting the number of layers, the number of neurons is selected randomly individually per each layer. The loss used for neural networks is the categorical cross entropy with no regularisation term.

Short name	Full-length name	Webpage
adult	Adult	<UCI_URL>adult
australian	Statlog (Australian Credit Approval)	<UCI_URL>statlog+(australian+credit+approval)
b-c-w	Breast Cancer Wisconsin	<UCI_URL>Breast+Cancer+Wisconsin+(Diagnostic)
cleveland	Heart Disease	<UCI_URL>heart+disease
diabetes	Diabetes	https://www.kaggle.com/uciml/pima-indians-diabetes-database
eye	EEG Eye State Data Set	<UCI_URL>EEG+Eye+State
german	Statlog (German Credit Data)	<UCI_URL>statlog+(german+credit+data)
haberman	Haberman’s Survival	<UCI_URL>haberman%27s+survival
heart	Statlog (Heart)	<UCI_URL>statlog+(heart)
hepatitis	Hepatitis	<UCI_URL>hepatitis
ionosphere	Ionosphere	<UCI_URL>ionosphere
iris	Iris	<UCI_URL>iris
monks-1	MONK’s Problems	<UCI_URL>MONK%27s+Problems
monks-2	MONK’s Problems	<UCI_URL>MONK%27s+Problems
monks-3	MONK’s Problems	<UCI_URL>MONK%27s+Problems
poker	Poker Hand Data Set	<UCI_URL>Poker%2BHand
sonar	Connectionist Bench	<UCI_URL>Connectionist+Bench+(Sonar,+Mines+vs.+Rocks)
thyroid	Thyroid Disease Data Set	<UCI_URL>thyroid+disease

Table B.1: Publicly available datasets, with the short name used in in our work, their full-length name and the webpage where the data and the description can be found. The UCI_URL is the following: <https://archive.ics.uci.edu/ml/datasets/>

Randomization

Before the initialization of the neural network weights, a random seed was set to reproduce the same results. The 21094 random seed was used for splitting the dataset and for random initialization of the neural network weights. For each fold of the 5-fold cross-validation, the random seed was increased by 1.

2.0.2 Datasets

All 18 datasets are publicly available, 17 on the UCI Machine Learning Repository website and one on the Kaggle website. In table B.1 we present a full list of the datasets used, correlated with their shortened and full-length name, and the corresponding webpage where the description and the data can be found.

Appendix C

Noisy Labels - Supplementary Materials

Supplementary Materials

3.0.1 Inter annotator agreement. Symmetric noise and symmetric ground truth distribution

Cohen’s κ coefficient measures the agreement between two raters who each classify n items into C mutually exclusive categories.

We define the agreement among raters a and b as p_o : $p_o = \sum_{c=1}^C \mathbb{P}(y_a = c \cap y_b = c)$ Cohen and others [56] suggest comparing the actual agreement (p_o) with the “chance agreement” that could be obtained if the labels assigned by the two annotators were independent (we will denote this quantity by p_e).

$$p_e = \sum_{c=1}^C \mathbb{P}(y_a = c) \mathbb{P}(y_b = c) \quad (\text{C.1})$$

Cohen’s κ coefficient is defined as the difference between the true agreement and the “chance agreement” normalized by the maximum value this difference can reach

$$\kappa := \frac{p_o - p_e}{1 - p_e}, \quad (\text{C.2})$$

If the raters are in complete agreement then $\kappa = 1$. If there is no agreement among the raters other than what would be expected by chance (i.e. $p_o = p_e$) $\kappa = 0$. It can also take negative values. A negative κ indicates an agreement worse than that expected by chance. This can be interpreted as no agreement at all between annotators. In our work, we assume that the two raters are a corrupted version of an observable “clean” (ground truth) label. In this setting, the label assigned by annotator a to an item and the respective uncorrupted label are not independent random variables. We found that in this setting the κ coefficient can take only non-negative values.

3.0.2 On the hypothesis of commutativity in Lemma 5.1

In Lemma 5.1 we found how to compute T given M and D . To find this relationship we require that $D^{\frac{1}{2}}$ commutes with T . This hypothesis is satisfied when D and T have a particular structure,

namely

$$\frac{\sqrt{d_i}}{\sqrt{d_j}} t_{ij} = t_{ij} \quad \forall i \text{ and } j.$$

That is satisfied or if $d_i = d_j$ or if $t_{ij} = 0$, namely every class so that the probability of going from class i to class j (and vice-versa) is not zero is equiprobable.

So T has to be block diagonal, or better reducible by a permutation of the classes to a block diagonal matrix and D has to have all equal elements on indices relatives to the same block in T . For instance

$$T = \begin{pmatrix} T_1 & 0 & 0 & 0 & 0 \\ 0 & T_2 & 0 & 0 & 0 \\ 0 & 0 & T_3 & 0 & 0 \\ 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & T_j \end{pmatrix} \quad \text{and} \quad D = \begin{pmatrix} D_1 & 0 & 0 & 0 & 0 \\ 0 & D_2 & 0 & 0 & 0 \\ 0 & 0 & D_3 & 0 & 0 \\ 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & D_j \end{pmatrix}$$

with

$$D_i = \begin{pmatrix} d_i & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & d_i \end{pmatrix}$$

T need not be block diagonal but must be reconducted to a block diagonal matrix by permuting the classes, for instance in the following case, we can obtain a matrix block diagonal by permuting classes 2 and 4

$$T = \begin{pmatrix} t_{11} & 0 & 0 & t_{14} \\ 0 & t_{22} & t_{23} & 0 \\ 0 & t_{23} & t_{33} & 0 \\ t_{14} & 0 & 0 & t_{44} \end{pmatrix} \quad \text{and} \quad D = \begin{pmatrix} d_1 & 0 & 0 & 0 \\ 0 & d_2 & 0 & 0 \\ 0 & 0 & d_2 & 0 \\ 0 & 0 & 0 & d_1 \end{pmatrix}$$

Notice that T can be rewritten as follows permuting classes 2 and 4

$$T = \begin{pmatrix} t_{11} & t_{14} & 0 & 0 \\ t_{14} & t_{44} & 0 & 0 \\ 0 & 0 & t_{33} & t_{23} \\ 0 & 0 & t_{23} & t_{22} \end{pmatrix}$$

From the technical point of view, we have noticed that solving this equation is extremely complicated without making such assumptions. Another assumption we could have used, also required by [221] to solve the same problem, is requiring that the matrix $D^{\frac{1}{2}}T$ has diagonal Jordan decomposition. However, this assumption is more complicated to translate at the level of the structure of the matrices T and D .

From a practical point of view, making such an assumption means that there are classes that annotators can confuse with one another while they never swap between them, other classes. For example, if the problem is to classify images and the classes are “cat”, “lynx”, “bats”, “bird”, “cougar”; we can think that the annotators have a non-zero probability of confusing with each other the feline classes “lynx”, “cat”, “cougar”, while they have zero probability of assigning a picture of a lynx the label “bird”. Commutativity is guaranteed in the case of a uniform distribution over the classes.

There are many applications where we expect the distribution over the classes to be uniform and not to have any class with a higher probability. In general, we can fall back to an approximation of this case by reducing the samples.

3.0.3 Proofs

Proof of Lemma 5.1

Proof. From Equation 3.8 we get:

$$M = TDT = D^{\frac{1}{2}}TTD^{\frac{1}{2}} \rightarrow D^{-\frac{1}{2}}MD^{-\frac{1}{2}} = T^2 \quad (\text{C.3})$$

Note that T and $D^{\frac{1}{2}}MD^{\frac{1}{2}}$ are positive definite (because D and M are positive definite) and hence they have eigenvalue decompositions of the following form:

$$T = U_T \Lambda_T U_T^T \quad (\text{C.4})$$

$$D^{-\frac{1}{2}}MD^{-\frac{1}{2}} = U_M \Lambda_M U_M^T \quad (\text{C.5})$$

where U_x are orthogonal matrices and Λ_x are diagonal positive definite matrices. It then follows that:

$$T^2 \stackrel{(a)}{=} U_T \Lambda_T^2 U_T^T = U_M \Lambda_M U_M^T \quad (\text{C.6})$$

where in (a) we used the fact that U_T is orthogonal. Since $U_M \Lambda_M U_M^T$ is an eigenvalue decomposition of T^2 we conclude that:

$$T = U_M \Lambda_M^{\frac{1}{2}} U_M^T, \quad T^{-1} = U_M \Lambda_M^{-\frac{1}{2}} U_M^T \quad (\text{C.7})$$

□

Proof of Lemma 5.2: bounds error on the estimation of M

Proposition 5.0.1. Let $M_{a,b}$ be the agreement matrix for annotators a and b defined in Equation 3.7 and $\widehat{M}_{a,b}$ be the estimated agreement matrix defined in eq. Equation 3.11. For every $\epsilon > 0$ it holds that

$$\mathbb{P}^n(|(M_{a,b})_{ij} - (\widehat{M}_{a,b})_{ij}| < \epsilon) \geq 1 - 2e^{-2\epsilon^2 n}.$$

And

$$\mathbb{P}^n\left(\forall i, j \in \{1, C\}^2 |(M_{a,b})_{ij} - (\widehat{M}_{a,b})_{ij}| < \epsilon\right) \geq 1 - 2C^2 e^{-2\epsilon^2 n}.$$

where \mathbb{P}^n denotes the probability according to which the n training samples are distributed, i.e. we are assuming that the samples are independently drawn according to the probability \mathbb{P} .

To simplify the notation we will omit the dependency from the annotators in the matrices: $M = M_{a,b}$ and $\widehat{M} = \widehat{M}_{a,b}$ $M_{ij} = \mathbb{P}(y_a = i, y_b = j)$ and $\widehat{M}_{ij} = \frac{1}{n} \sum_{h=1}^n \mathbf{1}((y_a)_h = i, (y_b)_h = j)$.

Proof. To prove the claim we only need to apply Hoeffding's inequality to the random variables $X_h^{ij} = \mathbf{1}_{y_{a_h}=i, y_{b_h}=j}$. Indeed it holds that $0 \leq X_h^{ij} \leq 1$ and $\widehat{M}_{ij} = \frac{1}{n} \sum_{h=1}^n X_h^{ij}$, while $\mathbb{E}[X_h^{ij}] = M_{ij}$.

Notice that the random variables $X_1^{ij} \dots X_n^{ij}$ are independent since we assume samples to be independent with respect to each other and so it follows that $(x_h, y_{a_h}, y_{b_h}), (x_k, y_{a_k}, y_{b_k})$ are independent.rf

$$\mathbb{P}\left(\left|\mathbb{E}[X_h^{ij}] - \frac{1}{n} \sum_{h=1}^n X_h^{ij}\right| > \epsilon\right) \leq 2e^{-2\epsilon^2 n}. \quad (\text{C.8})$$

From the previous equation, using union bounds we can obtain that

$$\mathbb{P}\left(\forall (i, j) \in \{1, C\}^2 \left|\mathbb{E}[X_h^{ij}] - \frac{1}{n} \sum_{h=1}^n X_h^{ij}\right| < \epsilon\right) \geq 1 - 2C^2 e^{-2\epsilon^2 n}. \quad (\text{C.9})$$

Namely

$$\mathbb{P}\left(\forall (i, j) \in \{1, C\}^2 \left|M_{ij} - \widehat{M}_{ij}\right| < \epsilon\right) \geq 1 - 2C^2 e^{-2\epsilon^2 n}. \quad (\text{C.10})$$

□

Lemma 5.1. Let A be a matrix in $\mathbb{R}^{C \times C}$ so that it exists $\epsilon > 0$ for all i, j $|A_{ij}| \leq \epsilon$. For every $p \in [1, \infty]$, if $\|\cdot\|_p$ denotes the matrix norm induced by the p -vector norm,

$$\|A\|_p \leq C\epsilon.$$

Proof.

$$\|A\|_p := \sup_{x: \|x\|_p=1} \|Ax\|_p$$

Let x be a vector of p -norm 1. $(Ax)_i = \sum_{j=1}^C A_{ij}x_j$

$$\|Ax\|_p = \left(\sum_{i=1}^C \left|\sum_{j=1}^C A_{ij}x_j\right|^p\right)^{\frac{1}{p}} \leq \left(\sum_{i=1}^C \left(\sum_{j=1}^C |A_{ij}x_j|\right)^p\right)^{\frac{1}{p}} \leq \epsilon \left(\sum_{i=1}^C \left(\sum_{j=1}^C |x_j|\right)^p\right)^{\frac{1}{p}}$$

Now, denoting by $\mathbf{1}$ the vector with all ones, using Hölder inequality we can obtain :

$$\sum_{j=1}^C |x_j| = \|\mathbf{1}x\|_1 \leq \|x\|_p \|\mathbf{1}\|_{\frac{p}{p-1}} = \|x\|_p C^{\frac{p-1}{p}}$$

So

$$\|Ax\|_p \leq \epsilon \left(\sum_{i=1}^C \|x\|^p C^{p-1}\right)^{\frac{1}{p}} = \epsilon C \|x\|_p = \epsilon C$$

□

Proof Lemma 5.2. For the previous Lemma it holds that if all the elements of the matrix are less or equal than ϵ , the p norm is bounded by ϵC

So we can derive that

$$\mathbb{P}\left(\|M_{a,b} - \widehat{M}_{a,b}\|_p > \epsilon\right) \geq \mathbb{P}\left(\forall (i, j) \in \{1, C\}^2 \left|M_{ij} - \widehat{M}_{ij}\right| < \frac{\epsilon}{C}\right) \geq 1 - 2C^2 e^{-2\frac{\epsilon^2}{C^2} n}. \quad (\text{C.11})$$

□

Proof of Theorem 5.3: bound error on the estimation of T

We start by introducing the following helpful remark and Lemmas.

Remark 5. We defined $\hat{T} = \underset{B}{\operatorname{argmin}} \|B - \hat{U}\hat{\Lambda}_M^{\frac{1}{2}}\hat{U}^T\|_2^2$, with B that satisfies all the constraints in Equation 3.18. We know that the matrix T we want to approximate satisfies all the constraints in Equation 3.18, so by definition

$$\|\hat{T} - \hat{U}\hat{\Lambda}_M^{\frac{1}{2}}\hat{U}^T\|_2^2 \leq \|T - \hat{U}\hat{\Lambda}_M^{\frac{1}{2}}\hat{U}^T\|_2^2,$$

from which it follows that

$$\|T - \hat{T}\|_2^2 \leq 2\|T - \hat{U}\hat{\Lambda}_M^{\frac{1}{2}}\hat{U}^T\|_2^2$$

so any bound we will found for $\|T - \hat{U}\hat{\Lambda}_M^{\frac{1}{2}}\hat{U}^T\|_2^2$ holds also for \hat{T} estimated as in Equation 3.17 with a coefficient 2.

Lemma 5.2. Let A be a square, symmetric, positive definite matrix, in $\mathbb{R}^{C \times C}$ and let \sqrt{A} the unique positive definite symmetric, matrix so that $\sqrt{A}\sqrt{A} = A$ (On the existence of this matrix, see Theorem 7.2.6 at p. 439 in [127]). The bounded operator $F_{\sqrt{\cdot}} : \mathcal{S} \rightarrow \mathcal{S}$ defined as follow $F_{\sqrt{\cdot}} : A = \sqrt{A}$, where we denote by \mathcal{S} the space of symmetric positive definite matrix, is differentiable and it hold the following upper bound for the induced 2 norm of the derivative

$$\|D[\sqrt{A}]\|_2 \leq \frac{1}{2\sqrt{\lambda_{\min}(A)}} \|\operatorname{vec}(A)\|_2. \quad (\text{C.12})$$

Proof. Let us consider the vector space of square matrices $M_C(\mathbb{R})$ with the 2 norm and let $D[\sqrt{A}]$ denote the operator that is the derivative of $F_{\sqrt{\cdot}}$ in this space and $D[A]$ the derivative of A . From the fact that $\sqrt{A}\sqrt{A} = A$ it follows that

$$D[\sqrt{A}]\sqrt{A} + \sqrt{A}D[\sqrt{A}] = D[A]. \quad (\text{C.13})$$

Equation C.13 is a special case of Sylvester equation, and using that \sqrt{A} is symmetric can be rewritten as

$$(I_C \otimes \sqrt{A} + \sqrt{A} \otimes I_C)\operatorname{vec}(D[\sqrt{A}]) = \operatorname{vec}(D[A]). \quad (\text{C.14})$$

It follow that

$$\operatorname{vec}(D[\sqrt{A}]) = (I_C \otimes \sqrt{A} + \sqrt{A} \otimes I_C)^{-1}\operatorname{vec}(D[A]) = (I_C \otimes \sqrt{A} + \sqrt{A} \otimes I_C)^{-1}\operatorname{vec}(A).$$

Notice that the eigenvalues of the square root of a symmetric, positive def matrix are the square root of the eigenvalues of the original matrices. Indeed if A can be decomposed as $A = U\Lambda U^T$, with U orthogonal matrix, it holds that $\sqrt{A} = U\sqrt{\Lambda}U^T$. Now the eigenvalues of $\sqrt{A} \otimes I_C + I_C \otimes \sqrt{A}$ are $\sqrt{\lambda_i} + \sqrt{\lambda_j}$ with $1 \leq i, j \leq C$, with λ_i eigenvalue of A . The minimum eigenvalue of a symmetric positive def matrix B is the maximum eigenvalue of the inverse, indeed if $B = VDVT^T$, with V orthogonal, $B^{-1} = VD^{-1}V^T$. So the minimum eigenvalue of $\sqrt{A} \otimes I_C + I_C \otimes \sqrt{A}$, that is the

maximum eigenvalue of $(\sqrt{A} \otimes I_C + I_C \otimes \sqrt{A})^{-1}$ is $2\lambda_{\min}(\sqrt{A})$. It follows that

$$\begin{aligned} \|(I_C \otimes \sqrt{A} + \sqrt{A} \otimes I_C)^{-1}\|_2 &= \sqrt{\lambda_{\max}((I_C \otimes \sqrt{A} + \sqrt{A} \otimes I_C)^{-2})} \\ &= \sqrt{\lambda_{\min}((I_C \otimes \sqrt{A} + \sqrt{A} \otimes I_C)^2)} \\ &= \lambda_{\min}((I_C \otimes \sqrt{A} + \sqrt{A} \otimes I_C)) \\ &= \frac{1}{2\sqrt{\lambda_{\min}(A)}}. \end{aligned}$$

So $\|\text{vec}(D[\sqrt{A}])\|_2 \leq \frac{1}{2\sqrt{\lambda_{\min}(A)}} \|\text{vec}(A)\|_2$. $\|\text{vec}(A)\|_2^2 = \sum_{k=1}^{C^2} a_k^2$ for every vector x of norm 1 (this implies $x_i < 1$)

$$\|Ax\|_2^2 = \sum_{k=1}^C \sum_{i=1}^C a_{ki}^2 x_i^2 \leq \sum_{k=1}^C \sum_{i=1}^C a_{ki}^2 = \|\text{vec}(A)\|_2^2.$$

It follows that the induce 2 norm of the derivative $\|D[\sqrt{A}]\|_2 \leq \frac{1}{2\sqrt{\lambda_{\min}(A)}} \|\text{vec}(A)\|_2$ \square

Let T and \hat{T} be defined as in Equation C.7 and Equation 3.16.

The following Lemma holds for two general double stochastic matrices.

Lemma 5.3. Let T and \hat{T} be two symmetric, stochastic matrices, it holds that :

$$\|T - \hat{T}\|_2 \leq \frac{\sqrt{C}\|T^2 - \hat{T}^2\|}{\lambda_{\min}(T^2) - \|T^2 - \hat{T}^2\|_2} \quad \text{and} \quad \|T - \hat{T}\|_2 \leq \frac{\sqrt{C}\|T^2 - \hat{T}^2\|}{\lambda_{\min}(\hat{T}^2) - \|T^2 - \hat{T}^2\|_2} \quad (\text{C.15})$$

Proof. From the previous Lemma and the mean absolute value

$$\|\sqrt{A} - \sqrt{B}\|_2 \leq \|A - B\|_2 \sup_{0 \leq \theta \leq 1} \|D[\sqrt{\theta A + (1-\theta)B}]\|_2$$

For Weyl's inequality $\lambda_{\min}(\theta T^2 + (1-\theta)\hat{T}^2) \leq \lambda_{\min}(\theta T^2) + \lambda_{\min}((1-\theta)\hat{T}^2) = \theta\lambda_{\min}(T^2) + (1-\theta)\lambda_{\min}(\hat{T}^2)$.

$$\begin{aligned} \sup_{0 \leq \theta \leq 1} \|D\sqrt{\theta T^2 + (1-\theta)\hat{T}^2}\|_2 &\leq \frac{1}{2} \sup_{0 \leq \theta \leq 1} \frac{\|\text{vec}(\theta T^2) + (1-\theta)\hat{T}^2\|_2}{\theta\lambda_{\min}(T^2) + (1-\theta)\lambda_{\min}(\hat{T}^2)} \\ &\leq \frac{1}{2} \sup_{0 \leq \theta \leq 1} \frac{\theta\|\text{vec}(T^2)\|_2 + (1-\theta)\|\text{vec}(\hat{T}^2)\|_2}{\theta\lambda_{\min}(T^2) + (1-\theta)\lambda_{\min}(\hat{T}^2)} \\ &\leq \frac{1}{2} \sup_{0 \leq \theta \leq 1} \frac{\|\text{vec}(T^2)\|_2 + \|\text{vec}(\hat{T}^2)\|_2}{\theta\lambda_{\min}(T^2) + (1-\theta)\lambda_{\min}(\hat{T}^2)} \\ &\leq \sup_{0 \leq \theta \leq 1} \frac{\sqrt{C}}{\theta\lambda_{\min}(T^2) + (1-\theta)\lambda_{\min}(\hat{T}^2)} \end{aligned}$$

In the last inequality we used that T and \hat{T} are doubly stochastic so $\sum_{i=1}^C T_{ij}^2 \leq (\sum_{i=1}^C T_{ij})^2 = 1$. So $\|\text{vec}\|_2 = \left(\sum_{i=1}^C \sum_{j=1}^C T_{ij}^2\right)^{\frac{1}{2}} \leq \sqrt{C}$. Moreover deriving $\frac{1}{\theta\lambda_{\min}(T^2) + (1-\theta)\lambda_{\min}(\hat{T}^2)}$ with respect to θ we find that

$$\sup_{0 \leq \theta \leq 1} \frac{1}{\theta\lambda_{\min}(T^2) + (1-\theta)\lambda_{\min}(\hat{T}^2)} = \begin{cases} \frac{1}{\lambda_{\min}(T^2)} & \text{if } \lambda_{\min}(T^2) < \lambda_{\min}(\hat{T}^2) \\ \frac{1}{\lambda_{\min}(\hat{T}^2)} & \text{if } \lambda_{\min}(T^2) > \lambda_{\min}(\hat{T}^2) \end{cases}$$

$$\sup_{0 \leq \theta \leq 1} \frac{1}{\theta \lambda_{\min}(T^2) + (1 - \theta) \lambda_{\min}(\widehat{T}^2)} = \frac{1}{\min(\lambda_{\min}(\widehat{T}^2), \lambda_{\min}(T^2))}.$$

Now,

$$\min(a, b) = \begin{cases} a = b - |b - a| & \text{if } a < b \\ b & \text{if } b \leq a \end{cases} \quad (\text{C.16})$$

We notice that for symmetric matrices $\|A\|_2 = \sqrt{\lambda_{\max}(A)^2} = \sqrt{(\lambda_{\max}(A))^2} = |\lambda_{\max}(A)|$. So we can Since $T^2 - \widehat{T}^2$ is symmetric: $\|T^2 - \widehat{T}^2\|_2 = |\lambda_{\max}(T^2 - \widehat{T}^2)|$.

It follows that

$$\min(\lambda_{\min}(\widehat{T}^2), \lambda_{\min}(T^2)) \geq \lambda_{\min}(T^2) - |\lambda_{\min}(T^2) - \lambda_{\min}(\widehat{T}^2)| \quad (\text{C.17})$$

$$\geq \lambda_{\min}(T^2) - |\lambda_{\min}(T^2) - \lambda_{\min}(\widehat{T}^2)| \quad (\text{C.18})$$

$$\geq \lambda_{\min}(T^2) - |\lambda_{\min}(T^2 - \widehat{T}^2)| \quad (\text{C.19})$$

$$\geq \lambda_{\min}(T^2) - |\lambda_{\max}(T^2 - \widehat{T}^2)| \quad (\text{C.20})$$

$$= \lambda_{\min}(T^2) - \|T^2 - \widehat{T}^2\|_2. \quad (\text{C.21})$$

In the previous equations we use that $|\lambda_{\min}(T^2) - \lambda_{\min}(\widehat{T}^2)| \leq |\lambda_{\max}(T^2 - \widehat{T}^2)|$. We now prove that it is true. Suppose without loss of generality that $\lambda_{\min}(T^2) > \lambda_{\min}(\widehat{T}^2)$. If it is the case $\lambda_{\min}(T^2) - \lambda_{\min}(\widehat{T}^2) = \lambda_{\min}(T^2) + \lambda_{\max}(-\widehat{T}^2) \leq \lambda_{\max}(T^2 - \widehat{T}^2) \leq |\lambda_{\max}(T^2 - \widehat{T}^2)|$, where we used Weyl's inequality.

If the $\lambda_{\min}(T^2) > \lambda_{\min}(\widehat{T}^2)$ following the same path we obtain $|\lambda_{\min}(\widehat{T}^2) - \lambda_{\min}(T^2)| \leq |\lambda_{\max}(\widehat{T}^2 - T^2)|$.

it follow that $\lambda_{\min}(T^2) - \lambda_{\min}(\widehat{T}^2) < \|T^2 - \widehat{T}^2\|_2$

□

Proof Theorem Theorem 5.3. From Lemma 5.3 we know that

$$\|T - \widehat{T}\|_2 \leq \frac{\sqrt{C} \|T^2 - \widehat{T}^2\|}{\lambda_{\min}(T^2) - \|T^2 - \widehat{T}^2\|_2} \quad (\text{C.22})$$

Now, in general

$$\frac{\sqrt{C}x}{b-x} < \epsilon \text{ iff } x < b \frac{\epsilon}{\sqrt{C} + \epsilon}.$$

It follows that

$$\mathbb{P}(\|T - \widehat{T}\|_2 < \epsilon) = \mathbb{P}\left(\|T^2 - \widehat{T}^2\|_2 < \lambda_{\min}(T^2) \frac{\epsilon}{\sqrt{C} + \epsilon}\right)$$

or

$$\mathbb{P}(\|T - \widehat{T}\|_2 < \epsilon) = \mathbb{P}(\|T^2 - \widehat{T}^2\|_2 < \lambda_{\min}(\widehat{T}^2) \frac{\epsilon}{\sqrt{C} + \epsilon}) \geq \mathbb{P}(\|T^2 - \widehat{T}^2\|_2 < \frac{\lambda_{\min}(\widehat{T}^2)}{\sqrt{C} + 1} \epsilon)$$

Since we can assume $\epsilon \leq 1$ (if $n > \frac{C^2(\sqrt{C}+1)^2(\ln(2C^2))^2}{2\lambda_{\min}(\widehat{T})^2}$). Notice that we are interested in convergence properties of \widehat{T} , so we are interested in founding these bounds for small ϵ .

Now $T^2 - \widehat{T}^2 = D^{1/2}(M - \widehat{M})D^{1/2}$.

So $\|T^2 - \widehat{T}^2\|_2 \leq \|M - \widehat{M}\|_2 \|D^{1/2}\|_2^2 = \|M - \widehat{M}\|_2 \|D\|_2 = \|M - \widehat{M}\|_2 \lambda_{\max}(D)$. As a consequence :

$$\begin{aligned} \mathbb{P}(\|T - \widehat{T}\|_2 < \epsilon) &\geq \mathbb{P}\left(\|M - \widehat{M}\|_2 \lambda_{\max}(D) < \frac{\lambda_{\min}(\widehat{T}^2)}{\sqrt{C} + 1} \epsilon\right) \\ &= \mathbb{P}\left(\|M - \widehat{M}\|_2 < \frac{\lambda_{\min}(\widehat{T}^2)}{(\sqrt{C} + 1)\lambda_{\max}(D)} \epsilon\right) \\ &\geq 1 - 2C^2 e^{-\frac{\epsilon^2}{C^2(\sqrt{C}+1)^2} \frac{\lambda_{\min}(\widehat{T}^2)^2}{\lambda_{\max}(D)^2} n} \end{aligned}$$

For the inverse:

$$T^{-1} - \widehat{T}^{-1} = T^{-1}(\widehat{T} - T)\widehat{T}^{-1} \quad (\text{C.23})$$

So,

$$\|T^{-1} - \widehat{T}^{-1}\|_2 \leq \|T^{-1}\|_2 \|\widehat{T} - T\|_2 \|\widehat{T}^{-1}\|_2 = \frac{1}{\lambda_{\min}(T)\lambda_{\min}(\widehat{T})} \|\widehat{T} - T\|_2$$

Following what we did for the κ in

$$\frac{1}{\lambda_{\min}(T)\lambda_{\min}(\widehat{T})} \leq \frac{1}{\min(\lambda_{\min}(T^2), \lambda_{\min}(\widehat{T}^2))} \leq \frac{1}{\lambda_{\min}(\widehat{T}^2) - |\lambda_{\min}(T^2) - \lambda_{\min}(\widehat{T}^2)|}$$

Than for Equation C.17

$$\frac{1}{\lambda_{\min}(T)\lambda_{\min}(\widehat{T})} \leq \frac{1}{\lambda_{\min}(\widehat{T}^2) - \|T^2 - \widehat{T}^2\|_2}$$

So

$$\|T^{-1} - \widehat{T}^{-1}\|_2 \leq \frac{\|T - \widehat{T}\|_2}{\lambda_{\min}(\widehat{T}^2) - \|T^2 - \widehat{T}^2\|_2} \leq \frac{\|T - \widehat{T}\|_2}{\lambda_{\min}(\widehat{T}^2) - 2\|T - \widehat{T}\|_2}$$

Where we used that

$$\|T^2 - \widehat{T}^2\|_2 \leq \|T(T - \widehat{T}) + (T - \widehat{T})\widehat{T}\|_2 \leq 2\|T - \widehat{T}\|_2$$

because T and \widehat{T} doubly stochastic.

So

$$\mathbb{P}\left(\|T^{-1} - \widehat{T}^{-1}\|_2 \leq \epsilon\right) \geq \mathbb{P}\left(\|T - \widehat{T}\|_2 \leq \epsilon \frac{\lambda_{\min}(\widehat{T})}{1 + 2\epsilon}\right) \quad (\text{C.24})$$

$$\geq \mathbb{P}\left(\|T - \widehat{T}\|_2 \leq \frac{\epsilon}{3} \lambda_{\min}(\widehat{T})\right) \quad (\text{C.25})$$

$$\geq 1 - 2C^2 e^{-\frac{\epsilon^2}{9C^2(\sqrt{C}+1)^2} \frac{\lambda_{\min}(\widehat{T})^4}{\lambda_{\max}(D)^2} n} \quad (\text{C.26})$$

□

Proof of Theorem 5.6: generalization gap bounds

Proposition 5.3.1. Let $\ell(t, y)$ be any bounded loss function and let $l(t, y)$ be the backward loss function defined in Eq. (3.24a).

We define $\hat{l}(t, y)$ as the loss obtained using $\hat{\Gamma}^{-1} := \hat{T}^{-1}$. If μ is the constant that bounded the loss ℓ , i.e. $\sup_{(t,y) \in [0,1]^C \times \mathcal{Y}} \ell(t, y) \leq \mu$. For every ϵ

$$\mathbb{P}(|l(t, y) - \hat{l}(t, y)| \geq \epsilon) \leq 2C^2 e^{-2 \frac{\epsilon^2}{C^2 \mu^2 L_{\phi, p}} n} \quad (\text{C.27})$$

Proof of Proposition 5.3.1. Using Cauchy–Schwarz inequality and the fact that ℓ is bounded by μ and that we obtain:

$$\begin{aligned} |l(t, y) - \hat{l}(t, y)| &= |(T^{-1} \cdot \ell(t) - \hat{T}^{-1} \cdot \ell(t))_y| \\ &= |[(T^{-1} - \hat{T}^{-1})\ell(t)] \cdot e_y| \\ &\leq \|(T^{-1} - \hat{T}^{-1})\ell(t)\|_2 \|e_y\|_2 \\ &\leq \|T^{-1} - \hat{T}^{-1}\|_2 \|\ell(t)\|_2 \\ &\leq \mu \|T^{-1} - \hat{T}^{-1}\|_2 \end{aligned}$$

So

$$\mathbb{P}(|l(t, y) - \hat{l}(t, y)| \leq \epsilon) \geq 1 - 2C^2 e^{-\frac{\epsilon^2}{\mu^2 2C^2 (\sqrt{C}+1)^2} \frac{\lambda_{\min}(\hat{T}^2)^4}{\lambda_{\max}(D)^2} n}$$

□

Proof Lemma 5.5 . For every f we have

$$|\hat{R}_i(f) - R_{l, \mathcal{D}}(f)| \leq |\hat{R}_i(f) - \hat{R}_l(f)| + |\hat{R}_l(f) - R_{l, \mathcal{D}}(f)|.$$

So using union bounds and by the classic results on Rademacher complexity bounds [192], and by the Lipschitz composition property of Rademacher averages, Theorem 7 in [186] it follows that

$$\mathbb{P}^n \left(\sup_{f \in \mathcal{F}} |\hat{R}_i(f) - R_{l, \mathcal{D}}(f)| \leq L \|T^{-1}\|_2 \mathfrak{R}_n(\mathcal{F}) + \frac{\epsilon}{2} \right) \geq \quad (\text{C.28})$$

$$\mathbb{P}^n \left(\sup_{f \in \mathcal{F}} |\hat{R}_i(f) - \hat{R}_l(f)| + \sup_{f \in \mathcal{F}} |\hat{R}_l(f) - R_{l, \mathcal{D}}(f)| \leq L \|T^{-1}\|_2 \mathfrak{R}_n(\mathcal{F}) + \frac{\epsilon}{2} \right) \geq \quad (\text{C.29})$$

$$1 - \mathbb{P}^n \left(\sup_{f \in \mathcal{F}} |\hat{R}_i(f) - \hat{R}_l(f)| > \frac{\epsilon}{4} \right) - \mathbb{P}^n \left(\sup_{f \in \mathcal{F}} |\hat{R}_l(f) - R_{l, \mathcal{D}}(f)| \leq L \|T^{-1}\|_2 \mathfrak{R}_n(\mathcal{F}) + \frac{\epsilon}{4} \right) \quad (\text{C.30})$$

$$\geq 1 - \mathbb{P}^n \left(\sup_{f \in \mathcal{F}} |\hat{R}_i(f) - \hat{R}_l(f)| > \frac{\epsilon}{4} \right) - 2e^{-\frac{n}{2} \left(\frac{\epsilon}{4\mu} \right)^2} \quad (\text{C.31})$$

Now,

$$\begin{aligned}
& \mathbb{P}^n \left(\sup_{f \in \mathcal{F}} |\hat{R}_l(f) - R_{l, \mathcal{D}}(f)| \leq L \|\widehat{T}^{-1}\|_2 \mathfrak{R}_n(\mathcal{F}) + \epsilon \right) = \\
& \mathbb{P}^n \left(\sup_{f \in \mathcal{F}} |\hat{R}_l(f) - R_{l, \mathcal{D}}(f)| \leq L \|\widehat{T}^{-1}\|_2 \mathfrak{R}_n(\mathcal{F}) + \epsilon \right) \\
& \mathbb{P}^n \left(\sup_{f \in \mathcal{F}} |\hat{R}_l(f) - R_{l, \mathcal{D}}(f)| \leq L \|T^{-1}\|_2 \mathfrak{R}_n(\mathcal{F}) + (\|\widehat{T}^{-1}\|_2 - \|T^{-1}\|_2) \mathfrak{R}_n(\mathcal{F}) + \epsilon \right) \geq \\
& 1 - \mathbb{P}^n \left(\left\{ \sup_{f \in \mathcal{F}} |\hat{R}_l(f) - R_{l, \mathcal{D}}(f)| \leq L \|T^{-1}\|_2 \mathfrak{R}_n(\mathcal{F}) + \frac{\epsilon}{2} \right\} \text{ and } \{ (\|\widehat{T}^{-1}\|_2 - \|T^{-1}\|_2) \mathfrak{R}_n(\mathcal{F}) \leq \frac{\epsilon}{2} \} \right) \geq \\
& 1 - \mathbb{P}^n \left(\sup_{f \in \mathcal{F}} |\hat{R}_l(f) - R_{l, \mathcal{D}}(f)| \leq L \|T^{-1}\|_2 \mathfrak{R}_n(\mathcal{F}) + \frac{\epsilon}{2} \right) - \mathbb{P}^n \left((\|\widehat{T}^{-1}\|_2 - \|T^{-1}\|_2) \mathfrak{R}_n(\mathcal{F}) \leq \frac{\epsilon}{2} \right) \geq \\
& 1 - 2e^{-\frac{n}{2} \left(\frac{\epsilon}{4\mu} \right)^2} - \mathbb{P}^n \left((\|\widehat{T}^{-1} - T^{-1}\|_2) \leq \frac{\epsilon}{2\mathfrak{R}_n(\mathcal{F})} \right) - \mathbb{P}^n \left(\sup_{f \in \mathcal{F}} |\hat{R}_l(f) - \hat{R}_l(f)| > \frac{\epsilon}{4} \right) \geq \\
& 1 - 2e^{-\frac{n}{2} \left(\frac{\epsilon}{4\mu} \right)^2} - 2C^2 e^{-\frac{\epsilon^2}{4\mathfrak{R}_n(\mathcal{F})^2 9C^2 (\sqrt{C}+1)^2} \frac{\lambda_{\min}(\widehat{T}^2)^4}{\lambda_{\max}(D)^2} n} - 2C^2 e^{-\frac{\epsilon^2}{4\mu^2 9C^2 (\sqrt{C}+1)^2} \frac{\lambda_{\min}(\widehat{T}^2)^4}{\lambda_{\max}(D)^2} n} \\
& \geq 1 - 2e^{-\frac{n}{2} \left(\frac{\epsilon}{4\mu} \right)^2} - 4C^2 e^{-\frac{1}{\max(\mathfrak{R}_n(\mathcal{F}), \mu)^2} \frac{\epsilon^2}{36C^2 (\sqrt{C}+1)^2} \frac{\lambda_{\min}(\widehat{T}^2)^4}{\lambda_{\max}(D)^2} n} \\
& \geq 1 - 4e^{-\left[\min \left(\frac{1}{8}, 2 \ln(C) \frac{1}{9\mathfrak{R}_n(\mathcal{F})^2 C^2} \frac{\lambda_{\min}(\widehat{T}^2)^4}{(\sqrt{C}+1)^2 \lambda_{\max}(D)^2} \right) \right] \frac{\epsilon^2}{4\mu^2} n} \\
& \geq 1 - 4C e^{-\left(\frac{1}{9\mathfrak{R}_n(\mathcal{F})^2 C^2} \frac{\lambda_{\min}(\widehat{T}^2)^4}{(\sqrt{C}+1)^2 \lambda_{\max}(D)^2} \right) \frac{\epsilon^2}{2\mu^2} n}
\end{aligned}$$

So with probability at least $1 - \delta$

$$\sup_{f \in \mathcal{F}} |\hat{R}_l(f) - R_{l, \mathcal{D}}(f)| \leq 2L \lambda_{\min}(\widehat{T}^2) \mathfrak{R}_n(\mathcal{F}) + \frac{6\mu \mathfrak{R}_n(\mathcal{F}) \lambda_{\min}(D) C^2 (\sqrt{C} + 1)}{\lambda_{\min}(\widehat{T})^2} \sqrt{\frac{1}{n} \ln \left(\frac{4C}{\delta} \right)}.$$

Or

$$\sup_{f \in \mathcal{F}} |\hat{R}_l(f) - R_{l, \mathcal{D}}(f)| \leq \left[2L \lambda_{\min}(\widehat{T}^2) + \frac{\mu \lambda_{\min}(D)}{\lambda_{\min}(\widehat{T})^2} \sqrt{\frac{1}{n} \ln \left(\frac{4C}{\delta} \right)} \right] \mathfrak{R}_n(\mathcal{F}) g(C). \quad (\text{C.32})$$

with $g(C) = 6C^2(\sqrt{C} + 1)$

□

Theorem 5.6. By the unbiasedness of l we have that $R_{l, \mathcal{D}}(\hat{f}) = R_{l, \mathcal{D}}(\hat{f})$. Moreover since $\hat{f} = \underset{f}{\operatorname{argmin}}(\hat{R}_l(f))$ we have $\hat{R}_l(\hat{f}) \leq \hat{R}_l(g) \forall g \in \mathcal{F}$.

Let f^* be so that $\min_{f \in \mathcal{F}} R_{\ell, \mathcal{D}}(f) = R_{\ell, \mathcal{D}}(f^*)$. It follows that

$$\begin{aligned} R_{\ell, \mathcal{D}}(\hat{f}) - \min_{f \in \mathcal{F}} R_{\ell, \mathcal{D}}(f) &= R_{\ell, \mathcal{D}}(\hat{f}) - \min_{f \in \mathcal{F}} R_{\ell, \mathcal{D}}(f) \\ &= R_{\ell, \mathcal{D}}(\hat{f}) - \widehat{R}_{\ell, \mathcal{D}}(\hat{f}) + \widehat{R}_{\ell, \mathcal{D}}(\hat{f}) - R_{\ell, \mathcal{D}}(f^*) \\ &\geq R_{\ell, \mathcal{D}}(\hat{f}) - \widehat{R}_{\ell, \mathcal{D}}(\hat{f}) - (R_{\ell, \mathcal{D}}(f^*) - \widehat{R}_{\ell, \mathcal{D}}(f^*)) \\ &\geq 2 \max_{f \in \mathcal{F}} |R_{\ell, \mathcal{D}}(f) - \widehat{R}_{\ell, \mathcal{D}}(f)| \end{aligned}$$

□

Lemma 5.4. Let us consider a vector $\mathbf{x} = (x_1, \dots, x_n)$ s.t. $\sum_{i=1}^n x_i = 1$ and $x_i > 0$ for all i , and a vector $\mathbf{a} = (a_1, \dots, a_n)$ s.t. $a_i > 0$ for $i = 1, \dots, n$. Let $\psi_{\mathbf{a}}(\mathbf{x}) = \prod_{j=1}^n x_j^{a_j}$, it holds that

$$\operatorname{argmax}_{(x_1, \dots, x_n): \sum_i x_i = 1} \psi_{\mathbf{a}}(\mathbf{x}) = (a_1, \dots, a_n)$$

Proof. Let us consider $\phi_{\mathbf{a}}(\mathbf{x}) = \log \psi_{\mathbf{a}}(\mathbf{x}) = \sum_{i=1}^n a_i \log(x_i)$. Recalling that $x_n = 1 - \sum_{i=1}^{n-1} x_i$

$$\nabla \phi(\mathbf{x}) = \begin{bmatrix} \frac{a_1}{x_1} - \frac{a_n}{1 - \sum_{i=1}^{n-1} x_i} \\ \frac{a_2}{x_2} - \frac{a_n}{1 - \sum_{i=1}^{n-1} x_i} \\ \vdots \\ \frac{a_1}{x_{n-1}} - \frac{a_n}{1 - \sum_{i=1}^{n-1} x_i} \end{bmatrix}$$

$$\nabla \phi(\mathbf{x}) \geq 0 \iff a_i \left(1 - \sum_{i=1}^{n-1} x_i\right) - a_n x_i \geq 0 \text{ for } i = 1, \dots, n-1$$

Namely, the maximum is reached for \mathbf{x} that solves the following linear system has to be solved:

$$\begin{bmatrix} a_1 + a_n & a_1 & \dots & a_1 \\ a_2 & a_2 + a_n & \dots & a_2 \\ \vdots & \vdots & \ddots & \vdots \\ a_{n-1} & \dots & a_{n-1} + a_n & \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \end{bmatrix} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_{n-1} \end{bmatrix}$$

We have that

$$A := \begin{bmatrix} a_1 + a_n & a_1 & \dots & a_1 \\ a_2 & a_2 + a_n & \dots & a_2 \\ \vdots & \vdots & \ddots & \vdots \\ a_{n-1} & \dots & a_{n-1} + a_n & \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \cdot [a_1, a_2, \dots, a_{n-1}] + a_n \mathbf{I}_{n-1}$$

A can be written as the sum of a rank-1 matrix and a_n times the identity. It holds that is $\sum_{i=1}^{n-1} a_i + a_n \neq 0$ then A is invertible so $\operatorname{rank}(A) = n - 1$ [262]. The non-homogeneous system also has one unique solution. We know that $x_i = a_i$ is a solution, so it's the unique solution.

□

Proof of Lemma 5.4

Lemma 5.5. For infinite annotators the posterior distribution over every sample calculated using the true T converges to the dirac delta distribution centered on the true label almost surely (i.e. $\lim_{H \rightarrow \infty} p_{c,i} \stackrel{\text{a.s.}}{=} \mathbb{1}(y_i = c)$).

Proof.

$$p_{c,i} = \frac{\mu_c \prod_{h=1}^H T_{c,y_h,i}}{\sum_{j=1}^C \mu_j \prod_{h=1}^H T_{j,y_h,i}} \quad (\text{C.33})$$

$$\prod_{h=1}^H T_{c,y_h,i} = \prod_{j=1}^C T_{c,j}^{N_{i,j}} \quad (\text{C.34})$$

where $N_{i,j}$ is the amount of annotators that labeled sample i as class j . Note that as a consequence of the strong law of large numbers for the conditional random variables that are independent with the same conditional distribution we have that the following equation is true almost surely:

$$\lim_{H \rightarrow \infty} \frac{N_{i,j}}{H} = \lim_{H \rightarrow \infty} \frac{\sum_{a=1}^H \mathbb{1}_{\{y_{a,i}=j\}}}{H} = \mathbb{E}[\mathbb{1}_{\{y_{a,i}=j\}} | y = j] = T_{y_i,j} \quad (\text{C.35})$$

Combining we get:

$$\lim_{H \rightarrow \infty} p_{c,i} = \lim_{H \rightarrow \infty} \frac{\mu_c \prod_{j=1}^C T_{c,j}^{N_{i,j}}}{\sum_{k=1}^C \mu_k \prod_{j=1}^C T_{k,j}^{N_{i,j}}} \quad (\text{C.36})$$

$$= \lim_{H \rightarrow \infty} \frac{\mu_c \left(\prod_{j=1}^C T_{c,j}^{T_{y_i,j}} \right)^H}{\sum_{k=1}^C \mu_k \left(\prod_{j=1}^C T_{k,j}^{T_{y_i,j}} \right)^H} \quad (\text{C.37})$$

$$= \lim_{H \rightarrow \infty} \frac{1}{1 + \sum_{\substack{k=1 \\ k \neq c}}^C \frac{\mu_k}{\mu_c} \left(\prod_{j=1}^C \left(\frac{T_{k,j}}{T_{c,j}} \right)^{T_{y_i,j}} \right)^H} \quad (\text{C.38})$$

$$\stackrel{(a)}{=} \mathbb{1}(y_i = c) \quad (\text{C.39})$$

where in (a) we used the fact that due to the assumption that T is strictly dominant, then the term $\prod_{j=1}^C T_{k,j}^{T_{y_i,j}}$ is maximized when $k = y_i$ and this term is strictly larger than all the other ones, see Lemma 5.4. Indeed, if there exists k s.t. $\prod_{j=1}^C \left(\frac{T_{k,j}}{T_{c,j}} \right)^{T_{y_i,j}} > 1$ than $\lim_{H \rightarrow \infty} p_{c,i} = 0$ because the denominator goes to ∞ .

So the only case for not having is that $\prod_{j=1}^C \left(\frac{T_{k,j}}{T_{c,j}} \right)^{T_{y_i,j}} \leq 1$ for all k . Suppose we know that the maximum of the function $\prod_{j=1}^C (x_j)^{a_j}$ is reached for $x_j = a_j \forall j = 1, \dots, C$ than we're done.

Indeed, we have that $\prod_{j=1}^C \left(\frac{T_{k,j}}{T_{c,j}} \right)^{T_{y_i,j}} > 1$, if and only if $\prod_{j=1}^C (T_{k,j})^{T_{y_i,j}} > \prod_{j=1}^C (T_{c,j})^{T_{y_i,j}}$ since we're considering all k , for $k = 1, \dots, C, k \neq c$. If $y_i \neq c$ it means that y_i is one of the values k can assume and since that one is the max, it means that for sure it will be greater than $\prod_{j=1}^C (T_{c,j})^{T_{y_i,j}}$.

Otherwise, if $y_i = c$ it means that $\prod_{j=1}^C (T_{c,j})^{T_{y_i,j}} > \prod_{j=1}^C (T_{k,j})^{T_{y_i,j}}$ so all elements are less than 1 and the limit goes to 1. \square

Proof of Proposition 5.1: relationship between ρ and κ .

Proof.

$$\begin{aligned}
p_o &= \mathbb{P}(y_a = y_B) = \sum_{k,h=1}^C \mathbb{P}(y_A = k, y_B = k | y = h) \mathbb{P}(y = h) \\
&= \sum_{k,h=1}^C \mathbb{P}(y_A = k | y = h) \mathbb{P}(y_B = k | y = h) \nu_h = \sum_{k,h=1}^C T_{h,k}^2 \nu_h \\
&= \sum_{h=1}^C (1-p)^2 c_h + \sum_{h=1}^C \left(\frac{p}{C-1} \right)^2 (C-1) c_h = (1-p)^2 + \frac{p^2}{C-1}
\end{aligned}$$

Now

$$\mathbb{P}(y_B = k) = \sum_{h=1}^C \mathbb{P}(y_B = k | y = h) \mathbb{P}(y = h) = \sum_{h=1}^C T_{hk} \nu_h = (T\nu)_k$$

In the previous equation, we used that T is symmetric.

$$\begin{aligned}
p_e &= \sum_{k=1}^C \mathbb{P}(y_A = k) \mathbb{P}(y_B = k) = \sum_{k=1}^C \mathbb{P}(y_A = k) \mathbb{P}(y_B = k) = c^T T^2 c \\
&= 2 \frac{p}{C-1} - \frac{Cp^2}{(C-1)^2} + \left(1 - \frac{Cp}{C-1} \right)^2 \nu^T \nu
\end{aligned} \tag{C.40}$$

If the distribution of the true label y is symmetric the probability vector $\nu = (\frac{1}{C}, \dots, \frac{1}{C})$. So $\nu^T \nu = \frac{1}{C}$ and so

$$\kappa = \frac{C^2 p^2 - 2C(C-1)p + (C-1)^2}{(C-1)^2} \tag{C.41}$$

From which it follows that

$$p = (1 - C^{-1})(1 - \sqrt{\kappa}) \tag{C.42}$$

□

3.0.4 Experiments

Estimation of T

From Figure C.1, we can notice that the error in the estimation decreases as $\frac{1}{\sqrt{n}}$ the n number of samples increases. The results with respect to the minimum eigenvectors and with respect to the maximum diagonal value are consistent with each other and very similar.

The results were obtained from a synthetic, generated dataset in which we generate the classes predicted by the annotators according to various T matrices, choosing as all possible (admissible) combinations that have $[0, 0.2, 0.4]$ out of the diagonal and $[0.6, 0.8, 1.0]$ on the diagonal. We can notice in Figure C.1 that the estimation becomes more precise as the number of annotators increases.

For experiments with 2, 3 and 7 annotators, we generate T as all possible symmetric, stochastic

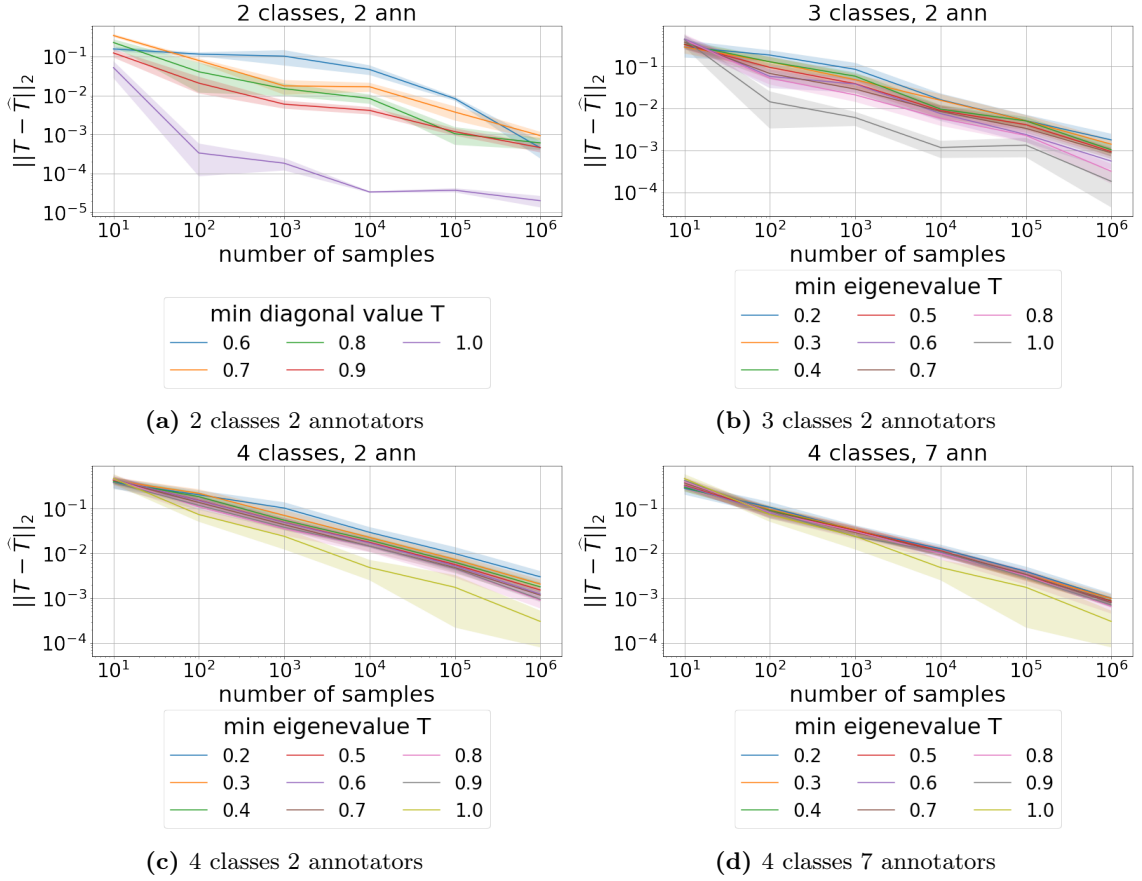


Figure C.1: Error in the Estimation of T . The error is $\|T - \hat{T}\|_2$. We aggregated the matrices with the same minimum eigenvalue rounded at the first decimal.

and diagonally dominant matrices with $[0.1, 0.2, 0.3, 0.4, 0.5]$ out of the diagonal and $[0.6, 0.8, 1.0]$ on the diagonal. Classes are uniformly distributed. For experiments with 10 annotators, we generate the matrices T as all possible (admissible) combinations that have $[0, 0.2, 0.4]$ out of the diagonal and $[0.6, 0.8, 1.0]$ on the diagonal. In this case, we both include uniform distribution of the true labels among the 4 classes and all the distributions so that the four classes can be partitioned into two groups of indices so that classes in the same group have the same probability. Namely, if the distributions on the classes are given by $\mathbf{d} = [d_1, d_2, d_3, d_4]$, admissible distributions are the ones for which there are two subsets of indices I and J so that $I \cup J = \{0, 1, 2, 3, 4\}$ and for all $i, k \in I : d_i = d_k$. The probability of the classes takes value in $[0.1, 0.2, 0.3, 0.4]$. This means that, for instance, we will find the distribution $[0.3, 0.3, 0.3, 0.1]$ or the distribution $[0.4, 0.1, 0.1, 0.4]$ but not $[0.3, 0.2, 0.1, 0.4]$.

Results for 2, 3 and 7 annotators were obtained by averaging over 3 runs. Results for 10 annotators were obtained by averaging over 10 runs. The error that appears on axis y in the plots is the difference in norm 2 of the true matrix T and the estimated matrix \hat{T} , obtained as explained in Section 3.3.4.

We recall that if the minimum eigenvalue is 1, the matrix T is the identity, and thus, the annotators always predict the exact class. The smaller the minimum eigenvalue, the noisier the dataset will be.

With Figure C.2 we wanted to see if datasets with a higher noise level have higher approximation

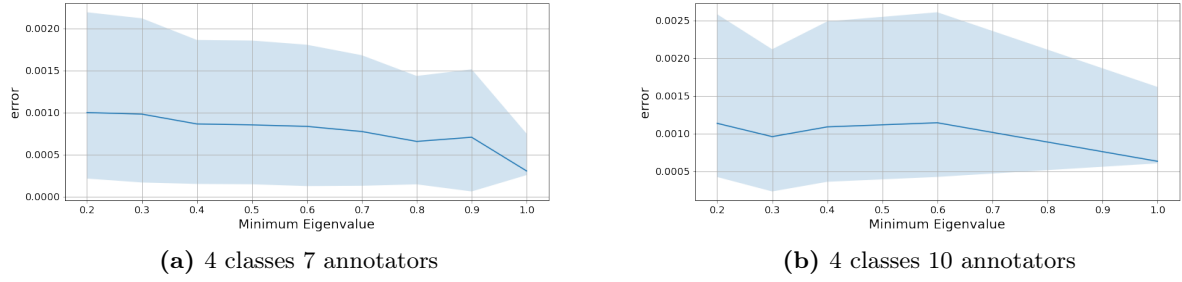


Figure C.2: The plots show the trend of the error estimation as the minimum eigenvalue increases

errors than less noisy datasets. The plots show a minor trend: the estimation error also decreases as the noise decreases. The trend is not particularly noticeable perhaps due to a large number of annotators.

We recall that if the minimum eigenvalue is 1 or the maximum value of the diagonals is 1, the matrix T is the identity, and thus, the annotators always predict the exact class. The smaller the minimum eigenvalue or the maximum value on the diagonal, the noisier the dataset will be.

Synthetic datasets

The synthetic dataset consists of two-dimensional features ($\mathbf{x} = (x_1, x_2)$). To create the dataset, we generate points uniformly at random in $[0, 1]^2$. Each point is then assigned a label (y) based on the predetermined label distribution for each experiment. We divide the space into sections using lines parallel to the bisector of the first and third quadrants (specifically, $x_2 = x_1$). See Figure C.3 for an example. Our dataset comprises 10000 samples. In Figure C.4 we see, for different amounts of noise, the results of the different aggregation methods when using a neural network without hidden layer (i.e. a Logistic Regression) trained with Cross Entropy Loss. When noise is absent, we check that, as expected, the results are all identical. In the presence of noise (0.6 and 0.8), we notice in general that the random aggregation is the worst. The others are equivalent, except for the posterior (ours) which obtains slightly higher results. Average, on the other hand, obtains a slightly lower value with minimum diagonal value of T equal to 0.8. However, attention must be drawn to the fact that the y-scale of the graph is very narrow and that in the case of 4 classes with a dataset constructed as in Figure C.3, a linear classifier is not able to reach perfect accuracy.

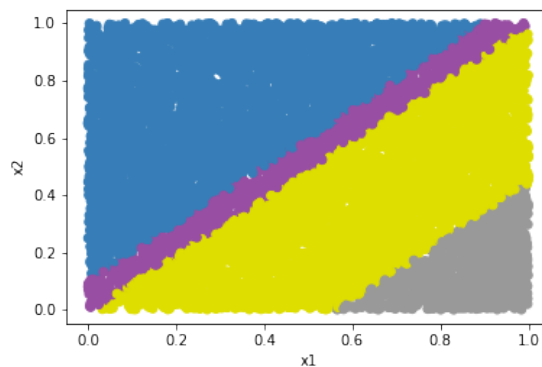


Figure C.3: Synthetic data for 4 classes with distribution (0.4,0.1,0.4,0.1)

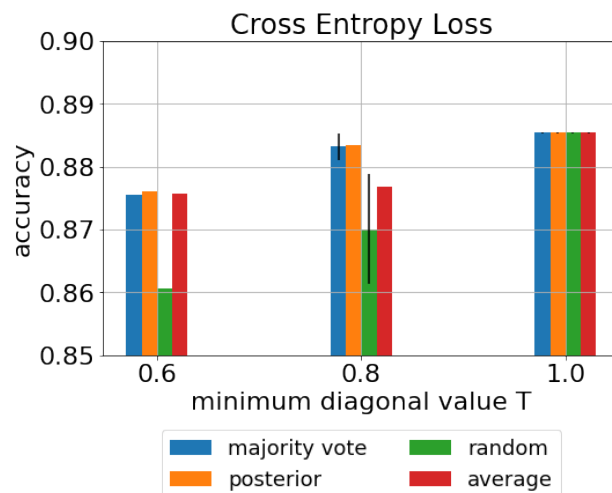


Figure C.4: 5 annotators, 4 classes, no hidden layer.

Figure C.5

Referring to Figure 3.9 and the other figures of this section. The minimum value on the diagonal of the matrix T denotes the annotators’ probability of assigning the correct label for the class in which the noise is maximum. As expected, random aggregation is the lowest performing method, and for all noise rates soft label methods perform better than methods using hard labels.

Figure C.4 shows the accuracy for the case of 4 classes and a NN with no hidden layer and 5 annotators. We can notice that even when the number of hidden neurons is insufficient to obtain perfect accuracy. Hence, the classifier is not the best possible; our approach for a high-noise dataset performs better.

The posteriors distribution is computed using the estimated T .

Implementation details

Logistic Regression is used for synthetic data with 2 classes, and a neural network with a hyperbolic tangent activation function with one hidden layer is used for the dataset with more classes. The data are separated into train, validation and test set using a split 64%, 16%, 20%. The models are trained with the following configuration: batch size 256, learning rate 10^{-3} , maximum number of epochs 1000, early stopping of training based on validation loss with patience of 100 epochs. Once the training is finished, the model with the lowest validation loss is retrieved.

For the experiments with CIFAR-10, the model, Resnet 34, is trained with the following configuration: batch size 128, learning rate 10^{-3} , with momentum (0.9) and learning rate decay (0.0005) the maximum number of epochs 1000, we also used early stopping of training based on validation loss with patience of 100 epochs. We didn’t use data augmentation. For the pre-trained model, we used the model provided by torch-vision, <https://pytorch.org/vision/main/models/generated/torchvision.models.resnet34.html#resnet34>.

All code is written in Python 3 Programming Language. The cvxpy package optimises \hat{T} , and the PyTorch library is used for the models. All the experiments have been run on a machine with this configuration: AMD EPYC 7373 Processor, 64GB RAM and NVIDIA GeForce RTX A4000 GPU.

Appendix D

Personalized Recourse - Supplementary Materials

Here, we outline how we modeled and simulated the preferences for the variables in the datasets.

- Gaussian Preferences: $\theta_2 \in (0, 10]$
- Exponential Preferences: $\theta \in (0, 10]$
- Categorical Preferences: $\theta_i \in (0, 1) \quad \forall i \in \{1, \dots, K\} \quad s.t. \sum_{i=1}^K \theta_i = 1$, where K is the number of categories the feature has

Adult Income

- age: exponential
- workclass: categorical
- education: categorical
- marital_status: categorical
- occupation: categorical
- race: degenerate
- gender: degenerate
- hours_per_week: gaussian
- income: target

GiveMeSomeCredit

- RevolvingUtilizationOfUnsecuredLines: gaussian
- age: exponential
- NumberOfTime30-59DaysPastDueNotWorse: exponential

- DebtRatio: gaussian
- MonthlyIncome: gaussian
- NumberOfOpenCreditLinesAndLoans: gaussian
- NumberOfTimes90DaysLate: exponential
- NumberRealEstateLoansOrLines: gaussian
- NumberOfTime60-89DaysPastDueNotWorse: exponential
- NumberOfDependents: exponential
- SeriousDlqin2yrs: target

HELOC

- ExternalRiskEstimate: gaussian
- MSinceOldestTradeOpen: gaussian
- MSinceMostRecentTradeOpen: gaussian
- AverageMInFile: gaussian
- NumSatisfactoryTrades: exponential
- NumTrades60Ever2DerogPubRec: gaussian
- NumTrades90Ever2DerogPubRec: gaussian
- PercentTradesNeverDelq: gaussian
- MSinceMostRecentDelq: gaussian
- MaxDelq2PublicRecLast12M: gaussian
- MaxDelqEver: exponential
- NumTotalTrades: exponential
- NumTradesOpeninLast12M: gaussian
- PercentInstallTrades: gaussian
- MSinceMostRecentInqexcl7days: gaussian
- NumInqLast6M: gaussian
- NumInqLast6Mexcl7days: gaussian
- NetFractionRevolvingBurden: gaussian
- NetFractionInstallBurden: gaussian

- NumRevolvingTradesWBalance: gaussian
- NumInstallTradesWBalance: gaussian
- NumBank2NatlTradesWHighUtilization: gaussian
- PercentTradesWBalance: gaussian
- RiskPerformance: target

Acknowledgements

This journey through the realm of Trustworthy Artificial Intelligence has been a profound and enlightening experience, made possible through the support, guidance, and contributions of numerous individuals and institutions. As I conclude this thesis, I extend my sincere gratitude to those who have been instrumental in its realization.

First and foremost, I extend my deepest thanks to my advisor, Fabrizio Silvestri, whose unwavering support, guidance, and wisdom have been invaluable throughout this research journey. His mentorship has been pivotal not only in the development of this thesis and our collaborative endeavors but also in shaping me as a researcher. Your expertise and dedication have been the guiding beacons on this scholarly voyage.

I am also indebted to the esteemed faculty members and colleagues at Sapienza University of Rome. In particular, I wish to express my appreciation for their valuable insights and contributions to my research to Professors Stefano Leonardi, Luca Becchetti, and Gabriele Tolomei. Additionally, I am grateful to Professors from other institutions, including Pietro Liò of Cambridge and Nicola Tonellotto of Pisa; special thanks to Prof. Antonio Pasquini for passing on his passion for studying and teaching. I extend my thanks to fellow doctoral candidates at Sapienza for their camaraderie and collaboration. My gratitude extends to all the members of the RSTLess group, with a special mention of Giovanni and Andrea. Their contributions and support have been invaluable to the success of this work. I must acknowledge Amazon, and in particular, Iftah and Shoal, for the enriching experiences I gained during my time with them.

I would like to express my appreciation to Maria Sofia, who has shared the entire doctoral journey with me, and I look forward to continuing our research path together. Special thanks go to Davide, a friend and fellow PhD candidate, whose adventurous tales have always been a source of amusement. Giulia, who brought joy to our doctoral classroom and to me personally, and with whom I hope to work for a long time. Valerio, a friend and colleague in the PhD program, whose sharp mind and humility I have always admired, and I regret not collaborating with him more extensively.

Thanks to my long-time friends, Marco and Edoardo, whose companionship consistently evokes cherished memories of the past. I am also deeply grateful to Fabio, a steadfast and authentic friend whose support has been invaluable. To my family, whose unwavering encouragement have been my constant motivation, I offer my heartfelt thanks. Your belief in my aspirations has been the cornerstone of my academic pursuit. My mother, who has shaped my life. My father, for laughter and assistance, even if at times not explicitly stated. My little brother, who always reminds me that there's no point being grown-up if you can't be childish sometimes. My sister, the true genius in our family, who has much more to show the world. My aunt Roberta, a great researcher, who instilled in me a passion for research from a young age and continues to be a source of inspiration and support. My beloved artist grandfather, dearly missed yet ever-present in my heart. To Aurora, who has been a constant source of comfort and strength, always by my side in times of adversity, as well as in joyful moments.

In closing, I recognize that this work is the culmination of the efforts of many, and I humbly acknowledge their contributions. This thesis stands as a testament to our shared dedication to advancing the field of Trustworthy AI, and I look forward to the ongoing pursuit of excellence in this domain.

