



SAPIENZA
UNIVERSITÀ DI ROMA

Sapienza University of Rome

Facoltà di Informatica
PhD in Computer Science

THESIS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

Capitalizing on Self-supervision and Pre-trained Models in Computer Vision

Thesis Advisor
Prof. Fabio Galasso

Candidate
Muhammad Rameez Ur Rahman
1977923

Academic Year MMXX-MMXXIII (XXXVI cycle)

Abstract

This thesis addresses the overarching challenge of advancing computer vision tasks under the constraints of limited labeled data and the imperative to capitalize on pre-existing knowledge encoded in pre-trained models. By exploring three distinct computer vision tasks - classification, regression, and segmentation - this work presents diverse frameworks aimed at transcending the conventional boundaries imposed by data scarcity and task-specific methodologies.

The first focus lies on Unsupervised Domain Adaptation (UDA) in visual recognition, a critical endeavor in bridging disparate visual domains for robust real-world performance. Existing approaches in UDA typically necessitate manual adaptation to specific backbone architectures, hindering adaptability over time as methods become outdated with evolving architectures. To circumvent this limitation, this thesis proposes a novel approach termed Adversarial Branch Architecture Search for UDA (ABAS). ABAS addresses the lack of target labels by employing a data-driven ensemble approach for model selection and explores auxiliary adversarial branches to drive domain alignment. Extensive validation on standard visual recognition datasets demonstrates ABAS's efficacy in enhancing modern UDA techniques, robustly yielding superior performances across diverse domains.

In the realm of regression tasks, the thesis delves into collaborative human pose forecasting, an understudied domain with the potential for improved performance through exploiting the correlated motion patterns of interacting individuals. By revisiting prevalent single-person practices and tailoring them to the collaborative setting, significant advancements are achieved. Notably, the integration of frequency input representations, space-time separable interaction encodings, and fully-learnable interaction adjacencies into a Graph Convolutional Network (GCN) framework showcases promising results. Furthermore, a novel initialization procedure for spatial interaction parameters enhances both performance and stability, culminating in a substantial performance boost over state-of-the-art methods on benchmark datasets.

Lastly, the thesis tackles semantic segmentation in autonomous driving scenarios, leveraging the unique capabilities of event cameras for low-latency operation in challenging lighting conditions. We introduce OVOSE, the first open-vocabulary semantic segmentation approach explicitly tailored for event-based data. OVOSE leverages knowledge distillation from pre-trained image-based models and synthetic event data to enhance segmentation performance. Additionally, we propose a novel dissimilarity network to recalibrate mask loss, mitigating the effects of sub-optimal reconstructions and enabling precise fine-tuning of the segmentation model. Through this novel approach, OVOSE demonstrates superior performance in dynamic environments, outperforming existing conventional image-based models and state-of-the-art methods in unsupervised domain adaptation for event-based semantic segmentation.

In summary, this thesis presents a holistic approach to computer vision tasks, unifying disparate methodologies under the common goal of leveraging pre-trained models and limited labels to achieve superior performance across diverse domains. By addressing specific challenges within classification, regression, and segmentation tasks, the proposed frameworks contribute towards advancing the frontier of computer vision in real-world applications.

Keywords: Neural Architecture Search, Domain Adaptation, Human Pose Forecasting, Event Cameras, Semantic Segmentation, Knowledge Distillation.

Contents

List of Figures	v
List of Tables	viii
1 Introduction	1
1.1 Publications	3
1.2 Publications under review	4
2 Adversarial Branch Architecture Search for Unsupervised Domain Adaptation	5
2.1 Overview	5
2.2 Related Work	7
2.3 Method	8
2.3.1 Background	8
2.3.2 ABAS Search space	9
2.3.3 Ensemble-based Model Selection (EMS)	10
2.3.4 Auxiliary branch optimization	11
2.4 Experiments	11
2.4.1 Preliminary analysis	15
2.4.2 Comparison with SOTA	15
2.4.3 Impact of the auxiliary branch configuration on the final accuracy	16
2.5 Discussion	16
3 Best Practices for 2-body Pose Forecasting	19
3.1 Overview	19
3.2 Related Work	20
3.3 Methodology	21
3.3.1 Input Representation	22
3.3.2 Encoding Best Practices	22
3.3.3 Decoding Best Practices	24
3.3.4 Novel Adjacency Matrix Initialization	24
3.4 Experiments	29
3.4.1 Benchmark and baselines	29
3.4.2 Implementation details	31
3.4.3 Evaluation of human pose forecasting	31
3.4.4 Evaluation of Best Practices	34
3.5 Discussion	36
4 OVOSE: Open-Vocabulary Semantic Segmentation in Event-Based Cameras	37
4.1 Overview	37
4.2 Related Work	39
4.2.1 Event Camera Semantic Segmentation	39
4.2.2 Open-Vocabulary Semantic Segmentation	39

4.2.3	Knowledge Distillation	40
4.3	Open-vocabulary Segmentation in Events	40
4.3.1	Overview of OVOSE	41
4.3.2	Distilling Image Embeddings	42
4.3.3	Feature Distillation	42
4.3.4	Mask Re-weighting	42
4.3.5	Category Label Supervision	43
4.4	Experiments and Results	44
4.4.1	Experimental Framework	44
4.4.2	Results	47
4.4.3	Ablation Study	48
4.5	Discussion	49
5	Final Remarks	51
6	Future Work	52
	Bibliography	53
A	Appendix	66
A.1	Ensemble Model Selection (EMS)	66
A.1.1	Training EMS	66
A.1.2	Unsupervised performance estimators	66
A.1.3	Implementation of the Diversity metric	67

List of Figures

1.1	Outline of thesis in flow diagram	2
2.1	The original DANN [37] paper proposed a specific implementation for each of the considered backbone architectures. Later authors (e.g. ALDA [19]) empirically adapted the core idea to different architectures through a process of trial and error. Whenever a new architecture is proposed, this tedious procedure needs to be repeated (note this issue is shared by any other UDA method based on an auxiliary branch). To address this, in ABAS we propose to automatically search for the optimal architecture of the auxiliary branch in a data driven way. Through ABAS, it is possible to seamlessly apply DANN to any future architecture without the need for human supervision. What follows is a brief description, more details can be found in Algorithm 1. At each iteration, the Bayesian Optimization (BO) acquisition function proposes a number of candidate branch architectures. After training, we extract a number of supervised (source) and unsupervised (target) features which are fed into our ensemble-based performance estimator. Its feedback is sent to BO, thus closing the loop.	6
2.2	Many UDA methods fall under this general framework. The auxiliary branch D can compute either a domain discriminator [19, 37], or a self-supervised [13, 127] loss. . .	6
2.3	Top: overview of how the ABAS search space applies to the main architecture (BS is batch size). Bottom: explicit search space definition. Being low-dimensional it can easily be optimized through BO; on the other hand (as shown in Fig. 2.9) each of these parameters has a strong effect on the final accuracy.	7
2.4	Different model configurations trained on Office31 (1st and 3rd) and Office-Home (2nd and 4th). The x and y axes show two different unsupervised validation metrics we use in EMS, while the color represents target accuracy. Clearly, by only using a single metric it is extremely hard to uniquely predict the performance, but the combined use of two metrics makes it easier to distinguish between good and bad models. This shows how different metrics can be complementary and employed to build a better estimator of model performance. The similarity between pairs of plots on different settings (1&3 and 2&4) suggests that an estimator built on one dataset can be successfully transferred on another. Further plots can be found in the Supplementary.	10
2.5	Accuracy histograms obtained by randomly sampling 40 architectures for ALDA’s [19] auxiliary branch and training them on three settings (10% didn’t converge and were excluded). Contrary to many NAS search spaces [154], ours is large enough to offer a significant impact on final performance (around 20%) on all settings. Equally important, while the default ALDA branch performs well on Office31 and Office-Home (the datasets it was tested on), it performs significantly worse on PACS. ABAS, on the other hand, can adapt itself to all settings, consistently outperforming the random sampling baseline. See sec. 2.4.1 for full details.	12
2.6	Samples from office31 dataset. [50, 119]	12
2.7	Samples from Office-Home dataset. [136]	13
2.8	Samples from PACS dataset. [70, 150]	13

2.9	Search space sensitivity analysis. We randomly sampled 40 configurations and trained them on a representative setting for each of the considered datasets (Office31, Office-Home, PACS). It is evident that the optimal value changes significantly across tasks. X-axis represents the search space hyper-parameter with respect to which the plot is computed.	14
3.1	The general architecture of a 2-body pose forecasting model employing best practices. First, 3D joint coordinates are mapped to frequencies by DCT coefficients, a best input representation practice. Secondly, body kinematics are encoded by layers of a GCN $\sigma(A_s A_t X W)$, with separable space-time adjacency matrices $\sigma(A_t, A_s)$, learned unconstrainedly, upon our proposed parameter initialization. Thirdly, the FC-based decoder outputs future poses for the two people, mapped to 3D coordinates with inverse-DCT (IDCT).	20
3.2	Comparison of feature activation variances, at layers 0, 2, 4 and 7, estimated during the model training, upon initialization with random “Uniform”, “Glorot” [42], “He” [48] against “Ours”, our proposed initialization technique.	26
3.3	Visual comparison of our proposed best-practice model (<i>Ours</i>) against ExPI [45]. The first three columns are observed, and the last four are predicted poses. Light-colored and dashed skeletons are GT, and darker and solid ones are predictions. Note the improved larger-displacement motions (<i>Cartwheel</i>).	33
4.1	Output of a regular RGB foundation model for semantic segmentation and OVOSE in event-based data.	38
4.2	Overview of OVOSE pipeline. Our algorithm comprises two components: the original grayscale image branch and the event-based branch. Initially, events are transformed into a grayscale image using the E2VID model. Subsequently, both the original and reconstructed grayscale images undergo text embedding through an image encoder and an MLP. The features from a frozen text-to-image diffusion UNet are then extracted for each tuple of image and text embedding. For each branch, a mask generator predicts class-agnostic binary masks and associated mask embedding features. Categorization is achieved through a dot product between mask embedding features and text embeddings. Both branches are initialized with ODISE weights [149], and knowledge distillation occurs from the original image branch to the event-based branch during training. Original and reconstructed images are input into a dissimilarity network to weigh the distillation in the outputs. During the evaluation, only the event-based branch is utilized.	40
4.3	Dissimilarity network takes original grayscale and reconstructed images as input, and it outputs an error map to reweight the mask loss. It can be seen that E2VID is unable to reconstruct the stripes and hence considered a high error area by the dissimilarity network	43
4.4	The impact of reweighting the mask loss, influenced by the dissimilarity between the grayscale and reconstructed images. Areas such as the person and the elephant’s trunk exhibit suboptimal reconstruction, leading to their exclusion in the reweighting process.	44
4.5	Qualitative samples from ESS in UDA closed-set, E2VID+ODISE, and OVOSE in open vocabulary setting. As compared to ESS and E2VID+ODISE, OVOSE produce accurate and less noisy predictions even though it is trained on a synthetic dataset.	46
4.6	Qualitative samples from DDD17 in UDA closed-set, E2VID+ODISE, and OVOSE in open vocabulary setting. ESS works well in predicting poles and pavement, however OVOSE does better overall prediction, especially vehicles, persons, vegetation, and construction.	46

4.7	Qualitative comparison of ABAS with ESS[128] on DSEC-Semantic dataset in supervised setting utilizing semantic labels. ABAS performs more accurately and close to ground truth semantic segmentation as compared to ESS.	49
A.1	Correlation between the metrics introduced in section 3.3 and the target accuracy, as computed on Office31.	68
A.2	Correlation between the metrics introduced in section 3.3 and the target accuracy, as computed on Office-Home.	68

List of Tables

2.1	Over 400 different points were randomly sampled from our search space and the corresponding models were then trained using ALDA [19] over either Office31 [119] (top 6 rows) or Office-Home [136] settings. Note how the configuration of the auxiliary branch significantly affects the model performance. <i>% Div.</i> stands for the fraction of runs which did not converge - <i>Min</i> is the minimum accuracy - <i>Max</i> is the maximum - <i>Mean</i> reports the mean $\pm$ the standard deviation, and <i>Range</i> is the total range of accuracies, defined as $Max - Min$	16
2.2	Results of ABAS on Office31 with a ResNet-50 backbone. + EMS: experiments run with our model selection.	17
2.3	Results of ABAS on Office-Home, using a ResNet-50 backbone. Note we were not able to reproduce the official ALDA numbers using the official code, + EMS: experiments run with our model selection strategy.	18
2.4	Results of ABAS on PACS, using a ResNet-50 backbone, + EMS: experiments run with our model selection strategy.	18
3.1	Results in millimeters for ExPI Common actions split. Our model achieves state-of-the-art results in all actions considered, at each predicted time instant.	29
3.2	Results in millimeters for ExPI Unseen actions split. On average, we outperform the baseline considered over short and long time horizons.	29
3.3	List of actions and their corresponding names	30
3.4	Results in millimeters for ExPI Single actions split. We outperform in 6 out of 7 stocks all baselines considered according to the MPJPE metric. For the other stocks our model is comparable with the current state of the art.	31
3.5	Results in millimeters for ExPI Common actions split. Our model achieves state-of-the-art results in all actions considered, at each predicted time instant.	32
3.6	Results in millimeters for ExPI Unseen actions split. On average, we outperform the baseline considered over short and long time horizons.	32
3.7	Results in millimeters for ExPI Single actions split. We outperform in 6 out of 7 stocks all baselines considered according to the MPJPE metric. For the other stocks our model is comparable with the current state of the art.	32
3.8	Error in millimeters on Human3.6M dataset. We show how our method adapted to single-person human pose forecasting is comparable with the best-performing techniques on average.	34
3.9	Combinations of best practices. From left to right, we have frequency encoding, fully learnable connections, Space-time separability, initialization, attention mechanism, hierarchy, fully connected layer as a decoder. †: we implement a Graph Attention Network (GAT) tailored for GCNs, similar in spirit to [45] designed for transformers.	34
3.10	Initialization procedures for best practices model.	36
3.11	Results in millimeters on the ExPI dataset, on average common actions split. We show the impact that different combinations of input-output frames have on performance. Using 10 input frames makes predictions in the short term more accurate, helping results to be more stable in the long term.	36

4.1	Results on DSEC Semantic in UDA and open-vocabulary setting. OVOSE not only outperforms all the UDA methods even though they are trained in closed-set settings but also translated open-vocabulary methods.	45
4.2	Results on DDD17 in UDA and open-vocabulary setting. OVOSE outperforms all the UDA methods trained in closed-set settings but also translated open vocabulary methods	45
4.3	Ablation of distillation strategies of ABAS on DSEC Semantic dataset.	48
4.4	Comparison of OVOSE with state-of-the-art methods on DSEC Semantic dataset in a supervised setting. OVOSE outperforms state-of-the-art ESS[128] by 5.33% mIoU .	48
A.1	This table shows how the different metrics introduced in section 3.3 of the main paper correlate with the target accuracy. Note that all metrics, with the exception of source accuracy are computed on the target.	67

Chapter 1

Introduction

In the field of computer vision, Unsupervised Domain Adaptation (UDA) offers a powerful way to transfer knowledge between domains, especially when labeled data in the target domain is scarce. This makes UDA highly attractive for real-world applications, but current methods face two limitations: 1) Architectural Bottleneck: Adapting existing UDA methods to new neural architectures often requires slow and manual trial-and-error, hindering the adoption of advancements in deep learning research. 2) Target Label Dependency: Conventional Neural Architecture Search (NAS) approaches rely on target labels for validation, making them inapplicable to UDA settings where such labels are unavailable.

To address above limitations, we introduce a novel Adversarial Branch Architecture Search (ABAS), that unlocks new possibilities for UDA [112]. ABAS automates the search for the optimal architecture of the auxiliary adversarial branch, a crucial component in many UDA methods. It eliminates manual design and leverages the power of multi-fidelity Bayesian optimization to efficiently explore the search space and identify the best architecture and hyperparameters. By exploiting the pre-trained backbone architecture, ABAS further boosts performance. Furthermore, a novel data-driven method called Ensemble Model Selection (EMS) is introduced. EMS can select the best UDA model on the target domain, without requiring target labels. This innovative approach significantly improves the performance of both ABAS and existing UDA methods.

ABAS and EMS pave the way for more efficient and effective UDA. They promote faster adoption of new architectural advancements and improve transferability in real-world applications. This research holds significant potential to advance the field of UDA and unlock its full potential for practical applications.

This thesis further delves into the realm of regression, predicting the future poses of individuals engaged in interactions has garnered significant attention due to its wide-ranging applications. The task of two-body pose forecasting involves jointly predicting the future body poses of two individuals engaged in various activities, ranging from sports to collaborative industrial tasks. The ability to anticipate the movements of interacting pairs not only aids in long-term pose tracking but also contributes to understanding synergistic actions in domains such as dance choreography and human-robot collaboration in industrial settings. Despite its practical significance, the task of two-body pose forecasting has remained largely unexplored and constrained to limited datasets, hindering advancements in the field. While there has been notable progress in single-person pose forecasting, the translation of these advancements to the two-body counterpart has been lacking.

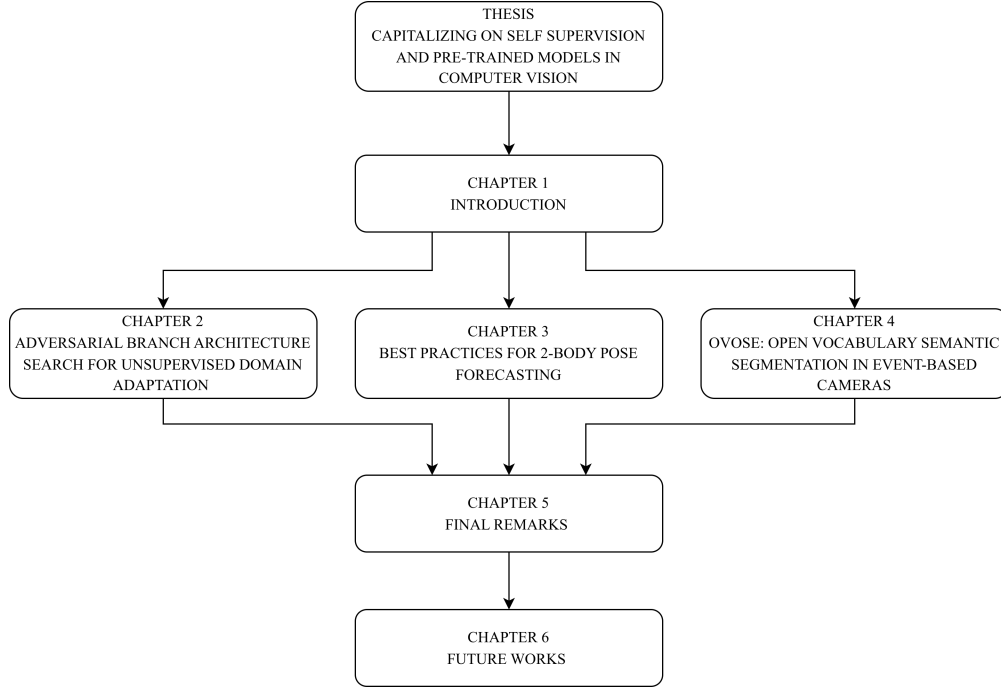


Figure 1.1: Outline of thesis in flow diagram

Existing techniques primarily focus on single-person methodologies, which overlook the intricate correlations in body-body motion crucial for accurately predicting interacting pairs' future poses.

Motivated by these challenges, we provide a systematic exploration of best practices in single-person skeleton-based pose forecasting to bootstrap research on two-body pose forecasting [107]. Leveraging recent modeling advancements, we propose a comprehensive analysis encompassing three key processing stages: input representations (Frequency encoding), encoding (Space-time separability, Learning the graph connectivity and weights, Attention and Hierarchical body parts modeling), and decoding (Convolutional-based decoder and FC-based decoder). Each stage is meticulously examined to identify and integrate the most effective techniques conducive to accurate and efficient two-body pose forecasting. Additionally, we introduce a novel initialization technique for learnable GCN parameters in the encoder, addressing the critical aspect of parameter initialization for performance, convergence speed, and model robustness.

The final model incorporating best practices demonstrates substantial improvements over existing techniques on EXPI dataset. Moreover, the thesis evaluates the generalization capabilities of the model across unseen actions and actors, establishing its robustness and versatility in real-world scenarios.

We conclude this work with an exploration of open vocabulary semantic segmentation in event cameras. In recent years, event cameras have emerged as a revolutionary technology in the domain of computer vision, offering unparalleled advantages in temporal resolution, low latency, and resistance to motion blur. These unique characteristics have propelled event cameras into various applications, ranging from autonomous driving to object recognition and semantic segmentation. Despite their remarkable performance under challenging conditions, the integration of event cameras into existing computer vision models presents significant challenges due to their distinct data format, which features asynchronous event streams rather than traditional image frames.

The significance of reliable visual data acquisition cannot be overstated, particularly in critical

applications such as autonomous vehicle operations. However, the scarcity of both raw and labeled data for event-based imagery poses a considerable obstacle to the effective integration of event cameras into computer vision models. This scarcity not only limits the ability to collect large-scale event datasets but also hampers the training of data-intensive deep learning techniques essential for robust semantic segmentation. In parallel, significant advancements have been made in traditional image-based semantic segmentation, propelled by the development of powerful foundation models. These models have demonstrated exceptional performance in closed-set semantic segmentation tasks, where the classes in the training and testing sets are consistent. However, directly applying these models to event-based imagery proves challenging, as event-based data lack the abundance of annotated examples necessary for effective model training.

In this context, we address the critical gap in the field of computer vision by introducing novel methodologies for open-vocabulary semantic segmentation tailored explicitly for event-based data. Our research builds upon the foundation of existing deep learning techniques while innovatively adapting them to the unique characteristics of event cameras. Specifically, we propose OVOSE, the first open-vocabulary semantic segmentation algorithm designed explicitly for event-based data. OVOSE operates as a two-branch network, with one branch dedicated to processing grayscale images and the other to handling event data. Leveraging recent advancements in text-to-image diffusion and Mask2Former models, OVOSE integrates a CLIP-style image encoder and MLP to derive embeddings for conditioning the text-to-image diffusion UNet. These embeddings guide the UNet’s features, which in turn influence Mask2Former for mask generation. Notably, to enhance model performance, we propose a knowledge distillation approach that transfers semantic insights from a foundation model trained on traditional RGB images to the event-based branch of OVOSE. Through extensive evaluations on DSEC-Semantic, DDD17 and TimeLens++ datasets, we demonstrate the superior performance of OVOSE compared to existing methods, including both closed-set segmentation models and straightforward adaptations of open-vocabulary models.

The outline of this thesis is presented in Figure 1.1 . Chapter 2 delves into the research on adversarial branch architecture search for unsupervised domain adaptation. Chapter 3 explores the study on 2-body human pose forecasting. Chapter 4 focuses on open vocabulary semantic segmentation in event-based cameras. The thesis concludes with final remarks and future directions in Chapters 5 and 6.

1.1 Publications

The following is an extensive list of research publications during doctoral studies, showcasing the work and contributions made in computer vision.

- Rahman, Muhammad Rameez Ur et al. “Best Practices for 2-Body Pose Forecasting.” 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) (2023): 3614-3624.
- Robbiano, Luc et al. “Adversarial Branch Architecture Search for Unsupervised Domain Adaptation.” 2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) (2021): 1008-1018.

1.2 Publications under review

The following is the list of publications currently under peer review.

- Rahman, Muhammad Rameez Ur et al. “OVOSE: Open-Vocabulary Semantic Segmentation in Event-Based Cameras.” (Under review in ICPR24)

Chapter 2

Adversarial Branch Architecture Search for Unsupervised Domain Adaptation

2.1 Overview

Unsupervised Domain Adaptation (UDA) enables the transfer of domain knowledge from a source domain to a target domain, for which possibly few data are available, but no target labels. Performance has increased steadily in recent years [19, 71, 88, 145, 151], but domain transfer remains constrained to specific setups [44, 57, 86]. In other words, whenever a new architectural backbone becomes available, optimally adapting existing UDA methods to it requires a manual and slow process of trial-and-error.

While Neural Architecture Search (NAS) has recently made large progress [82, 117, 129, 130] in removing the heuristic component of neural architecture design, it does not directly apply to UDA for two main reasons: **i.** NAS requires target labels for model validation when searching for the optimal architecture, which are not available in UDA; **ii.** the vast majority of UDA methods applied to visual recognition tasks, from classification to detection up to segmentation, require the use of pre-trained (backbone) models to reach state-of-the-art performance on real world datasets [19, 28, 145, 151]. To the best of our knowledge, no research has addressed these essential aspects in the context of NAS for UDA.

We propose a novel Adversarial Branch Architecture Search (ABAS - Fig. 2.1) for UDA to address both limitations. Our work builds on most recent and robust UDA methods which use *auxiliary adversarial branches* [13, 19, 37, 127]: a secondary network (the adversarial branch) is attached to a main backbone (see Fig. 2.2 for an illustration). While the main network is trained supervisedly on the source images and labels, the secondary branch reduces the distributional gap between the source and target domains, without target labels, by adversarially making the domains indistinguishable. The architecture of the auxiliary branch, which plays a huge role in the final accuracy of the method (as shown in Fig. 2.5), has so far been hand-crafted.

In ABAS, as first main contribution, we address the lack of target labels by a novel data-driven Ensemble approach for Model Selection (EMS), which directly applies to target data. Model selection in UDA is challenging and often overlooked. Only very few works have so far focused on it [100, 157], but they only consider source features. By contrast, considering the actual target features yields better domain transfers, as we show.

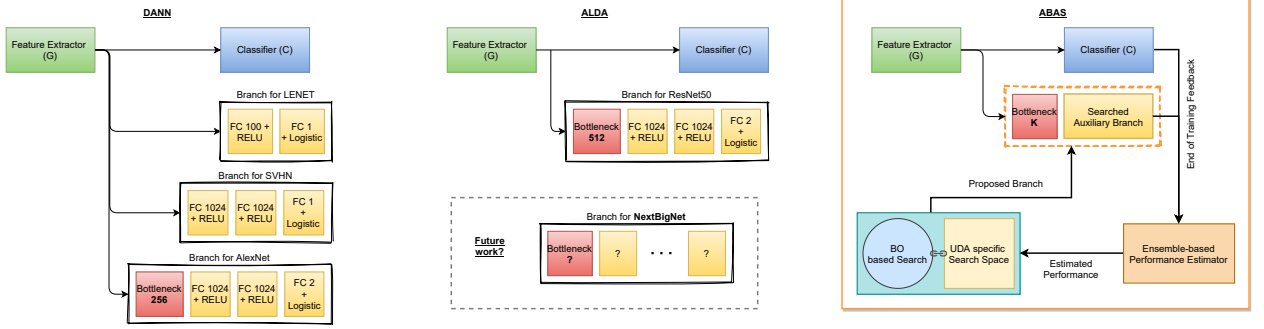


Figure 2.1: The original DANN [37] paper proposed a specific implementation for each of the considered backbone architectures. Later authors (e.g. ALDA [19]) empirically adapted the core idea to different architectures through a process of trial and error. Whenever a new architecture is proposed, this tedious procedure needs to be repeated (note this issue is shared by any other UDA method based on an auxiliary branch). To address this, in ABAS we propose to automatically search for the optimal architecture of the auxiliary branch in a data driven way. Through ABAS, it is possible to seamlessly apply DANN to any future architecture without the need for human supervision. What follows is a brief description, more details can be found in Algorithm 1. At each iteration, the Bayesian Optimization (BO) acquisition function proposes a number of candidate branch architectures. After training, we extract a number of supervised (source) and unsupervised (target) features which are fed into our ensemble-based performance estimator. Its feedback is sent to BO, thus closing the loop.

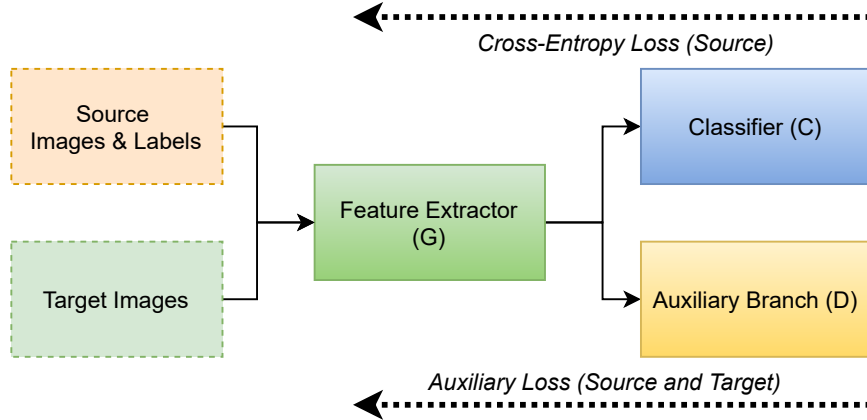
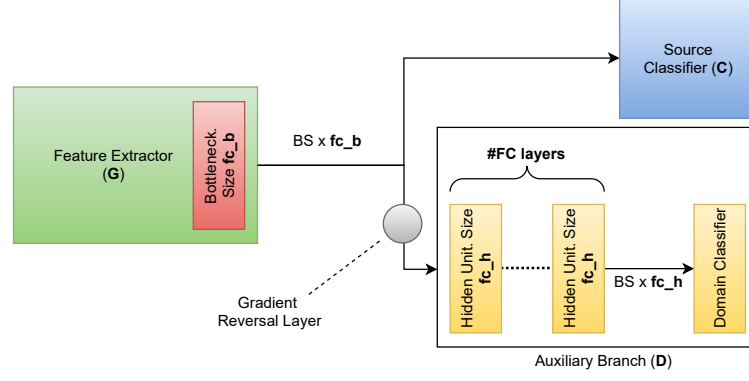


Figure 2.2: Many UDA methods fall under this general framework. The auxiliary branch D can compute either a domain discriminator [19, 37], or a self-supervised [13, 127] loss.

As a second main contribution we propose using a multi-fidelity Bayesian Optimization (BOHB) [33] method to search for both the architecture and the hyper-parameters of the auxiliary branch.

ABAS exploits the backbone architecture and pre-training, which is key to surpass the current state-of-the-art performance.

We conduct an in-depth performance evaluation on a popular UDA technique, DANN [37], and on a more recent state-of-the-art method, ALDA [19]. We test their ABAS extensions on three computer vision datasets, PACS [70], Office31 [119] and Office-Home [136]. Our study confirms that the default DANN and ALDA branches perform well on specific backbones and source-target cases, i.e. for Office31 and Office-Home, well above random architectures (cf. Fig. 2.5 and Sec. 2.4). However, they underperform on PACS, where ABAS also finds the best UDA architecture.



Hyperparameters	Ranges	Type
Domain adaptation weight λ	$[0, 2]$	real
Dropout Probability	$[0, 1]$	real
# FC layers	$[1, 6]$	integer
Bottleneck size fc_b	$[2^6, 2^{10}]$	integer
Hidden units size fc_h	$[2^6, 2^{12}]$	integer

Figure 2.3: Top: overview of how the ABAS search space applies to the main architecture (BS is batch size). Bottom: explicit search space definition. Being low-dimensional it can easily be optimized through BO; on the other hand (as shown in Fig. 2.9) each of these parameters has a strong effect on the final accuracy.

2.2 Related Work

Unsupervised Domain Adaptation (UDA). Research on UDA has been recently divided into four families of methods [166]: discrepancy, adversarial discriminative, adversarial generative and self-supervision based methods. Adversarial discriminative approaches provide top performance, but they are significantly influenced by the specific architecture of the auxiliary branch, cf. Fig. 2.5: un-optimized architectures may diverge and/or yield a wide span of accuracies. We find this strongly motivates research on architecture search for these approaches.

Starting from [37], UDA has been cast as the problem of learning domain-agnostic features via an auxiliary branch and an adversarial training. The extra branch is attached to the backbone and tasked to learn to discriminate the domains. Then its gradient is reversed to make the backbone features indistinguishable between the source and the target domain. A few recent improvements are notable: [52] considers the cross-covariance between samples; [83] takes the classifier prediction into account; [133] generalizes a similar approach to the semi-supervised setting; [14] extends the discriminative gradient reversal to the pixel level; and finally [19] combines the adversarial training with self-training.

All of the above propose specific embodiments of the auxiliary adversarial branch, but offer no guidelines on how to adapt it to different backbones. To the best of our knowledge, this is the first work proposing a data-driven pipeline to learn it.

Model Selection for UDA. Hyper-parameter and model selection are extremely challenging in the UDA setting, as target labels are not available. In practice, many works simply fix the hyper-parameters across a range of datasets [105, 118, 121, 134].

A more principled approach based on importance-weighted cross-validation (IWCV) was first proposed by [126]: the more the source sample is similar to the target, the higher the importance. The IWCV estimator is unbiased, but has unbounded variance; Deep Embedded Validation [157],

was proposed to reduce the variance and adapt IWCV to deep features. Similarly, [100] proposes a reduced-variance IWCV estimator for hyper-parameter optimization in the context of multi-source DA. The current literature focuses on the source, only implicitly taking target features into consideration. By contrast, we propose a performance estimator which mostly uses the actual target features (see Sec. 2.3.3).

Neural Architecture Search (NAS) and UDA-NAS. There is a wealth of literature on NAS [169] leveraging diverse search strategies: Bayesian Optimization (BO), evolutionary, reinforcement learning, and gradient-based, as recently surveyed in [32, 111]. BO is one of the most efficient strategies for sample-based hyper-parameter tuning [124], and has been recently applied to NAS [143, 156]. Multi-fidelity BO (BOHB) [33], which further reduces the search cost by performing partial evaluations, has also been successfully applied to NAS [117, 159]. ABAS exploits pre-trained backbones and efficiently fine-tunes the architectures, enabling us to use a more accurate sample-based technique. In particular, we adopt BOHB for its sota performance.

A recent pre-print [77] has proposed a DARTS-like [82] method for UDA. While that shares similar motivations with ABAS, it is unable to leverage pre-trained models and as such it can only be applied to simple low-resolution datasets. Furthermore model selection is performed by a simple validation on the source domain, which we show to be insufficient in the general case.

2.3 Method

We propose a data-driven pipeline to search for the optimal architecture of the auxiliary branch in the context of unsupervised domain adaptation. As mentioned in section 2.2, a number of UDA approaches require the use of an auxiliary branch: while we explicitly explore how this idea applies to works based on domain adversarial learning, it is trivially extendable to other methods (see section 2.5). Here we apply ABAS to 1) the first UDA adversarial approach, DANN [37] and to 2) the current state-of-the-art variation, ALDA [19]. In both cases, the auxiliary branch is adversarially trained to reduce the distributional gap at the feature level.

2.3.1 Background

Due to space constraints, we provide here only a mostly intuitive overview and refer to the referenced papers for a formal presentation of background methods.

Adversarial branches for domain adaptation Let us define a dataset $\mathbf{X}_s = \{\mathbf{x}_s^i, y_s^i\}_{i=0}^{N_s}$ drawn from a labeled source domain $\mathcal{S}$, and a dataset $\mathbf{X}_t = \{\mathbf{x}_t^j\}_{j=0}^{N_t}$ from a different unlabeled target domain $\mathcal{T}$, sharing the same set of categories. Our goal is to maximize the classification accuracy on $\mathbf{X}_t$ while training on $\mathbf{X}_s$. As it can be seen in Figure 2.2, the overall architecture consists of 3 components: the feature extractor G , the labeled classifier C and the adversarial branch D . The basic intuition behind discriminative adversarial approaches for domain adaptation is that a domain classifier is trained to distinguish between source and target samples; as its gradient is reversed (Fig. 2.3), the features in G are trained so that source and target have similar representations. The model is trained by optimizing the following objective function:

$$\min_{G, C, D} \mathcal{L}_{CE}(C(G(x_s)), y_s) + \lambda \mathcal{L}_{Adv}(D(G(x_s \cup x_t))) \quad (2.1)$$

Where $\mathcal{L}_{CE}$ is the standard cross-entropy loss, computed on source samples only, and $\mathcal{L}_{Adv}$ is the adversarial loss trained to reduce the domain gap. In the case of DANN [37],

$$\mathcal{L}_{Adv}(D(G(x_i)), d_i) = d_i \frac{1}{D(G(x_i))} + (1 - d_i) \frac{1}{1 - D(G(x_i))} \quad (2.2)$$

With d_i being a binary variable (domain label) indicating whether the sample i belongs to the source or to the target.

ALDA [19], is a modern DANN variant, currently state-of-the-art, which combines domain adversarial training with self-training. While its overall loss contains multiple terms, we focus here only on the adversarial loss:

$$\mathcal{L}_{Adv}(x_s, y_s, x_t) = \mathcal{L}_{BCE}(c^{(x_s)}, y_s) + \mathcal{L}_{BCE}(c^{(x_t)}, u^{(\hat{y}_t)}) \quad (2.3)$$

L_{BCE} is the Binary Cross-Entropy loss, $c^{(x)}$ is the corrected pseudo-label vector generated by multiplying the confusion matrix and the pseudo-label vector $\hat{y}$, and $u^{(\hat{y}_t)}$ is the opposite distribution of $\hat{y}_t$. Intuitively, the authors propose to use a discriminator branch to predict the confusion matrix instead of the domain probabilities. The confusion matrix is then used to correct the pseudo-label vector obtained by feeding target samples into the network, which is then used as the target label for the training process. The discriminator is trained to produce different corrected vectors for each domain, while the feature extractor learns to fool it. ABAS will be used to search for the best performing architecture for D in both DANN and ALDA.

Bayesian Optimization for AutoML Several optimization strategies could be used to guide the search, but our search space is compact and has low dimensionality (cf. Sec. 2.3.2), which makes it particularly suitable for Bayesian Optimization (BO). For efficiency, we adopt BOHB, a multi-fidelity combination of BO and Hyperband [73]. While single fidelity BO evaluates all samples with full budget, BOHB resorts to partial evaluations with smaller-than-full budget, excluding bad configurations early in the search process and reserving computational resources for the most promising configurations. So, given the same time budget, it evaluates many more configurations and it achieves faster optimization than competing methods. Details of the BOHB’s role in ABAS are reported in Sec. 2.3.4

2.3.2 ABAS Search space

UDA methods for visual object categorization require the use of a pre-trained model (typically on ImageNet) to be competitive. Hence, the architecture of G and C are usually fixed as a subset of the backbone: in practice the only choice being made is where G ends and C begins (see Fig. 2.2). Our search space (Fig. 2.3) is a combination of hyper and architectural parameters: the domain loss weight λ (eq. 2.1) and the dropout probability of the fully connected layers in D are hyperparameters, while the number of fully connected layers ($\#fc$ layers), size of the hidden units (fc_h) and of the bottleneck (fc_b) fully characterize the width and the height of the auxiliary branch D . Giving the bottleneck layer a separate width allows to control the overall capacity of the auxiliary branch by limiting how much information it is fed.

2.3.3 Ensemble-based Model Selection (EMS)

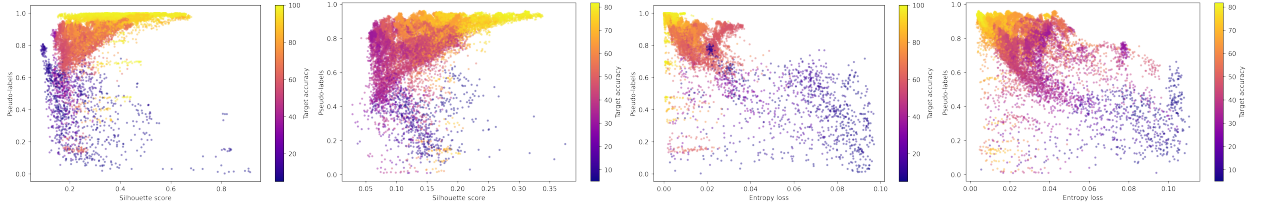


Figure 2.4: Different model configurations trained on Office31 (1st and 3rd) and Office-Home (2nd and 4th). The x and y axes show two different unsupervised validation metrics we use in EMS, while the color represents target accuracy. Clearly, by only using a single metric it is extremely hard to uniquely predict the performance, but the combined use of two metrics makes it easier to distinguish between good and bad models. This shows how different metrics can be complementary and employed to build a better estimator of model performance. The similarity between pairs of plots on different settings (1&3 and 2&4) suggests that an estimator built on one dataset can be successfully transferred on another. Further plots can be found in the Supplementary.

Model selection in the context of UDA is extremely challenging, as target labels are not available. To overcome this, we propose using an ensemble of weakly correlating predictors. Experimentally, we show how a linear regressor over these metrics, trained on one dataset, is capable of capturing the ranking between models on a *different* unseen dataset, thus highlighting the generalization power of this approach.

A description of the considered metrics follows ([S] or [T] implies on which domain the metric is computed). 1. **Entropy** [T] [43] minimization is commonly used in UDA methods as a regularizer for unlabeled samples [116]; intuitively a low entropy corresponds to well separated classes. Mathematically the target entropy is defined as $H(C(G(X_T))) = -\frac{1}{N_t} \sum_{x^i \in X_T} C(G(x^i)) \log(C(G(x^i)))$. 2. **Diversity** [T], recently proposed by [144], measures the diversity of class predictions in a batch. It is defined as the entropy of the predicted category distributions, $H(\hat{q}(\mathcal{T}))$. 3. **Silhouette and Calinski-Harabasz score** [T] [12, 115]: the most confident label is assigned to the target samples, which are then clustered with K-Means using the predicted class centroids. The Silhouette and Calinski score can then be used to estimate how well separated are the different classes. 4. **Source (weighted) loss** [S]: the cross-entropy loss can be computed on the labeled source samples to estimate the performance of the predictor. Additionally it is possible to weight the samples according to their similarity to the target, as in Importance Weighted Validation [126, 157]. We initially experimented with the source-weighted accuracy in EMS, but due to a number of practical drawbacks we decided to adopt the vanilla source accuracy instead. Specifically, the performance of Importance Weighting methods heavily depends on how well the regression model estimates the density ratio, which is not trivial to successfully implement; secondly, it requires a separate source validation set (not always available on small settings such as Office31). 5. **Time consistent pseudo-labels** [T]: inspired by [167], we propose to use the mode of all model predictions during training (one prediction per epoch) as target pseudo-labels, as the label predicted more often is likely to be the true one. Once training has ended, we compute these pseudo-labels and use them to retroactively estimate the target accuracy at each epoch.

Note how all of these metrics, with the exception of the source accuracy, are computed on the target domain. These metrics are all relatively cheap to compute and have good correlation on their own with the target accuracy, but as we show in Fig. 2.4 and in the Supplementary, they are all

Algorithm 1 BO auxiliary branch optimization

```

1: Input: Domain adaptation method  $Q$ , Performance estimator  $E$ , BO surrogate model
    $p(f|\Theta, D)$  and acquisition function  $\alpha(\Theta|D)$ 
2: for  $t = 1$  to  $T$  do
3:   Recommend  $\{\Theta_t^j\}_{j=1}^B = \arg \max \alpha_{t-1}(\Theta|D)$ 
4:   for  $j = 1$  to  $B$  in parallel do
5:     Build the auxiliary branch and evaluate  $E(Q(\Theta_t^j))$  to obtain its corresponding performance
       metric  $f_t^j$ 
6:   end for
7:   Update  $D$  and thus  $p(f|\Theta, D)$  with  $\{\Theta_t^j, f_t^j\}_{j=1}^B$ 
8: end for
9: return The best model according to  $E(Q(\Theta^*))$ 

```

individually vulnerable to degenerate cases. To overcome this, we learn a simple least-squares linear regressor to predict the target accuracy. Since it is not possible to use the target labels of interest we instead learn the regressor on one experimental setting and then use it on a different one. It is worth mentioning that although the output of the regressor cannot be used to estimate the *accuracy* of a model on an unseen dataset, its output has a strong correlation with the ground-truth ranking, as we show in section 2.4.1. We use this predictor to not only return the best sample, but also the best snapshot during training.

2.3.4 Auxiliary branch optimization

A full overview can be found in Fig. 2.1 and Algorithm 1: ABAS combines BOHB [33] and an adversarial method of choice (sec. 2.3.1), with our UDA specific search space and model selection strategy. Given a fixed budget, ABAS performs a number of rounds, alternating between sampling and evaluating. At each step, we sample the Bayesian acquisition function $\alpha(\Theta|D)$ for B different configurations $\{\Theta_t^j\}_{j=1}^B$. The configurations are used to build auxiliary branches for method Q and the resulting network is trained on the target setting. After training, supervised (from the source) and unsupervised (from the target) features are collated for our model selection module, as described in section 2.3.3. The ensemble predictor finally gives feedback to the BO process. This procedure is repeated for a given number of rounds and in the end the best performing model is returned. The pipeline itself is relatively simple and, as shown in the next section, capable of significantly improving over existing sota methods. Finally, while we decided to use BOHB in this particular implementation, it would be feasible to use a different search strategy.

2.4 Experiments

Datasets. We evaluate our method on three publicly available multi-domain datasets. The **Office31** dataset [119], composed of 4,652 images, is a standard benchmark for domain adaptation. As shown in Fig. 2.6, the images depict objects belonging to 31 categories and taken from three different domains: Amazon (A), DSLR (D), and Webcam (W). Webcam and DSLR pictures were manually captured in the same office environment with a consumer webcam and a reflex camera. Amazon samples are images of products downloaded from amazon.com.

The **Office-Home** dataset [136] contains objects from 65 categories and a total of around

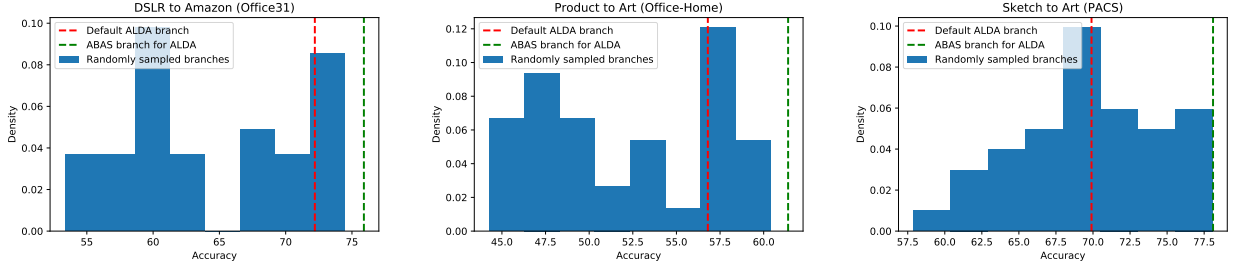


Figure 2.5: Accuracy histograms obtained by randomly sampling 40 architectures for ALDA’s [19] auxiliary branch and training them on three settings (10% didn’t converge and were excluded). Contrary to many NAS search spaces [154], ours is large enough to offer a significant impact on final performance (around 20%) on all settings. Equally important, while the default ALDA branch performs well on Office31 and Office-Home (the datasets it was tested on), it performs significantly worse on PACS. ABAS, on the other hand, can adapt itself to all settings, consistently outperforming the random sampling baseline. See sec. 2.4.1 for full details.



Figure 2.6: Samples from office31 dataset. [50, 119]

15,500 images from four domains: Art, Clipart, Product, and Real-World. Like Office31’s Amazon domain, the Product domain features product pictures taken from websites. Real-world images were captured with a regular digital camera. Sample from Office-Home dataset are shown in Fig. 2.7.

PACS [70] is composed of 9,991 images from four domains: Art paint, Cartoon, Sketch and Photo, each containing objects from 7 categories. While PACS is commonly used for domain generalization, we employ it in our single-source setting to explore a diverse test-bed. Samples from PACS are shown in Fig. 2.8.

Baseline methods To highlight the improvements brought by our ABAS pipeline, we apply it to two representative UDA methods: DANN [37], the original domain adversarial approach, still competitive to this day and base on which all methods of this family build on, and ALDA [19] the current state-of-art among discriminative adversarial approaches.

Baseline implementation For all our baselines, we use a ResNet-50 backbone pre-trained on ImageNet. The DANN and ALDA baselines use the same adversarial branch architecture (two 1024



Figure 2.7: Samples from Office-Home dataset. [136]

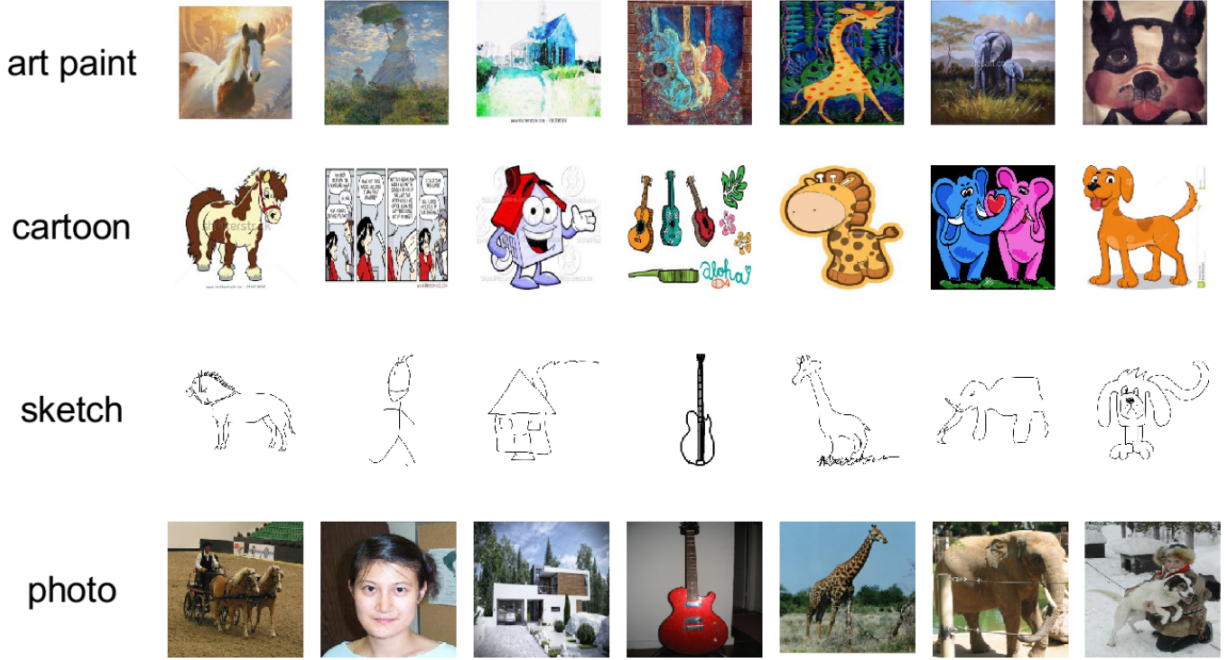


Figure 2.8: Samples from PACS dataset. [70, 150]

fully connected layers), connected right after a 512-sized bottleneck layer preceding the ResNet-50 output. The initial learning rate is set to 0.001, and it is adjusted during training following

$$\mu_p = \frac{\mu_0}{(1 + \alpha \cdot p)^\beta}, \quad (2.4)$$

where $\alpha = 10$, $\beta = 0.75$ and p grows from 0 to 1 during training. The weight decay is 0.0005, and the network is trained with SGD (momentum 0.9, batch size 36). Since the bottleneck and adversarial branch are trained from scratch, for those parts of the network we set the initial learning rate to 0.01 and the weight decay to 0.001. For each run (no EMS) we select the last two snapshots (evaluated each 100 iterations) and average them. For the EMS baselines we select the best snapshot of the run according to the value predicted by the EMS regressor. Each run was repeated 4 times and the final accuracies were averaged.

Training protocol ABAS is based on an outer BOHB loop alternated with an inner loop in which different networks are trained. For BOHB, we set $\eta = 3$, minimum budget 2000 iterations, maximum budget 6000 iterations and run it for 24 iterations. Intuitively, the most promising solutions evaluated at a lower budget will be re-run at a higher budget. For each sample, BOHB

indicates a computational budget and an architectural configuration, which are used to build the appropriate auxiliary branch in the context of the adversarial method of choice.

In all experiments we use a ResNet50 [49] backbone pretrained on ImageNet; the adversarial branch (built according to each specific sample, see sec. 2.3.2) is located immediately after a bottleneck layer preceding the final classification layer (Fig. 2.3). Each model is trained end to end, using the same training code and protocol as [19]: on all settings, we use SGD optimizer, batch size 36, momentum 0.9, weight decay 0.0005 and initial learning rate of 0.001. We adjust the learning rate by $\mu_p = \frac{\mu_0}{(1+\alpha \cdot p)^\beta}$, where p is the training progress linearly changing from 0 to 1, $\mu_0 = 0.001$, $\alpha = 10$ and $\beta = 0.75$, following [37], the weight of the gradient reversal layer is gradually increased from 0 to 1: $\rho = \frac{2}{(1+\exp(-\gamma \cdot p))} - 1$ where γ is set to 10 for all experiments. Since the bottleneck and the auxiliary branch are trained from scratch, we use a higher learning rate ($10\mu_p$) and weight decay (0.001) for those parts of the network. At each iteration, a source and target batch are fed into the backbone. The cross-entropy loss is computed on the source, while both source and target samples contribute to the adversarial branch loss (DANN or ALDA loss, depending on the setting). At the end of training, both source and target features are then fed into the performance estimator (as detailed in section 2.3.3) to provide feedback to the Bayesian Optimization process.

To train our regressors, we randomly sampled 200 configurations on each dataset and trained them while collecting 100 snapshots for each run (total of 20,000 data points). At each snapshot we collected the metrics described in sec. 2.3.2 thus building a dataset to train our regressor on; note that we only use the pseudo-label metric to assess the best snapshot in a single run, not to compare across runs. All input features were standardized to zero mean and unit standard deviation. For the regressor we used a linear least-squares model as implemented in Scikit-Learn [102]. On Office31 we used a regressor trained on Office-Home and vice-versa. The PACS regressor was trained on Office31.

Software and hardware Our network training code is written in PyTorch and based on the publicly available ALDA [19] repository, likewise, we use HpBandSter, the official BOHB [33] implementation. The BOHB algorithm is run for 24 iterations using 8 parallel workers, each running on a Tesla V100 GPU. On average, it takes 6, 10 and 6.4 hours to run the full ABAS pipeline on PACS, Office-Home and Office31 respectively, leading to a cost of 80 GPU hours on the largest setting.

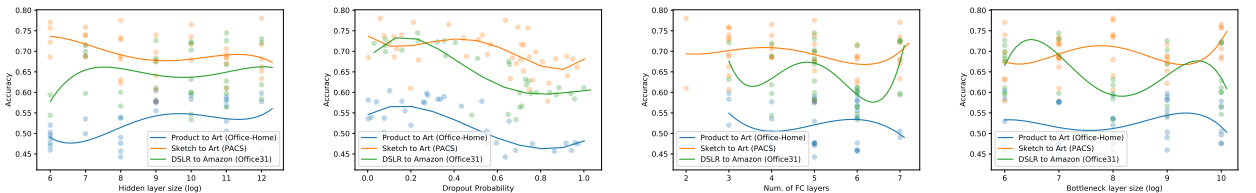


Figure 2.9: Search space sensitivity analysis. We randomly sampled 40 configurations and trained them on a representative setting for each of the considered datasets (Office31, Office-Home, PACS). It is evident that the optimal value changes significantly across tasks. X-axis represents the search space hyper-parameter with respect to which the plot is computed.

2.4.1 Preliminary analysis

Search space expressivity and sensitivity. Given the added cost of automatically searching for the optimal architecture, it is important to understand whether 1) the design of the adversarial branch has an impact on the final accuracy and 2) if a "good" architecture is actually data-dependent. To assess this, we followed NAS best practices [154] and randomly sampled 40 configurations on each setting to evaluate our search space. We observed the following: the design of the adversarial branch can make or break a training. Even excluding a significant (10%) portion of models which failed to converge, on average the design of the adversarial branch accounted for at least 20% in absolute performance (Fig. 2.5 - full results in the supplementary) - note that λ was fixed to focus on the architecture only. Furthermore, as shown in Fig. 2.9, the "best" configuration heavily depends on the specific setting being considered. Indeed, while the adversarial branch proposed in ALDA performs well on Office31 and Office-Home (the datasets it was originally tested on), it significantly lags behind on several PACS settings (Fig. 2.5 and Table 2.4).

Performance estimation. As discussed in sec. 2.2, and recently highlighted by [44], strong model selection should be a key element of any transfer learning approach; more so when proposing a UDA method capable of explicitly tuning the model hyper-parameters themselves. In our next experiments we analyzed how well different metrics correlate with the target accuracy. We sampled 40 configuration on each setting and computed the metrics discussed in section 2.3.3 at each epoch of training. Full overview on the results can be found in the supplementary, while representative plots are shown in Fig. 2.4. Note how, individually, all metrics perform as weak predictors, correlating with the target accuracy but failing in specific instances. By considering a linear combination we provide a much stronger estimate.

To assess our model selection strategy in isolation from the NAS component, we retrained vanilla ALDA [19] and reported the performance according to either the last epoch (as done in the paper) or according to our model selection (Tables 2.2,2.3,2.4). It is worth mentioning that a) we could not replicate their results (even though we used the official code) and b) on Office-Home and PACS our model selection improved average results by up to 2 full p.p.

2.4.2 Comparison with SOTA

Comparison with current sota methods can be found in tables 2.2 to 2.4. ABAS significantly improves over the UDA methods it builds on (even when they use our same model selection strategy), with both DANN and ALDA and on all three datasets. The smallest average improvement we observe is 1.9% (Office31 - ALDA 87.5 vs ABAS-ALDA 89.4) while the largest is 3.1% (Office-Home - DANN 56.0 vs ABAS-DANN 59.1). With the exception of Office-Home, where we couldn't replicate the official ALDA results, ABAS-ALDA is the new sota on all settings. It is worth mentioning that on one of the easiest Office31 settings (D-W), ABAS-ALDA is outperformed by a few different methods; we speculate that, since DSLR and Webcam are very close, the adversarial branch fails to capture meaningful differences. Indeed, ALDA similarly performs poorly on this setting.

2.4.3 Impact of the auxiliary branch configuration on the final accuracy

As can be seen in Table 2.1, the design choices which are part of our search space have a tremendous impact on the final performance of an ALDA [19] training - in other words, the architecture and the hyper-parameters of the auxiliary branch can make or break the UDA approach that builds on it.

Setting	% Div.	Min	Max	Mean $\pm$ Std.	Range
A-W	0.23	64.59	95.79	83.4 ± 09.8	31.19
A-D	0.28	72.49	91.57	84.4 ± 06.2	19.08
D-W	0.15	17.11	99.06	89.0 ± 21.7	81.95
D-A	0.28	46.59	70.15	59.3 ± 06.4	23.55
W-D	0.22	16.47	100.00	89.3 ± 24.2	83.53
W-A	0.17	34.97	74.55	63.8 ± 08.3	39.58
Ar-Cl	0.08	26.60	50.71	41.7 ± 06.4	24.11
Ar-Pr	0.14	59.12	65.14	63.2 ± 02.1	06.01
Ar-Rw	0.22	43.30	75.89	70.0 ± 06.3	32.59
Cl-Ar	0.14	36.55	57.87	51.6 ± 07.3	21.32
Cl-Pr	0.14	55.58	66.41	60.9 ± 03.9	10.84
Cl-Rw	0.15	46.12	70.25	63.2 ± 05.6	24.13
Pr-Ar	0.14	40.15	57.21	49.8 ± 04.9	17.06
Pr-Cl	0.11	26.27	49.52	39.7 ± 06.5	23.25
Pr-Rw	0.07	41.47	75.64	66.5 ± 10.7	34.16
Rw-Ar	0.07	59.81	67.86	64.7 ± 02.3	08.06
Rw-Cl	0.07	15.40	54.80	46.2 ± 07.6	39.40
Rw-Pr	0.14	70.30	81.36	77.1 ± 02.6	11.06
AVG	0.2	42.9	72.4	64.6 ± 7.9	29.49

Table 2.1: Over 400 different points were randomly sampled from our search space and the corresponding models were then trained using ALDA [19] over either Office31 [119] (top 6 rows) or Office-Home [136] settings. Note how the configuration of the auxiliary branch significantly affects the model performance. *% Div.* stands for the fraction of runs which did not converge - *Min* is the minimum accuracy - *Max* is the maximum - *Mean* reports the mean $\pm$ the standard deviation, and *Range* is the total range of accuracies, defined as $Max - Min$.

2.5 Discussion

The shift from hand-crafted to principled architecture search is ongoing, as it is witnessed by the flourishing NAS research and techniques. ABAS fills in two important requirements of NAS for UDA: it provides a data-driven strategy (EMS) for model selection, circumventing the lack of target labels; and it focuses on searching the architecture of auxiliary branches attached to a pre-trained backbone, essential practise for state-of-the-art performance.

Only most recent research has stressed the importance of model selection for UDA [44]. The lack of a validation identically distributed as the test set, has led researchers to indulge on selecting hyper-parameters on the test set. Hopefully the propositions here (cf. EMS) would be initial steps towards more methodologically sound model selection practises.

Finally, we emphasize that ABAS provides tools (search space, EMS, branch optimization),

	A-W	A-D	D-W	D-A	W-D	W-A	AVG
ResNet-50 [49]	68.4±0.2	68.9±0.2	96.7±0.1	62.5±0.3	99.3±0.1	60.7±0.3	76.1
DANN [37]	82.0±0.4	79.7±0.4	96.9±0.2	68.2±0.4	99.1±0.1	67.4±0.5	82.2
ADDA [134]	86.2±0.5	77.8±0.3	96.2±0.3	69.5±0.4	98.4±0.3	68.9±0.5	82.9
JAN [85]	85.4±0.3	84.7±0.3	97.4±0.2	68.6±0.3	99.8±0.2	70.0±0.4	84.3
MADA [103]	90.0±0.1	87.8±0.2	97.4±0.1	70.3±0.3	99.6±0.1	66.4±0.3	85.2
CBST [170]	87.8±0.8	86.5±1.0	98.5±0.1	71.2±0.4	100±0.0	70.9±0.7	85.8
CAN [162]	92.5	90.1	98.8	72.1	100.0	69.9	87.2
CDAN+E [84]	94.1±0.1	92.9±0.2	98.6±0.1	71.0±0.3	100.0±0.0	69.3±0.3	87.7
MCS [79]	75.1	71.9	96.7	58.8	99.4	57.2	76.5
DMRL [145]	90.8±0.3	93.4±0.5	99.0±0.2	73.0±0.3	100.0±0.0	71.2±0.3	87.9
DM-ADA [151]	83.9±0.4	77.5±0.2	99.8±0.1	64.6±0.4	99.9±0.1	64.0±0.5	81.6
3CATN [71]	95.3±0.2	94.1±0.3	99.3±0.5	73.1±0.2	100±0.0	71.5±0.6	88.9
ALDA [19]	95.6±0.5	94.0±0.4	97.7±0.1	72.2±0.4	100.0±0.0	72.5±0.2	88.7
ResNet-50	67.0±1.9	76.4±1.4	94.8±1.3	56.2±1.0	98.7±0.1	58.3±0.9	75.2
ResNet-50 + EMS	70.4±2.0	77.1±2.0	95.6±0.6	58.2±2.6	99.1±0.5	60.1±1.1	76.8
DANN	85.0±0.5	82.5±0.5	96.7±0.2	63.9±1.1	99.2±0.3	64.7±0.7	82.0
DANN + EMS	85.5±2.1	82.8±2.2	97.1±1.0	64.1±0.5	99.8±0.2	64.5±2.0	82.3
ABAS-DANN	89.4	87.6	98.4	64.1	99.8	69.3	84.8
ALDA	94.8±0.6	91.6±0.9	98.3±0.4	69.6±0.9	99.9±0.0	70.8±0.7	87.5
ALDA + EMS	95.0±1.4	91.9±0.7	98.6±0.4	69.3±2.1	99.7±0.3	71.1±0.8	87.6
ABAS-ALDA	96.1	95.0	98.5	75.9	100.0	70.7	89.4

Table 2.2: Results of ABAS on Office31 with a ResNet-50 backbone. + EMS: experiments run with our model selection.

which comply with the use of most other NAS methods and potentially all UDA techniques based on an auxiliary branch. A notable example is self-supervision-based UDA methods using an auxiliary branch. Since they adopt the same exact macro-architecture as in Fig. 2.2, then ABAS seamlessly applies there.

	Ar-Cl	Ar-Pr	Ar-Rw	Cl-Ar	Cl-Pr	Cl-Rw	Pr-Ar	Pr-Cl	Pr-Rw	Rw-Ar	Rw-Cl	Rw-Pr	AVG
ResNet-50 [49]	34.9	50.0	58.0	37.4	41.9	46.2	38.5	31.2	60.4	53.9	41.2	59.9	46.1
DANN [37]	45.6	59.3	70.1	47.0	58.5	60.9	46.1	43.7	68.5	63.2	51.8	76.8	57.6
JAN [85]	45.9	61.2	68.9	50.4	59.7	61.0	45.8	43.4	70.3	63.9	52.4	76.8	58.3
CDAN+E [84]	50.7	70.6	76.0	57.6	70.0	70.0	57.4	50.9	77.3	70.9	56.7	81.6	65.8
TAT [81]	51.6	69.5	75.4	59.4	69.5	68.6	59.5	50.5	76.8	70.9	56.6	81.6	65.8
ALDA [19]	53.7	70.1	76.4	60.2	72.6	71.5	56.8	51.9	77.1	70.2	56.3	82.1	66.6
ResNet-50	35.6	61.5	70.4	44.0	56.1	57.4	46.5	31.9	69.1	62.4	38.0	75.6	54.1
ResNet-50 + EMS	36.5	63.1	71.7	45.8	57.5	59.3	47.9	32.3	69.9	63.5	37.3	76.0	55.1
DANN	39.8	58.0	68.1	48.6	57.0	59.9	46.9	38.4	68.8	63.2	47.7	75.9	56.0
DANN + EMS	40.3	60.4	70.1	49.6	57.5	60.5	47.9	38.3	69.1	63.7	47.7	76.8	56.8
ABAS-DANN	44.9	61.1	71.2	52.7	60.4	62.5	50.1	43.1	70.0	65.4	50.9	77.1	59.1
ALDA	46.4	68.6	74.6	57.6	67.0	69.4	57.2	46.3	75.6	69.2	53.3	80.8	63.8
ALDA + EMS	47.5	70.1	75.2	58.8	67.5	69.7	58.1	46.6	76.2	69.9	54.0	81.0	64.6
ABAS-ALDA	51.5	71.7	75.5	59.8	69.4	69.5	59.8	47.1	77.7	70.6	55.2	80.2	65.7

Table 2.3: Results of ABAS on Office-Home, using a ResNet-50 backbone. Note we were not able to reproduce the official ALDA numbers using the official code, + EMS: experiments run with our model selection strategy.

	P-A	C-A	S-A	A-P	C-P	S-P	A-C	S-C	P-C	A-S	C-S	P-S	AVG
ResNet-50 [49]	62.3	71.2	30.8	96.6	89.7	38.0	57.0	53.5	27.6	43.0	59.1	30.7	55.0
ResNet-50 + EMS	62.0	72.0	25.6	97.7	90.9	38.8	57.5	53.2	23.0	41.3	58.7	28.2	54.1
DANN [37]	79.9	88.4	70.3	97.3	95.1	57.1	82.4	70.8	64.7	62.5	68.7	61.4	74.9
DANN + EMS	80.3	89.0	72.6	98.1	95.6	56.2	82.2	73.3	64.0	63.4	69.7	59.6	75.3
ABAS-DANN	91.7	90.7	68.2	98.3	96.2	82.8	87.2	71.2	73.7	68.7	76.0	47.2	82.9
ALDA [19]	89.3	91.9	69.9	98.3	97.3	63.4	85.1	75.2	74.3	79.2	70.6	60.7	79.6
ALDA + EMS	90.2	92.0	72.3	98.4	97.8	69.5	86.0	82.1	72.1	80.7	75.1	66.1	81.9
ABAS-ALDA	93.1	91.8	78.1	98.7	97.8	70.8	88.7	84.9	69.7	79.8	69.5	64.9	82.3

Table 2.4: Results of ABAS on PACS, using a ResNet-50 backbone, + EMS: experiments run with our model selection strategy.

Chapter 3

Best Practices for 2-body Pose Forecasting

3.1 Overview

Human 2-body pose forecasting predicts the future body poses of two people in interaction jointly. The task is relevant to long-term pose tracking [4], to understanding interacting pairs in sports such as dancing [45] and to the collaborative assembly in industry [31, 64], towards human-robot collaboration [122]. Considering the concurrent prediction of two bodies helps in cases where the people act synergistically. However, this task has remained mostly unexplored and limited to the dataset of [45]¹. Also, this differs from the related task of human trajectory forecasting, where social interaction has been key to most recent progress [65, 97, 98, 158].

There has been vast progress in single human pose forecasting [26, 47, 89], which has not transferred to the 2-body counterpart. Single-person techniques [8, 25, 46] tested on two-people data underperform, which is unsurprising, as they neglect the body-body motion correlations [45]. This motivates the current work, where the most recent modeling advancements are analyzed and integrated. Here, we refer to the best and complementary modeling aspects as *best practices*, which we leverage to bootstrap research on 2-body forecasting.

We propose a systematic analysis of single-person skeleton-based best practices by considering three processing stages (cf. Fig. 3.1): input representation, encoding, and decoding. For the first stage, we identify Discrete Cosine Transform (DCT) [11, 45, 90–92] as an asset to cope with the periodic body movements. For the second stage, we set to encode the body kinematics by Graph Convolutional Networks (GCN), which power the vast majority of most recent techniques [45, 47, 90–92, 125] and subsume general MLP-based formulations [47]. Here we evaluate as best practices the separability of space and time dynamics [125], the learnable adjacencies versus kinematic trees [153], attention [45], and hierarchical body representations [26]. Finally, for the third stage, we contrast the widely-adopted [89, 122, 125] decoding with convolutional networks (*a.k.a.* Temporal Convolutional Network–TCN [6]) with the simpler Fully Connected (FC) layers [47].

We propose a novel initialization technique for the learnable GCN parameters in the encoder. A large body of literature asserts the importance of initialization for performance, convergence speed,

¹Beyond [45], another multi-body dataset has been introduced by [34], but annotations are only available for one individual at the time of writing.

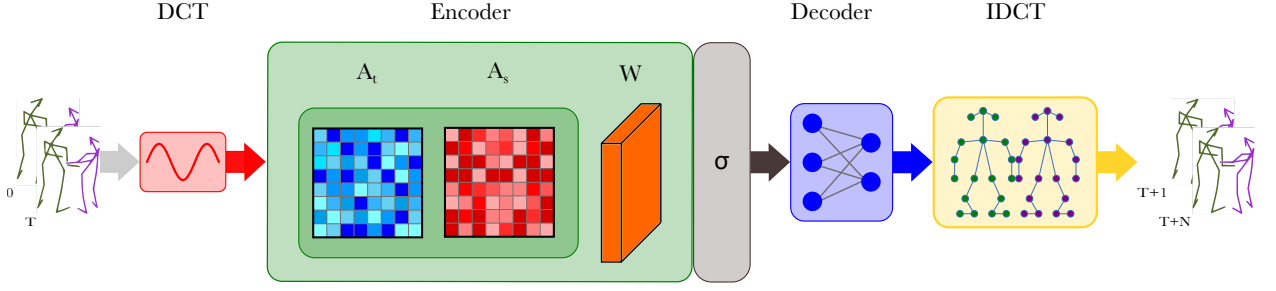


Figure 3.1: The general architecture of a 2-body pose forecasting model employing best practices. First, 3D joint coordinates are mapped to frequencies by DCT coefficients, a best input representation practice. Secondly, body kinematics are encoded by layers of a GCN $\sigma(A_s A_t X W)$, with separable space-time adjacency matrices $\sigma(A_t, A_s)$, learned unconstrainedly, upon our proposed parameter initialization. Thirdly, the FC-based decoder outputs future poses for the two people, mapped to 3D coordinates with inverse-DCT (IDCT).

and robustness, and theory has been devised for MLP [42] and ConvNets [48, 66]. Up until recently, there has been a limited necessity for ad-hoc GCN initialization theories since techniques leveraged mainly shallow networks with fixed graphs structures (e.g., the people neighbors[5, 69, 132], the kinematic tree [26, 153]) or spectral normalizations [60, 62]. Since we determine that unconstrained learnable GCN affinities are best practices, we also develop a novel theory (See Sec. 3.3.4) and experimental study (See Sec. 3.4.4) on the initialization of GCN parameters.

Integrating the selected best practices into a 2-body pose forecasting model yields a large-margin improvement of 21.9% *wrt* the state-of-the-art (SoA) on the most recent ExPI dataset [45]. The best-practice model is also 5 times faster than the current best technique and only has 2% of its parameters. The improvement is similarly consistent in generalization tests, across unseen actions with an overall improvement of 14.7% (cf. Table 3.2) and 14.2% for unseen actors (cf. Table 3.4). And the same best-practice model performs on par (cf. Table 3.8) with the leading single-person pose forecasting techniques on the established Human3.6M dataset [55], without any hyper-parameter tuning. The novel initialization, proposed for the unconstrained learning of GCN affinities, contributes an average performance improvement of 3.5%, and it increases stability, as it reduces the long-term forecasting performance variance by (at least) a factor of 2.

The main contributions are summarized as follows:

- We thoroughly evaluate all leading best practices from single-person pose forecasting and bootstrap research on the 2-body task counterpart;
- We propose a novel theory and experimental study on the initialization of GCNs, applying to unconstrained learnable affinities, accounting for an increase in performance of 3.5% and a 2-fold increase in stability;
- On a closed-set dataset configuration, the best-practice model outperforms the 2-body forecasting SoA by a large margin of 21.9% while employing 2% of the parameters and running 5 times faster.

3.2 Related Work

Here we review related work from the field of human pose forecasting, specifically approaches of spatio-temporal pose modeling and hierarchical body representations. Additionally, we review rel-

evant literature from initialization and multi-agent trajectory forecasting.

Human pose forecasting. Established methodologies for (single) human pose forecasting include Temporal Convolutional Network [69], Recurrent Neural Network [35, 89, 91, 137] and Transformer Networks [2, 45]. The MLP-based approach of [47] holds SoA performance.

Graph Convolutional Networks (GCN) [62, 153] are most popular on the task [26, 74, 122], due to their simplicity and effectiveness. GCNs model the kinematic body part interactions by a plain adjacency matrix at a fraction of the parameters of the otherwise required attention mechanism [45, 90]. In this realm, [90] integrates DCT to consider motion frequency; [26, 75] adopt multi-scale hierarchical representations, grouping joints to model relations between coarser body parts; [122, 125] factorize the spatial and temporal adjacency matrices, and they propose to learn them, unconstrainedly, without kinematic tree priors nor spectral normalization.

As we know, the only work that addresses multi-body pose forecasting is [138]. However, they utilize datasets that do not contain highly interactive actions. For comparison, we ran their model with our setup as a comparison with our proposed method (See Tab. 3.1). By contrast, for the task of 2-body pose forecasting, [45] provides the solely-available dataset (ExPI) and the only 2-body-specific technique, adaptation of [90] with cross-person attention. Not surprisingly, this outperforms single-person techniques.

Initialization. A proper initialization improves performance and accelerates convergence [67], limiting vanishing and exploding gradients [42, 48]. Techniques have been concerned with initializing the weights of linear [42] and convolutional [48, 66, 95] layers, generalizing from hyperbolic ($\tanh$) to rectified-linear unit (ReLU) activations. For GCNs, spectral techniques [62, 76, 163] rely on the spectral normalization of the adjacency matrix to elude vanishing and exploding gradients, while spatial techniques [5] resort to degree-normalized transition matrices, derived from the adjacency. In all prior study cases, the graph connectivity is given. To the best of our knowledge, this work presents the first theoretical and empirical analysis of GCN initialization in the case of unconstrained learnable graph connectivity and edge weights.

Multi-agent trajectory forecasting. For trajectory forecasting, employed techniques include attention [53, 65, 158] and graph-based modeling [72, 98, 123]. The multi-agent relations may parallel the joint-joint interaction. However, nodes in a graph of joints have a fixed cardinality and a semantic meaning (head, torso, hand, etc.), which does not apply to general agent-agent graphs. Notably, best trajectory forecasting techniques model the agent-agent interaction [53, 65, 72, 98, 123, 158], which aligns with the motivation of this work, to forecast the poses of people jointly.

3.3 Methodology

We explore the best models for single-body pose forecasting [26, 47, 90, 91, 125] and select best practices for the 2-body task. We group and evaluate practices in three processing stages (cf. Fig. 3.1): 1) input representation (Sec. 3.3.1); 2) encoding of the body kinematics in the observed frames (Sec. 3.3.2); 3) decoding of the future poses (Sec. 3.3.3). In Sec. 3.3.4, we provide a theory

for the proposed unconstrained-GCN initialization. To facilitate reading, we mark with a green check ✓ the selected *best practices* upon evaluation, cf. Sec. 3.4.

Problem formalization. Across T frames, we observe the motion of two human bodies $\mathcal{B}^1$ and $\mathcal{B}^2$, each consisting of J three-dimensional joints. At time t , the 3D body pose of each person is given by corresponding tensors $\mathcal{B}_t^1, \mathcal{B}_t^2 \in \mathbb{R}^{3 \times J}$. We define the concatenation of two bodies at timeframe t as $\mathbf{x}_t = \mathcal{B}_t^1 || \mathcal{B}_t^2$, thus the observed motion history in T frames is $\mathcal{X}_{in} = [\mathbf{x}_1, \dots, \mathbf{x}_T] \in \mathbb{R}^{T \times 3 \times 2J}$. Our goal is to predict the future N frames’ poses $\mathcal{X}_{out} = [\mathbf{x}_{T+1}, \dots, \mathbf{x}_{T+N}] \in \mathbb{R}^{N \times 3 \times 2J}$.

Preliminaries on the encoder-decoder baseline. We adopt an encoder-decoder architecture [122, 125], and following [153, 161], we encode the observed body parts and their kinematic interaction through a GCN, defined as

$$\mathbf{Y} = \sigma(\mathbf{A}\mathbf{X}\mathbf{W}), \quad (3.1)$$

where $\mathbf{A}$ is the adjacency matrix, $\mathbf{W}$ learnable weights and σ an activation function. Other encodings such as RNNs [22, 30] and MLPs [47] have been proposed, whereas we opt for a graph-based model to exploit the non-euclidean nature of graphs. As a decoder, we examine either a single fully connected layer as in [47] or a convolutional architecture [89, 125].

3.3.1 Input Representation

Most recent techniques [1, 47, 90, 91] use Discrete Cosine Transform (DCT) to represent 3D coordinate input as frequencies, under the claim that this captures the dynamic patterns of moving people better.

Frequency encoding ✓

Given the j -th body joint and the t -th timeframe we define the i -th DCT coefficient as

$$\mathcal{F}(\mathcal{X}^{in})_{j,i} = \sqrt{\frac{2}{T}} \sum_{t=1}^T x_{j,t} \frac{1}{\sqrt{1 + \delta_{i1}}} \cos(\alpha) \quad (3.2)$$

$$\alpha = \frac{\pi}{2T}(2t-1)(i-1), \quad (3.3)$$

where the Kronecker delta function $\delta_{ij} \in \{0, 1\}$ has null value if $i \neq j$ and 1 otherwise. After inference, frequencies are remapped to the pose representation via the inverse DCT decoding function $\mathcal{F}^{-1}$. Previous works [90, 92] truncate high frequencies to avoid jittery motion; we consider the impact of the number of retained DCT coefficients and discover that employing all of them yields the best performances. Studies on the impact of DCT coefficients are shown in Sec. 3.4 and Table 3.9.

3.3.2 Encoding Best Practices

Best-performing single-pose forecasting GCN encoders have considered two main aspects: the space-time separability of adjacency weight matrices and learning the body kinematic graph connectivity and weights. We detail these two aspects and empirically compare them in Table 3.9. Furthermore,

we also consider hierarchical representations of the skeleton proposed by [26], but this is not a best practice, as we determine experimentally. Nor is it a good practice to add attention, as we discuss in this section and quantitatively evaluate in the next.

Space-time separability ✓

Each graph’s intra-relations are expressed through a GCN-based framework that encodes the spatiotemporal motion and the relationships between keypoints in one’s skeleton [125, 153]. Tensor $\mathbf{X} \in \mathbb{R}^{T \times 2J \times C}$ represents a couple’s skeleton pose and motion, adjacency matrices $A_s \in \mathbb{R}^{T \times 2J \times 2J}$ and $A_t \in \mathbb{R}^{2J \times T \times T}$ are responsible for learning spatial and temporal interactions respectively, as in [125]. Matrices are fully learnable, no kinematic tree is used, and the model is free to grasp the relation between body joints. Thus, this module is formulated as follows:

$$\mathbf{Y} = \sigma(A_s A_t \mathbf{X} W), \quad (3.4)$$

where σ is an activation function and $W \in \mathbb{R}^{C \times C'}$ is a tensor of learnable weights defined as a convolution with kernel dimension $k = 1$. Thus, it is conceptually similar to a fully connected layer. However, unlike the MLP design of [47], GCN shares the weights of W across all channels.

Learning the graph connectivity and weights ✓

Some works [26, 153] use inductive biases based on the human body, such as kinematic trees or specifically-devised connectivity weights. In contrast, others learn the graph adding a constraint on the optimization by spectral normalization [61]. Instead, we follow what is done in the most recent work [125]: unconstrained optimization of graph edges and weights i.e., we set A_{st} for nonseparable GCN and A_s, A_t in case of space-time separable GCN as a fully learnable matrix. This is effectively a best practice, experimentally proven in Table 3.9.

Attention

A GCN model equipped with attention is also known as a Graph Attention Network (GAT) [135]. In a GAT, attention re-defines the adjacency matrix terms as a function of the node embeddings. We employ attention to encode the relation between the two actor embeddings B_h^1 and B_h^2 :

$$\mathcal{B}_h^1 = \mathcal{B}^1 W_1, \mathcal{B}_h^2 = \mathcal{B}^2 W_2, \quad (3.5)$$

Where $\mathcal{B}^1, \mathcal{B}^2 \in \mathbb{R}^{T \times J \times C}$ and $W_1, W_2 \in \mathbb{R}^{C \times C}$ are learnable weights to map features in a high-dimensional space. We use these features to calculate attention weights as follows:

$$\eta = \text{softmax}\left(\sigma(\mathcal{B}_h^1 W_3 \| (\mathcal{B}_h^2 W_4)^\top)\right), \quad (3.6)$$

Where $\mathcal{B}_h^1, \mathcal{B}_h^2 \in \mathbb{R}^{T \times J \times C}$, $W_3, W_4 \in \mathbb{R}^{C \times 1}$ and σ is a LeakyRelu activation function. We apply softmax to get attention weights $\eta \in \mathbb{R}^{T \times n \times m}$ constituting n joints in $\mathcal{B}^1$ and m joints in $\mathcal{B}^2$ and reweight B_h^1 and B_h^2 as follows:

$$\mathcal{B}_{out}^1 = \mathcal{B}_h^1 \eta, \mathcal{B}_{out}^2 = \mathcal{B}_h^2 \eta^\top, \quad (3.7)$$

Where $\mathcal{B}_h^1, \mathcal{B}_h^2 \in \mathbb{R}^{T \times J \times C}$ and $\mathcal{B}_{out}^1, \mathcal{B}_{out}^2$ are the outputs of attention module. We observe that in its more common use [135], graph attention is used to estimate the interaction coefficients of the

adjacency matrix A . This is done by learning a function (general MLP) of two node embeddings. By contrast, when the nodes of the graph are semantically given (body parts of a leader and follower person), one may learn the interaction coefficient (i.e., each term of A) directly, with a joint function of all nodes (not just pairs). The direct estimation results in better performance, as shown by the experiments in Sec. 3.4.4. Hence, the GCN with fully-learned parameters is selected as a best practice rather than attention.

Hierarchical body parts

To the best of our knowledge, a high-level motion representation improves the prediction of human poses[75]. [26] achieves this by concatenating the higher level as an extra node and hand-crafting ad-hoc neighborhoods of nodes.

We integrate a module within the model that enables it to decrease the number of skeleton keypoints for both bodies. We allow the model to naturally learn aggregations between nodes by excluding artificial aggregations while shifting between hierarchies. We employ a linear layer that learns an optimized aggregation when downscaling, and the same is done when upscaling to retrieve the original size skeleton. Although we gain a small improvement by adopting hierarchies, it becomes a limiting factor rather than a gain when combined with other best practices.

3.3.3 Decoding Best Practices

In earlier works, convolutions have been employed for the decoding stage [41, 87, 125]. However, the most recent SoA method chose a plain, fully connected layer [47]. In this section, we will analyze the two solutions, and in Sec. 3.4, we will show why we choose the latter.

Convolutional-based decoder

In the convolutional-based decoder, convolutional layers applied to the temporal dimension are responsible for estimating the pose. It aims to forecast the subsequent frames, $t + 1$ to $t + n$, given the first t frames. This structure is known as Temporal Convolutional Network (TCN) [41, 87, 125].

FC-based decoder ✓

The decoder consists of a single linear layer [47] in charge of mapping the observed T frames to the predicted N .

3.3.4 Novel Adjacency Matrix Initialization

We propose a novel initialization methodology, aiming to preserve variance during the forward pass, which matches the preservation of gradients in the backward. Since over several layers a non-unit variance results in vanishing or exploding signals, and neither of those is good for training, as they stall the gradient, we aim to preserve the variance. To do that, under the assumption of a neural network consisting of only linear layers and linear activation functions, [42] proposes to estimate the standard deviation by considering the number of neurons in both the current and previous layer.

It is particularly relevant for our model because it comprises 8 layers while GCNs are often shallow [62]. We propose to randomly initialize the fully learnable matrices A_s , A_t , and W according to a uniform distribution, whose bounds are defined in such a way that considers both the number of graph nodes and the number of timeframes.

Convolutions on graphs that adopt a normalized adjacency matrix [62, 132] use a well-known graph and do not let all nodes interact with each other. Furthermore, normalization avoids vanishing and exploding gradient, yet it limits the performance and, in the end, fully-learnable yields the best performances [122, 125]. Here is the importance of randomly initializing an *ad hoc* fully learnable adjacency matrix, avoiding exploding or vanishing gradients. The response from the Separable GCN at layer l , according to Eq. (3.4), is

$$\mathbf{X}^{l+1} = \sigma \left(A_s^l A_t^l \mathbf{X}^l W^l \right), \quad \forall l. \quad (3.8)$$

Let's assume matrices A_s , A_t , and W to be independent, have zero mean [42, 48] and uniformly distributed. To constrain variance, hence stabilize training and avoid exploding or vanishing gradient, constraining the variance of the output product of n^l neurons at layer l times W to 1 [48] is a sufficient condition, i.e.,

$$\frac{1}{k} n^l \text{Var}[W^l] = 1, \quad \forall l, \quad (3.9)$$

where $k = 2$ in the case of Re-LU activations, which are asymmetric [48] (while $k = 1$ for symmetric activations such as the $\tanh$). For the spatial matrix, rather than the number of neurons n^l , we consider the number of nodes v , which A_s integrates

$$\frac{1}{k} (n_v^l) \text{Var}[A_s^l] = 1, \quad \forall l. \quad (3.10)$$

Similarly, we consider t time frames to initialize the temporal matrix A_t ,

$$\frac{1}{k} (n_t^l) \text{Var}[A_t^l] = 1, \quad \forall l. \quad (3.11)$$

When initializing W with a zero-mean uniform distribution, the constraint of Eq. (3.9) yields the following distribution for the initialization:

$$W^l \sim U \left[-\sqrt{\frac{k}{n^l}}, \sqrt{\frac{k}{n^l}} \right], \quad \forall l. \quad (3.12)$$

The spatial and temporal matrix constraints of Eqs. (3.10) and (3.11) translate to the following initializing distributions for A_s and A_t respectively:

$$A_s^l \sim U \left[-\sqrt{\frac{k}{n_v^l}}, \sqrt{\frac{k}{n_v^l}} \right], \quad (3.13)$$

$$A_t^l \sim U \left[-\sqrt{\frac{k}{n_t^l}}, \sqrt{\frac{k}{n_t^l}} \right], \quad \forall l. \quad (3.14)$$

Detailed Proof on initialization Here we provide more detailed proof for Eqs. 10-11, 13-14 of the main paper. At each layer l , we assume learnable matrices $W^l \in \mathbb{R}^{C \times C'}$, $A_s^l \in \mathbb{R}^{T \times 2J \times 2J}$ and $A_t^l \in \mathbb{R}^{2J \times T \times T}$ to be independent, have zero mean and be uniformly distributed. With T being the number of timeframes, J being the number of joints in one person, and C and C' being the number of input and output channels.

First, we review and demonstrate the proposed initialization for the forward (Sec. 3.3.4) and

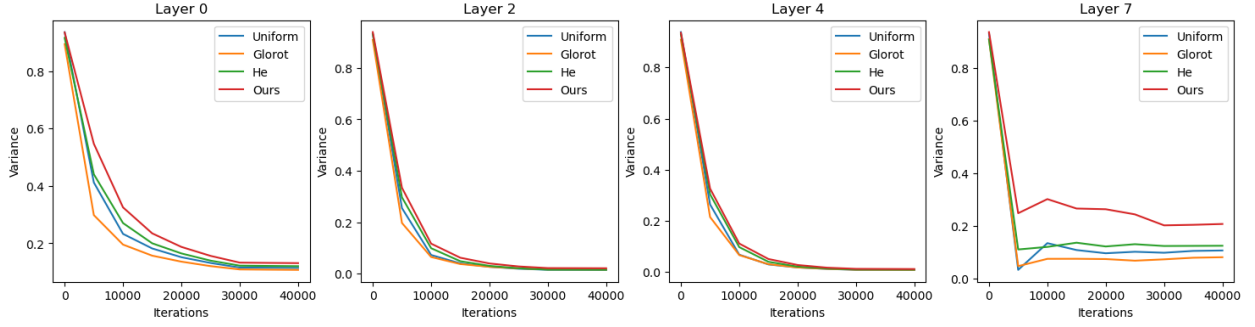


Figure 3.2: Comparison of feature activation variances, at layers 0, 2, 4 and 7, estimated during the model training, upon initialization with random “Uniform”, “Glorot” [42], “He” [48] against “Ours”, our proposed initialization technique.

backward passes (Sec. 3.3.4). Then, in Sec. 3.3.4, we illustrate how the initialization results in better training robustness.

Forward propagation Let us consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ to encode the body kinematics, with all joints at all observed frames as the $2J \times T$ nodes defining the vertex set V , and edges $\epsilon \in \mathcal{E}$ connecting them.

Following up on Eq. (4) from the main paper, the response of a separable GCN [125] layer is

$$\begin{cases} Y^l = A_s^l A_t^l X^l W^l \\ X^l = \sigma(Y^{l-1}), \end{cases} \quad (3.15)$$

where $X \in \mathbb{R}^{T \times 2J \times C}$ is the C -dimensional embedding of each node.

W may be interpreted as a fully connected layer acting on each of the graph node embeddings separately, i.e., on each of the joints from the two people at all times, for a total of $2J \cdot T$ connections. W may be assumed to have C' neurons, i.e. to output $n = C'$ neural activations per node. The matrices A_s and A_t act on the spatial and temporal number of connections of the graph, respectively (please also see [125] for more details). Specifically, A_s may be considered to model the interaction of each node with all $2J$ others at the same frame, by means of $n_v = 2J$ neurons. Correspondingly, we may consider A_t to model the interaction of each node with those of the same joint at all T times, by means of $n_t = T$ neurons.

The number of interactions corresponds to the number of terms that are summed. Assuming matrices to be i.i.d. [48], the variance of the sum yields the sum of variances, thus

$$\text{Var}[Y^l] = n^l n_v^l n_t^l \text{Var}[A_s^l A_t^l X^l W^l]. \quad (3.16)$$

Assuming A_s^l , A_t^l , and W^l to have zero mean [48], the variance of the product of independent variables is

$$\begin{aligned} \text{Var}[Y^l] &= n^l n_v^l n_t^l \text{Var}[A_s^l] \text{Var}[A_t^l] \\ &\quad \mathbb{E}[(X^l)^2] \text{Var}[W^l]. \end{aligned} \quad (3.17)$$

We consider the PReLU as our activation function, i.e.

$$\sigma(X^l) = \max(0, Y^{l-1}) + a \min(0, Y^{l-1}), \quad (3.18)$$

with a being a learnable parameter that, when set to 0, reduces to the ReLU².

This means that for a generic a , $\mathbb{E}[X^l] \neq 0$. Let A_s^{l-1} , A_t^{l-1} , and W^{l-1} have symmetric zero-centered distributions [48]. This may then be also implied for Y^{l-1} , and we may write

$$\mathbb{E}\left[\left(X^l\right)^2\right] = \frac{1+a^2}{2} \text{Var}\left[Y^{l-1}\right]. \quad (3.19)$$

Substituting for Eq. (3.19) in Eq. (3.17) we get

$$\begin{aligned} \text{Var}\left[Y^l\right] &= \frac{1+a^2}{2} n_v^l n_t^l \text{Var}\left[A_s^l\right] \text{Var}\left[A_t^l\right] \\ &\quad \text{Var}\left[Y^{l-1}\right] \text{Var}\left[W^l\right]. \end{aligned} \quad (3.20)$$

Considering L layers, this yields the following variance formulation for the entire separable GCN model:

$$\begin{aligned} \text{Var}\left[Y^L\right] &= \text{Var}\left[Y^1\right] \prod_{l=2}^L \frac{1+a^2}{2} n_v^l n_t^l \\ &\quad \text{Var}\left[A_s^l\right] \text{Var}\left[A_t^l\right] \text{Var}\left[W^l\right]. \end{aligned} \quad (3.21)$$

In order to have the same input and output signal variance for the entire model, it suffices to assume that each layer l has the same input and output signal variances. This corresponds to setting the variance induced by the multiplicative parameters to be 1 i.e.,

$$\frac{1+a^2}{2} n_v^l n_t^l \text{Var}\left[A_s^l\right] \text{Var}\left[A_t^l\right] \text{Var}\left[W^l\right] = 1. \quad (3.22)$$

Towards this goal, it suffices to set each parameter initialization variance as follows

$$\frac{1+a^2}{2} n_v^l \text{Var}\left[A_s^l\right] = 1 \quad (3.23)$$

$$\frac{1+a^2}{2} n_t^l \text{Var}\left[A_t^l\right] = 1 \quad (3.24)$$

$$\frac{1+a^2}{2} n^l \text{Var}\left[W^l\right] = 1 \quad (3.25)$$

Backward propagation The gradient of a separable GCN is

$$\begin{cases} \nabla X^l = A_s^l A_t^l \nabla Y^l \tilde{W}^l \\ \nabla Y^l = \frac{d}{dY^l} \nabla X^{l+1}, \end{cases} \quad (3.26)$$

with $\tilde{W} \in \mathbb{R}^{C' \times C}$, while A_s and A_t have the same dimensionality as in the forward pass. Our backward response number is $\tilde{n}^l = C$ for $\tilde{W}$, and it is still n_v and n_t for A_s and A_t , respectively,

²Also recall that a small a e.g., 0.01, is the LeakyRelu and $a = 1$ is the linear case.

thus

$$\text{Var} [\nabla X^l] = n_v^l n_t^l \tilde{n}^l \text{Var} [A_s^l A_t^l \nabla Y^l \tilde{W}^l]. \quad (3.27)$$

We let A_s^l , A_t^l , $\tilde{W}^l$ and ∇Y^l be independent. Let us assume A_s^l , A_t^l , and $\tilde{W}^l$'s to be zero-centered symmetric distributions, and ∇X^l to have zero mean [48]. Similarly to the forward pass, we have to consider the PReLU activation function. If we assume that $\frac{d}{dY^l}$ and ∇X^{l+1} are independent [48], we get

$$\mathbb{E} [\nabla Y^l] = \frac{1+a^2}{2} \mathbb{E} [\nabla X^{l+1}] = 0, \quad (3.28)$$

$$\mathbb{E} [(\nabla Y^l)^2] = \frac{1+a^2}{2} \text{Var} [\nabla X^{l+1}]. \quad (3.29)$$

Considering Eq. (3.27) and the assumed independence, we elaborate on Eq. (3.29) as follows

$$\text{Var} [\nabla X^l] = n_v^l n_t^l \tilde{n}^l \text{Var} [A_s^l] \text{Var} [A_t^l] \quad (3.30)$$

$$\begin{aligned} & \text{Var} [\nabla Y^l] \text{Var} [\tilde{W}^l], \\ &= \frac{1+a^2}{2} n_v^l n_t^l \tilde{n}^l \text{Var} [A_s^l] \text{Var} [A_t^l] \\ & \text{Var} [\nabla X^{l+1}] \text{Var} [\tilde{W}^l]. \end{aligned} \quad (3.31)$$

For L layers, this yields

$$\begin{aligned} \text{Var} [X^2] &= \text{Var} [\nabla X^{L+1}] \prod_{l=2}^L \frac{1+a^2}{2} n_v^l n_t^l \tilde{n}^l \\ & \text{Var} [A_s^l] \text{Var} [A_t^l] \text{Var} [\tilde{W}^l]. \end{aligned} \quad (3.32)$$

In order to avoid exploding and vanishing gradients, a sufficient condition is to set the gradient of each layer to maintain the signal variance throughout the backpropagation

$$\frac{1+a^2}{2} n_v^l n_t^l \tilde{n}^l \text{Var} [A_s^l] \text{Var} [A_t^l] \text{Var} [\tilde{W}^l] = 1. \quad (3.33)$$

Finally, it suffices to set the variance initialization of each of the parameter matrices as follows:

$$\frac{1+a^2}{2} n_v^l \text{Var} [A_s^l] = 1 \quad (3.34)$$

$$\frac{1+a^2}{2} n_t^l \text{Var} [A_t^l] = 1 \quad (3.35)$$

$$\frac{1+a^2}{2} \tilde{n}^l \text{Var} [\tilde{W}^l] = 1 \quad (3.36)$$

Note that the result in Eq. (3.33) resonates with what was obtained for the forward pass, in Eq. (3.22).

This ensures that the same initialization may be adopted to yield the signal and gradient requirements both in the forward and backward passes.

Training variance The model output should maintain unit variance during training for the sake of activation functions and robust training, avoiding vanishing gradient [42, 48]. This becomes more challenging when adopting deeper networks, which regards our case, as we consider a separable-GCN twice as deep as the original one of [125] (8 Vs. 4 layers).

In Fig. 3.2, we consider the variances of the feature activations at sampled layers during training and compare the result of our proposed initialization against the “Uniform”, “Glorot” [42], and “He” [48] techniques. Observe how “Ours” yields feature variances that are consistently closer to the desired unit variance. This is especially true for the last layer (layer 7), arguably the most challenging.

Action	A1				A2				A3				A4				A5				A6				A7				Average ↓			
Time (msec)	200	400	600	1000	200	400	600	1000	200	400	600	1000	200	400	600	1000	200	400	600	1000	200	400	600	1000	200	400	600	1000	200	400	600	1000
LTD [91]	70	125	157	189	131	242	321	426	102	194	260	357	62	117	155	197	72	131	173	231	81	151	200	280	112	223	315	442	90	169	226	303
HisRep [90]	52	103	139	188	96	186	256	349	57	118	167	240	45	93	131	180	51	105	149	214	61	125	176	252	71	150	222	333	62	126	177	251
MSR-GCN [26]	56	100	132	175	102	187	256	365	65	120	166	244	50	95	127	172	54	100	138	202	70	132	182	258	82	154	218	321	69	127	174	248
MRT [138]	50	98	134	188	79	155	212	307	53	106	152	229	47	95	131	185	52	105	149	215	58	118	166	242	65	136	199	299	58	116	163	238
siMLPe [47]	49	102	137	177	88	180	244	336	57	122	174	254	45	100	137	182	50	103	144	206	59	126	175	250	77	164	134	348	60	128	178	250
XIA [45]	49	98	140	192	84	166	234	346	51	105	154	234	41	84	120	161	43	90	132	197	55	113	163	242	62	130	192	291	55	112	162	238
Ours	34	71	105	159	56	121	181	292	36	78	118	195	30	66	98	145	35	74	113	171	41	88	129	193	47	108	166	261	39	86	129	202

Table 3.1: Results in millimeters for ExPI Common actions split. Our model achieves state-of-the-art results in all actions considered, at each predicted time instant.

Action	A8			A9			A10			A11			A12			A13			A14			A15			A16			Average ↓		
Time (msec)	400	600	800	400	600	800	400	600	800	400	600	800	400	600	800	400	600	800	400	600	800	400	600	800	400	600	800	400	600	800
LTD [91]	252	333	387	174	228	268	139	184	217	239	324	394	175	226	259	148	191	220	176	240	286	143	178	192	146	193	226	177	233	272
HisRep [90]	157	219	257	134	190	233	96	146	187	195	283	358	121	169	206	92	129	160	129	193	245	80	104	121	112	154	187	124	176	218
MSR-GCN [26]	177	239	295	143	179	213	157	222	281	230	289	335	188	245	290	148	198	248	234	319	384	176	232	278	162	218	266	179	238	288
MRT [138]	170	231	308	145	199	270	141	245	338	225	327	481	131	180	253	120	169	238	165	229	322	110	151	209	105	144	201	146	205	291
siMLPe [47]	165	220	258	137	198	246	104	154	198	210	301	432	114	156	187	94	132	160	140	204	255	91	119	138	120	166	204	131	183	225
XIA [45]	156	216	256	126	175	213	96	152	205	191	287	377	118	165	203	91	129	162	122	183	232	81	107	128	106	150	185	121	174	218
Ours	113	164	203	114	167	209	85	136	183	153	231	304	100	148	188	82	125	162	91	138	179	79	109	132	85	124	156	100	149	191

Table 3.2: Results in millimeters for ExPI Unseen actions split. On average, we outperform the baseline considered over short and long time horizons.

3.4 Experiments

We thoroughly evaluate the proposed best practices on the most recent and challenging 2-body pose forecasting dataset ExPI [45], comparing against the SoA and the best single-pose forecasting techniques adapted to the task. The selected best practices also perform on par with the SoA in single-pose forecasting on the established Human3.6M dataset [55].

3.4.1 Benchmark and baselines

Datasets. The dataset used for multi-body pose forecasting, ExPI [45], is a collection of two different dancing pairs performing Lindy Hop sessions, dubbed “extreme human interaction” by the authors [45]. Data were collected in a multi-camera platform with 68 synchronized and calibrated RGB cameras and a motion capture system with 20 mocap cameras. The missing points were manually fixed to ensure good data quality. ExPI contains 115 sequences at 25 fps with 18 body joints for each of the two persons involved. These agents are grouped in two couples, dubbed $(\mathcal{A}_c^1, \mathcal{A}_c^2)$, which perform 16 different actions. Actions A1 to A7 are common to both couples; A8 to A13 performed only by $\mathcal{A}_c^1$ and A14-A16 by $\mathcal{A}_c^2$. Tab. 3.3 shows each action A_i , with $i = 1, \dots, 16$. Refer to the supplementary material in [45] for a more detailed explanation.

Action	Name
A_1	A-frame
A_2	Around the back
A_3	Coochie
A_4	Frog classic
A_5	Noser
A_6	Toss out
A_7	Cartwheel
A_8	Back flip
A_9	Big ben
A_{10}	Chandelle
A_{11}	Check the challenge
A_{12}	Frog-turn
A_{13}	Twisted toss
A_{14}	Crunch-toast
A_{15}	Frog-kick
A_{16}	Ninja-kick

Table 3.3: List of actions and their corresponding names

Based on this, ExPI provides three different splits to test the model on:

- **Common.** Training and test set are composed only of actions performed by both couples. The ones belonging to $\mathcal{A}_c^2$ define the train set, and $\mathcal{A}_c^1$'s the test set.
- **Unseen.** Differently from the previous one, this split has common actions to both $\mathcal{A}_c^1$ and $\mathcal{A}_c^2$ as the train set and couple-specific actions as the test one. This subset allows us to test for generalization.
- **Single.** In this split, a single action from couple $\mathcal{A}_c^2$ is used as a train set, and the same action from couple $\mathcal{A}_c^1$ as the test set. It allows testing how the model generalizes to a new couple for each action.

We also test on Human3.6M [55], an established dataset for single-person pose forecasting. It consists of a total of 3.6 million poses, acquired at 25 fps, depicting seven actors performing 15-day real-life actions, e.g., walking, sitting, and talking on the phone. Following [26, 90, 92], we train on subjects S1, S6, S7, S8, S9, we use S11 for validation, and S5 for testing.

Evaluation metrics. We validate performance by the *Mean per joint position Error*, defined as the MPJPE [55, 91] and renamed as JME in [45] at a future frame t :

$$L_{\text{JME}} = L_{\text{MPJPE}} = \frac{1}{V} \sum_{v=1}^V \|\hat{x}_{vt} - x_{vt}\|_2, \quad (3.37)$$

where $\hat{x}_{vt}$ and x_{vt} are the 3-dimensional vectors of a target joint and the ground truth, respectively. For the joint evaluation of the 2-body position error, the two body poses are normalized into the same reference system. In this work, we keep the MPJPE notation.

3.4.2 Implementation details

In this section, we thoroughly describe the implementation procedures that we have used in training and testing. Furthermore, we describe the iterative approach used during testing.

Training and testing details We use 10 frames as input and 10 frames as output during training. We use an iterative mechanism at test time to make a 1-second prediction(25 frames). We exclusively use our predictions as input for subsequent iterations. We extensively analyze the iterative mechanism impact in the supplementary materials. We adopt the ADAM [59] optimizer and a learning rate of 1×10^{-5} , decayed to 5×10^{-8} after 30K iterations. The model converges in 40K iterations, i.e., the training takes 23 min on a single Nvidia P6000 GPU. We also get an average prediction time on CPU³ of 0.07 seconds compared with the fastest [45]’s 0.4. At each layer, we adopt batch normalization [54] and residual connections.

Baselines. We select the latest and best-performing single-body pose forecasting models, and we adapt them to predict the motion of two people. XIA-Transformer [45] is the only 2-body pose forecasting method in the literature. XIA uses a transformer to encode skeleton features and model the body-body interaction via attention. We consider [138] the only multi-body model based on a Transformer architecture. Due to the lack of multi-body pose forecasting models, we also compare them to single ones. LTD [91] consists of a cascade of GCN blocks acting on frequencies, and its extension, HisRep [90], inserts a motion attention mechanism based on DCT coefficients operating on sub-sequences of the input. MSR-GCN [26] is a hierarchical GCN-based technique that applies multi-scale aggregations, so coarser scales represent groups of body joints and coarser motion. In Table 3.8 we compare ourselves, again, to LTD [91], HisRep [90] and MSR-GCN [26] and, additionally, on two recent single-body models. SeS-GCN [122] adopts an all-separable GCN with a teacher-student approach, and the SoA [47], which consists of MLPs encoding spatial and temporal relationships.

Action	A1				A2				A3				A4				A5				A6				A7			
Time (msec)	200	400	600	1000	200	400	600	1000	200	400	600	1000	200	400	600	1000	200	400	600	1000	200	400	600	1000	200	400	600	1000
LTD [91]	70	126	155	183	131	243	312	415	102	194	252	338	62	117	153	203	71	131	171	231	81	151	199	299	112	223	306	411
HisRep [90]	66	118	153	190	128	231	308	417	74	143	205	295	64	120	159	191	63	121	166	227	90	168	232	312	88	166	232	332
MSR-GCN [26]	64	108	136	170	119	210	282	385	79	144	189	265	59	103	134	173	65	118	162	225	86	151	201	283	96	178	255	362
MRT [138]	63	120	160	218	97	190	249	346	77	148	193	240	51	102	139	186	61	118	163	226	58	115	151	198	82	172	244	340
siMLPe [47]	60	113	145	200	104	202	268	373	76	150	205	305	58	110	151	203	64	123	163	218	76	152	207	277	93	180	254	341
XIA [45]	64	120	160	199	109	200	275	381	59	117	174	277	60	116	162	209	53	106	152	221	65	122	166	223	74	144	203	301
Ours	52	94	128	179	89	176	242	329	42	90	129	200	49	96	134	185	48	99	140	196	52	105	144	198	68	140	204	305

Table 3.4: Results in millimeters for ExPI Single actions split. We outperform in 6 out of 7 stocks all baselines considered according to the MPJPE metric. For the other stocks our model is comparable with the current state of the art.

3.4.3 Evaluation of human pose forecasting

We evaluate our model quantitatively and qualitatively on ExPI’s [45] provided splits. We further test our model’s generalization power on the single-body dataset Human3.6M.

³An AMD Ryzen 5 3600 6-Core processor.

Action	A1				A2				A3				A4				A5				A6				A7				Average ↓			
Time (msec)	200	400	600	1000	200	400	600	1000	200	400	600	1000	200	400	600	1000	200	400	600	1000	200	400	600	1000	200	400	600	1000	200	400	600	1000
LTD [91]	51	92	116	132	51	91	116	148	43	80	103	130	38	70	89	111	39	70	90	116	42	75	94	123	52	101	139	198	45	83	107	137
HisRep [90]	34	69	97	130	44	84	115	150	32	65	91	121	27	56	82	112	28	58	85	121	34	66	88	115	42	83	120	171	34	69	97	131
MSR-GCN [26]	41	75	99	126	54	96	129	180	41	74	98	135	34	61	82	106	33	59	79	109	42	71	93	124	57	103	146	210	43	77	104	141
MRT [138]	34	69	95	128	39	78	106	142	30	59	83	115	28	57	79	110	28	57	79	108	34	68	91	120	39	80	114	160	33	67	92	126
siMLPe [47]	32	69	94	115	44	93	122	160	33	73	102	138	26	61	87	114	28	60	84	112	32	69	93	123	45	94	127	171	34	74	101	133
XIA [45]	32	68	99	128	41	82	116	163	29	58	84	116	24	50	73	96	24	51	75	109	31	62	86	114	41	81	115	160	32	65	93	127
Ours	24	51	76	114	31	66	93	132	23	49	70	103	19	41	60	85	21	44	64	93	24	52	73	100	29	64	95	143	24	52	76	110

Table 3.5: Results in millimeters for ExPI Common actions split. Our model achieves state-of-the-art results in all actions considered, at each predicted time instant.

Action	A8			A9			A10			A11			A12			A13			A14			A15			A16			Average ↓		
Time (msec)	400	600	800	400	600	800	400	600	800	400	600	800	400	600	800	400	600	800	400	600	800	400	600	800	400	600	800	400	600	800
LTD [91]	106	136	155	91	119	135	72	96	116	95	123	146	85	106	116	74	91	101	86	115	137	98	125	134	85	110	124	88	113	129
HisRep [90]	86	120	142	73	104	128	54	82	104	101	144	476	61	82	94	49	67	80	73	105	129	53	73	86	64	89	104	68	96	116
MSR-GCN [26]	88	118	142	90	113	136	90	122	148	103	134	155	101	135	160	74	98	121	103	143	173	87	111	132	84	106	122	91	120	143
MRT [138]	89	121	161	79	108	145	69	100	147	97	133	174	71	96	127	66	88	117	83	113	149	72	98	132	67	92	121	77	105	141
siMLPe [47]	95	125	141	82	114	134	63	93	115	124	174	212	61	80	92	50	67	79	83	116	138	59	81	90	72	99	116	77	106	124
XIA [45]	82	116	142	69	97	120	52	79	104	95	137	171	58	80	93	51	70	84	70	105	134	53	73	88	63	88	104	66	94	116
Ours	68	95	115	66	95	116	52	78	103	86	124	150	54	76	91	47	68	84	59	86	108	53	77	94	53	77	94	60	86	121

Table 3.6: Results in millimeters for ExPI Unseen actions split. On average, we outperform the baseline considered over short and long time horizons.

Action	A1				A2				A3				A4				A5				A6				A7			
Time (msec)	200	400	600	1000	200	400	600	1000	200	400	600	1000	200	400	600	1000	200	400	600	1000	200	400	600	1000	200	400	600	1000
LTD [91]	51	99	129	163	61	110	150	229	53	96	131	188	46	81	106	142	44	79	106	147	53	100	162	176	70	133	163	198
HisRep [90]	51	93	114	127	51	91	116	162	43	80	100	126	38	70	88	118	39	70	90	125	42	75	93	123	52	101	137	188
MSR-GCN [26]	45	83	106	118	57	102	135	178	39	72	100	132	41	77	103	119	35	70	97	125	46	82	107	137	48	90	121	169
MRT [138]	36	69	93	123	44	81	106	138	41	76	96	114	30	61	81	105	33	64	88	121	34	64	83	104	42	83	114	157
siMLPe [47]	43	84	107	137	55	107	142	182	47	91	120	164	39	76	101	129	38	75	99	128	47	90	118	150	58	110	150	197
XIA [45]	43	84	115	131	53	99	136	185	35	68	98	140	37	74	106	128	29	59	86	125	39	72	94	119	43	82	112	152
Ours	34	63	86	115	41	79	105	138	27	55	77	110	31	64	88	119	27	55	77	107	30	58	78	103	38	78	109	154

Table 3.7: Results in millimeters for ExPI Single actions split. We outperform in 6 out of 7 stocks all baselines considered according to the MPJPE metric. For the other stocks our model is comparable with the current state of the art.

ExPI Common Actions. Table 3.1 shows the results obtained from our best model with our selected best practices. These outperform every tested method by a large margin, both the SoA single-person and the SoA 2-body pose forecasting techniques. The overall mean improvement is 22% over all actions and all time horizons. In particular, on all actions, the improvement for short-term future predictions (200 msec) is 29% and 15% for the long-term.

ExPI Unseen Actions. Table 3.2 also showcases improvements using the proposed best practices. On average, across all forecasting horizons, the improvement is 14%.

ExPI Single Actions. In Table 3.4, also for the case of single actions, the best practices report an average improvement of 14.2%. They outperform all other tested techniques in 6 (out of 7) actions at all predicted time horizons. It confirms the generalization of our model to new people.

ExPI qualitative. In Fig. 3.3, the current SoA, ExPI [45], is compared against the best-practice model (*Ours*), qualitatively. The first three columns depict observations; the following four are future motion predictions. The light-colored pictograms represent ground-truth motion. The best practices provide, in general, better predictions. Best improvements are observed in the case of large motion displacements, cf. the last two rows, action “Cartwheel”.

Evaluation of single-person pose forecasting. We test how the 2-body best practices transfer back to single-person pose forecasting for a sanity check. In Table 3.8, observe that the best

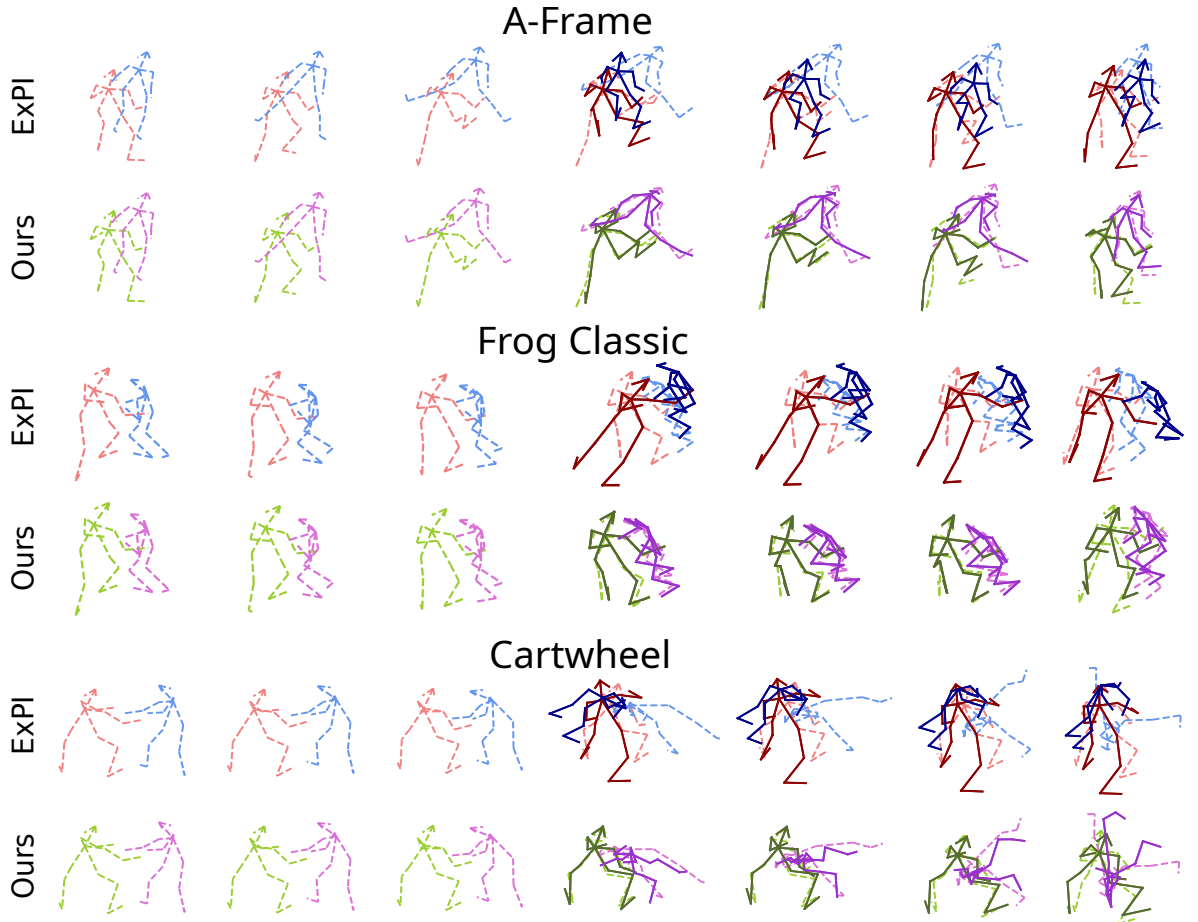


Figure 3.3: Visual comparison of our proposed best-practice model (*Ours*) against ExPI [45]. The first three columns are observed, and the last four are predicted poses. Light-colored and dashed skeletons are GT, and darker and solid ones are predictions. Note the improved larger-displacement motions (*Cartwheel*).

<i>Time Horizon (msec)</i>	MPJPE ↓			
	<i>160</i>	<i>400</i>	<i>560</i>	<i>1000</i>
LTD [91]	23.4	58.9	78.3	114.0
HisRep [90]	22.6	58.3	77.3	112.1
MSR-GCN [26]	25.5	63.3	81.1	114.1
SeS-GCN [122]	29.0	64.0	84.4	113.9
siMLPe [47]	21.7	57.3	75.7	109.4
Ours w/o init.	27.3	64.6	83.1	116.3
Ours	26.8	63.1	81.1	113.2

Table 3.8: Error in millimeters on Human3.6M dataset. We show how our method adapted to single-person human pose forecasting is comparable with the best-performing techniques on average.

Model		Input Repr.	Encoding					Decoding	MPJPE ↓				Param. ↓
		Freq. Enc. ✓	Learn. ✓	Sep. ✓	Init. ✓	Att.	Hier.	FC ✓	200	400	600	1000	(M)
1	[45]	✓	✓			✓			55	112	162	238	8.5
2	Space-time GCN		✓						108	152	255	379	1.08
3	(kin. tree)			✓					81	129	183	260	0.18
4			✓	✓					55	112	156	224	0.18
5	Input repr. practice	✓	✓	✓					41	88	135	219	0.18
6			✓	✓	✓				53	106	148	216	0.18
7	Encoder practices		✓	✓		✓ [†]			55	112	157	228	9.9
8			✓	✓			✓		51	104	148	223	0.18
9	Decoder practices		✓	✓				✓	51	104	145	212	0.17
10		✓	✓	✓				✓	41	89	133	208	0.17
11		✓	✓	✓			✓	✓	51	104	146	217	0.17
12	Best model	✓	✓	✓	✓			✓	39	86	129	202	0.17

Table 3.9: Combinations of best practices. From left to right, we have frequency encoding, fully learnable connections, Space-time separability, initialization, attention mechanism, hierarchy, fully connected layer as a decoder. †: we implement a Graph Attention Network (GAT) tailored for GCNs, similar in spirit to [45] designed for transformers.

practices (*Ours*) yield results within a small margin compared to SoA. Note that, for the sake of this experiment, we just run the 2-body best-practice model *as is*. Without any hyper-parameter tuning. Furthermore, the initialization gives an overall 2.4% over the counterpart model that does not use it.

3.4.4 Evaluation of Best Practices

In this section, we refer to Table 3.9 and thoroughly assess each selected practice. First, we select a baseline GCN model. Secondly, we assess each practice’s performance, added as a standalone extension. Thirdly, we integrate practices. Best practices are assessed based on their standalone performance improvement and complementarity. Finally, in Table 3.10, we evaluate the impact of the proposed initialization in more detail.

Baseline selection. We first select a baseline model on which we test each best practice. We identify three possible GCN-based encoder architectures:

- Space-time GCN [153]: this is a plain GCN model $\sigma(A\mathbf{X}W)$ with learnable A (learnable connectivity and graph weights)
- Space-time separable GCN with learnable kinematic tree: inspired by [153] and [125] to factorize the adjacency matrix into two spatial and temporal learnable matrices, whereby the

spatial connectivity is constrained to the kinematic tree

- Space-time separable GCN with fully-learnable connections: lastly, we evaluate a space-time separable GCN with fully-learnable adjacencies matrices taking inspiration from [125].

As shown in Table 3.9, the space-time GCN with separability (row 3) has an overall decrease of error by 25% compared to the base GCN (row 2). A considerable additional performance boost (18% over all frames) is also given by using the separability and fully-learnable connections (row 4) instead of limiting the learning procedure on the kinematic tree. The simple space-time separable GCN already outperforms XIA [45] while having a fraction of the parameters, although XIA includes DCT representations and attention. Thus GCN with separability and fully-learnable connections is a good baseline to build upon.

Standalone best practices. Table 3.9 shows input representation (row 5), encoding (rows 6-8), and decoding practices (row 9). When considering the input representation and decoding techniques, DCT, and fully connected (FC) layer as decoder, it is clear that both have a considerable impact. The DCT provides a significant boost in short-term predictions, up to 25%, while the FC-based decoder offers a more substantial increase in long-term predictions, up to 7% against TCN (when the box is not $\checkmark$). Regarding the encoder practices, the novel initialization procedure and a hierarchical architecture improve the chosen baseline by 5% and 4%, respectively. On the other hand, using the attention technique did not lead to any gain in performance and is hence not considered a best practice.

Integrated best practices. Rows 10-12 in Table 3.9 refers to the combinations of techniques that performed best independently.

Integrating the input representation using DCT coefficients and the FC-based decoder indicates how these two methods can be used in addition to the standard method. Secondly, we include a Graph Attention Network as explained in Sec. 3.3.2 to account for the interaction. The performance does not benefit from it, and the number of parameters is considerably higher. Lastly, a hierarchical structure lowers performance when combined with other practices, so we do not consider it a best practice. Our proposed initialization improves our best practice model by another 3.5%.

Impact of initialization. Table 3.10 shows the average of multiple runs for different initialization methods and the corresponding standard deviation. We compare our strategy with the Uniform sampling and the two established methodologies of [42], and [48]. Our proposed initialization exceeds or is on par with the others on average, having more than 2.6% improvement over uniform sampling over the longer time horizon. Note also the lower standard deviation of performance for our proposed technique, especially for the most challenging long-term prediction horizon (at least 2x lower), which we interpret as improved stability.

Iterative approach We test different combinations of T input frames and N output frames (See Tab. 3.11) and notice that some perform better than others. Most notable works in pose forecasting [45, 47, 125] use $T = 50$ input frames and $N = 10$ output ones. Conversely, the best results are obtained using $T = 10$ a mechanism [45, 90, 91] that iteratively feeds both parts of the observed history and new predictions as an input.

<i>Time Horizon (msec)</i>	MPJPE ↓			
	<i>200</i>	<i>400</i>	<i>600</i>	<i>1000</i>
Uniform	39.7 ±0.7	87.6 ±0.7	132.2 ±0.5	207.7 ±1.1
Glorot et al.[42]	40.3 ±0.1	89.4 ±1.2	134.3 ±1.5	207.9 ±1.8
He et al.[48]	40.2 ±0.4	88.6 ±0.7	133.4 ±1.4	206.6 ±1.2
Ours	39.2 ±0.4	86.4 ±0.6	129.4 ±1.0	202.2 ±0.5

Table 3.10: Initialization procedures for best practices model.

<i>Input Frames</i>	<i>Output Frames</i>	MPJPE (ms)↓			
		<i>200</i>	<i>400</i>	<i>600</i>	<i>1000</i>
50	1	43	113	176	274
50	5	52	114	162	243
50	10	62	126	174	243
50	50	68	131	177	244
10	1	36	98	164	294
10	5	37	89	141	238
<i>Used</i>	10	39	86	129	202

Table 3.11: Results in millimeters on the ExPI dataset, on average common actions split. We show the impact that different combinations of input-output frames have on performance. Using 10 input frames makes predictions in the short term more accurate, helping results to be more stable in the long term.

3.5 Discussion

This work has identified, reviewed, and experimentally evaluated best practices for 2-body pose forecasting, to bootstrap research in the mostly unexplored task. Best practices have a large impact on SoA performance, and the novel initialization adds further improvement in performance and stability. Notably, predicting the future of two people in interaction yields better estimates than considering each person separately, so 2-body forecasting is recommended for applications such as sports and collaborative assembly in factories.

Chapter 4

OVOSE: Open-Vocabulary Semantic Segmentation in Event-Based Cameras

4.1 Overview

Event cameras, known for their exceptional temporal resolution, low latency, and motion blur resistance, have transformed various deep learning applications [36]. Use cases include autonomous driving [16], visual odometry [20], object detection [96, 104, 171], object recognition [23, 58], and semantic segmentation [9]. Their outstanding performance in challenging conditions makes event cameras optimal for capturing reliable visual data [94]. This capability is crucial for accurate object identification and classification, especially in scenarios like autonomous vehicle operations [40]. Despite their success, integrating event cameras into existing computer vision models is challenging. Their unique data format, featuring asynchronous event streams without traditional image frames [128], necessitates a reassessment of established techniques. While traditional image-based semantic segmentation has made notable progress [78, 149, 152], event cameras, being less prevalent in real-world scenarios, suffer from a scarcity of both raw and labeled data. This fact raises two intertwined issues: the impossibility of collecting internet-scale event datasets and, as a consequence, the difficulty in training data-intensive deep learning techniques. This challenge obstructs effective semantic segmentation model training, especially in scenarios lacking established benchmarks or methodologies.

Recently, powerful foundation models have demonstrated their effectiveness in segmenting conventional images [27, 63, 78]. Some of these models extend the closed-set capabilities of classical semantic segmentation models [17, 18], performing well in open-vocabulary settings where the set of classes in training and testing are disjoint. However, applying these powerful open-vocabulary models directly to event-based images is challenging, and retraining them is impractical due to the scarcity of annotated event-based data [56]. The alternative approach of converting events into images using E2VID [110], and pairing it with an open-vocabulary model does not perform satisfactorily as illustrated in Fig. 4.1. This challenge arises from the inherently ill-posed nature of events-to-image translation [110], creating a domain gap between real grayscale and reconstructed event images.

In this paper, we introduce OVOSE, the first Open-Vocabulary Semantic Segmentation algorithm tailored for event-based data. OVOSE operates as a two-branch network, with one branch

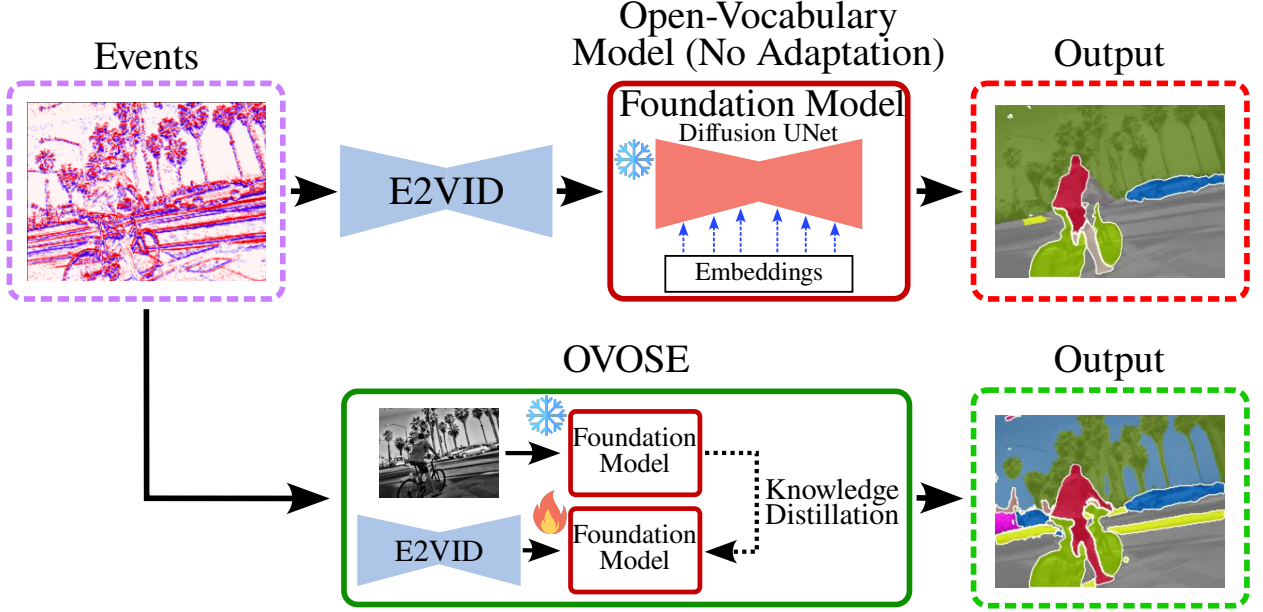


Figure 4.1: Output of a regular RGB foundation model for semantic segmentation and OVOSE in event-based data.

dedicated to grayscale images and the other to event data. Each branch incorporates a copy of an image foundation model, with OVOSE adapting the event branch for optimal performance in event-based data. Our algorithm integrates text-to-image diffusion [108] and the Mask2Former model [21]. Using a CLIP-style image encoder and MLP, we derive embeddings for conditioning the text-to-image diffusion UNet. The UNet’s features guide Mask2Former for mask generation. We categorize the Mask2Former outputs using a frozen CLIP-style text encoder for open-vocabulary segmentation. To enhance model performance, we distill knowledge from the image branch to the events branch, using an E2VID model [110] for translating events to reconstructed images. OVOSE, characterized by its simplicity and effectiveness, outperforms existing models in event-based semantic segmentation, including Unsupervised Domain Adaptation (UDA) methods, demonstrating its effectiveness in addressing the open-vocabulary segmentation problem in event-based data.

Our main contributions can be summarized as follows:

- To the best of our knowledge, we present the first open-vocabulary semantic segmentation approach tailored explicitly for event-based data.
- We introduce OVOSE that distills knowledge to transfer semantic insights from a foundation model trained on regular images to enhance open-vocabulary segmentation performance for event-based data.
- To mitigate the effects of sub-optimal reconstructions, we investigate various mask reweighting strategies and introduced a novel dissimilarity network. This dissimilarity network recalibrates the mask loss by leveraging the differences between reconstructed and original images, enabling precise fine-tuning of the segmentation model and thus producing robust and accurate predictions.
- We perform extensive evaluations in open-vocabulary semantic segmentation for event-based data in two datasets. OVOSE readily outperforms existing closed-set semantic segmentation

methods and straightforward adaptations of open-vocabulary models. A set of ablation studies validates the key components of our algorithm.

4.2 Related Work

4.2.1 Event Camera Semantic Segmentation

Previous work in event-based semantic segmentation includes [3] with an Xception-type network for the DDD17 dataset. Despite their success, the semantic labels generated for the DDD17 dataset using the Xception-type network suffer from low quality. Another notable advancement, as proposed by [38], involved leveraging a synthetic event dataset converted from videos, leading to a substantial enhancement in the overall performance compared to [3]. Knowledge distillation from images to events or vice versa, coupled with transfer learning, has been explored by [139, 140]. However, these methods rely on labeled image datasets such as Cityscapes [24] or labeled event datasets. Messikommer *et al.* [93] proposed a UDA method, aligning image and event embedding spaces using adversarial learning. Despite its effectiveness, this approach suffers from hallucinations [128]. In a more recent contribution, Sun *et al.* [128] introduced a UDA technique to reduce the domain gap between events and images, showcasing improvements across event-based datasets like DSEC-Semantic [40] and DDD17. Still, it necessitates real event-based data for aligning image and event embeddings. A recent approach [155] proposed a self-supervised learning framework for learning event representations using paired events and RGB images. They fine-tuned their model for semantic segmentation by adding a decoder to the pre-trained network but required semantic labels from DDD17 or DSEC-Semantic for fine-tuning. Notably, all these methods are tailored for closed-set semantic segmentation scenarios, where models are trained on a limited number of known classes. In contrast to these methodologies, we propose an open-vocabulary segmentation method designed explicitly for event-based cameras, relying solely on synthetic datasets.

4.2.2 Open-Vocabulary Semantic Segmentation

Prior works for open-vocabulary semantic segmentation in regular images have primarily focused on creating a shared embedding space that links image pixels to class names and descriptions [10, 146, 164]. Another set of approaches, such as [68, 148, 160], relies on CLIP [106] for text and image embedding. Alternatively, some methods combine the CLIP text encoder with the Vision Transformer (ViT) [29] image encoder. OpenSeed [160] is proposed to learn jointly open-vocabulary segmentation and detection. A recently proposed method, ODISE [78], innovatively merges pre-trained text-image diffusion and discriminative models [99, 108, 113, 120] to excel in open-vocabulary panoptic segmentation, leveraging their complementary strengths. However, all these methods are designed for image-based open-vocabulary segmentation. Our experiments show that naively applying these methods for event-based data gives sub-optimal predictions due to the domain gap between images and events. We instead propose to transfer knowledge from a large image foundation model to a model tailored for event data.

4.2.3 Knowledge Distillation

Prior knowledge distillation (KD) [51] [142] methods focus on single modality, *i.e.*, image using logits [15, 147] or features [101, 114] or across different modalities [7, 141, 165] utilizing paired data. KD is also applied to event cameras [140] to distill knowledge from the image-based teacher model to the event-based student model. However, they require labeled image datasets and unlabelled real events and frames to ensure effective transfer for semantic segmentation. Furthermore, [139] employs event-to-image transfer for semantic segmentation, but it gives poor performance [128] when applied for semantic segmentation. A recent approach [56] was employed to transfer knowledge from images to events for image classification. Unlike these approaches, we employ a synthetic training dataset and distill knowledge from an image foundation model to a foundation model tailored for open vocabulary semantic segmentation in events. The problem becomes complex as we strive to bridge the gap between events and images, and additionally tackle the synthetic-to-real gap. We show in the experiments the effectiveness of knowledge distillation of the proposed method.

4.3 Open-vocabulary Segmentation in Events

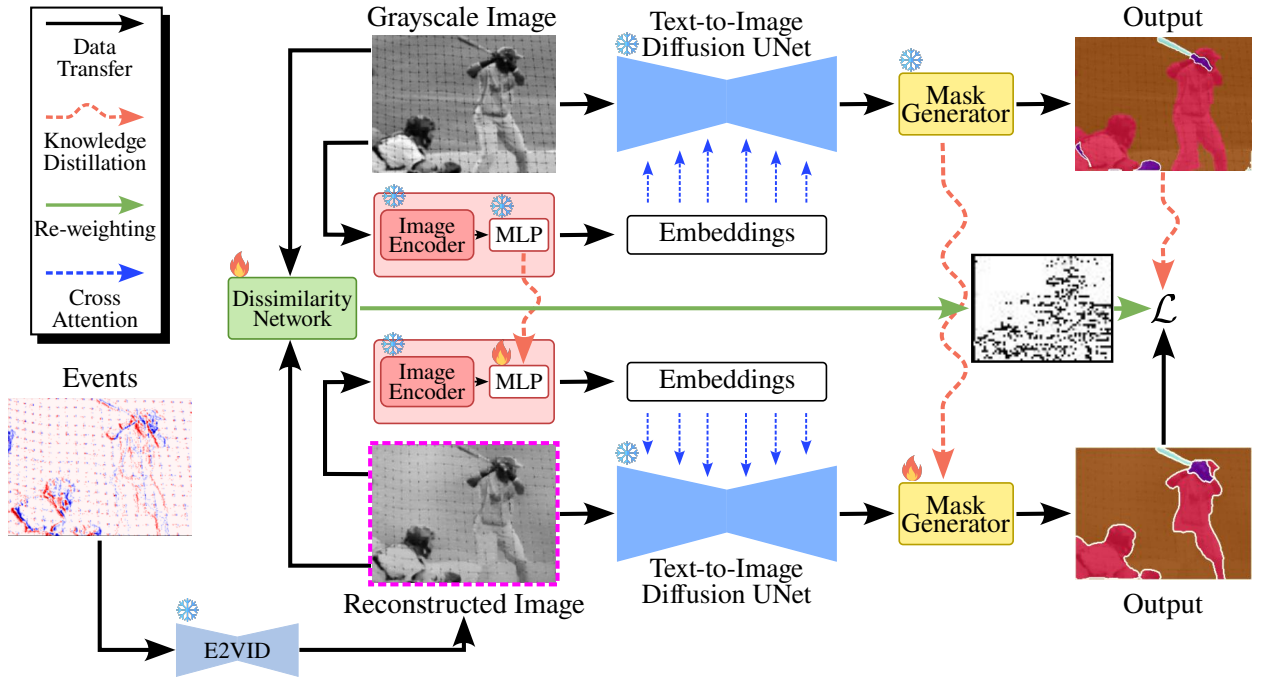


Figure 4.2: Overview of OVOSE pipeline. Our algorithm comprises two components: the original grayscale image branch and the event-based branch. Initially, events are transformed into a grayscale image using the E2VID model. Subsequently, both the original and reconstructed grayscale images undergo text embedding through an image encoder and an MLP. The features from a frozen text-to-image diffusion UNet are then extracted for each tuple of image and text embedding. For each branch, a mask generator predicts class-agnostic binary masks and associated mask embedding features. Categorization is achieved through a dot product between mask embedding features and text embeddings. Both branches are initialized with ODISE weights [149], and knowledge distillation occurs from the original image branch to the event-based branch during training. Original and reconstructed images are input into a dissimilarity network to weigh the distillation in the outputs. During the evaluation, only the event-based branch is utilized.

Preliminaries. Each pixel inside an event camera operates independently and asynchronously, reporting changes in brightness as they occur and staying silent otherwise. Each pixel stores a

reference brightness level and continuously compares it to the current value. If the difference in brightness exceeds a threshold, that pixel resets its reference level and generates an event $\mathbf{e}_i = [x_i, y_i, t_i, p_i]$: a discrete packet that contains the pixel address (x_i, y_i) , timestamp t_i and polarity $p_i \in \{-1, 1\}$ representing an increase or decrease of a brightness level. As a consequence, the full event data stream can be represented as $\mathbf{E} \in \mathbb{R}^{N \times 4}$, where N is the number of events in the stream. In this paper, we transform events into grid-like representations [39], such as voxel grid [128, 168] to facilitate further processing. We utilize a voxel grid representation of events as input to our model.

Problem statement. Open-vocabulary represents a generalization of the zero-shot task in semantic segmentation. In this setting, a model predicts masks for unseen classes $\mathcal{C}^{unseen}$ by learning from labeled data of seen classes $\mathcal{C}^{seen}$. The sets of seen and unseen classes are separate and do not overlap, *i.e.*, $\mathcal{C}^{unseen} \cap \mathcal{C}^{seen} = \emptyset$. The objective of this work is to train a model F_{θ} with parameters θ to predict the segmentation map of some stream of events. To solve this, we have a unlabeled training set $\mathcal{X}_{train} = \{\mathbf{E}_i^{(t)}, \mathbf{I}_i^{(t)}\}_{i=1}^{N_t}$, where $\mathbf{E}_i^{(t)}$ is the i th stream of events, $\mathbf{I}_i^{(t)} \in \mathbb{R}^{H \times W}$ is the i th original grayscale image¹ with H and W the height and width of the image, and N_t is the number of streams in the dataset. For the training set, we additionally need the set of seen classes $\mathcal{C}^{seen}$. We evaluate F_{θ} in a testing set $\mathcal{X}_{test} = \{\mathbf{E}_i^{(s)}, \mathbf{Y}_i^{(s)}\}_{i=1}^{N_s}$ where $\mathbf{Y}_i^{(s)} \in \{0, 1\}^{H \times W \times |\mathcal{C}^{unseen}|}$.

4.3.1 Overview of OVOSE

Figure 4.2 shows the pipeline of OVOSE. Our algorithm is divided into two sections: (i) the image branch that takes as input the original grayscale images $\mathbf{I}_i^{(t)} \in \mathcal{X}_{train}$, and (ii) the event branch that takes as input the stream of events $\mathbf{E}_i^{(t)} \in \mathcal{X}_{train}$ or $\mathbf{E}_i^{(s)} \in \mathcal{X}_{test}$. It is worth clarifying that during the evaluation we only use the event branch. During training, the whole image branch is frozen and only used to distill knowledge to the event branch. For the event branch, we use the pre-trained E2VID model with parameters θ_e [110] to transform some stream of events $\mathbf{E}$ into a reconstructed image $\hat{\mathbf{I}}$, thus $F_{\theta_e}(\mathbf{E}) = \hat{\mathbf{I}}$. Taking as input $\mathbf{I}$ or $\hat{\mathbf{I}}$, the forward pass of each branch is identical, so we explain only one of these in the following.

We first employ a frozen image encoder of type CLIP $\mathcal{V}(\cdot)$ [106] to encode $\mathbf{I}$ or $\hat{\mathbf{I}}$ into an embedding space. Subsequently, a learnable MLP is used to project the image embedding into implicit text embeddings. We use $\mathbf{I}$ or $\hat{\mathbf{I}}$ along with the implicit text embeddings as input to a text-to-image diffusion model [78, 108] for feature extraction. More precisely, we employ a UNet architecture to do the denoising process. Formally for the image branch, we have:

$$\mathbf{f} = F_{\theta_U}(\mathbf{I}, \text{MLP}(\mathcal{V}(\mathbf{I}))), \quad (4.1)$$

where $\mathbf{f}$ is the feature vector from the diffusion network in the image branch, and θ_U is the parameters of the UNet model. Therefore, we use $\mathbf{f}$ as input to a Mask2Former model [21] to produce class-agnostic binary masks with their corresponding mask embeddings. For categorization, we use a text encoder of type CLIP $\mathcal{T}(\cdot)$ to embed the categories in $\mathcal{C}^{seen}$. We thus perform a dot product between text and mask embeddings to categorize each mask.

We use the estimated segmentation map of the image branch as the ground truth of the event branch by computing a loss function between both outputs. However, as we kept E2VID frozen, under sub-optimal reconstructions, we weighed this loss function using the output of a dissimilarity

¹The symbol we use in this paper for images $\mathbf{I}$ should not be confused as the identity matrix.

network to give more emphasis to the regions where E2VID provides good reconstruction values. We further perform knowledge distillation from the Mask2Former in the image to the Mask2Former in the event branch, and similarly for the MLP networks. The parameters of the two branches are initialized with the weights of the ODISE model [149].

4.3.2 Distilling Image Embeddings

To address the difference in image embeddings between the output of the MLP for the grayscale image and the corresponding output for the synthetic (reconstructed) image, we implement knowledge distillation. This involves transferring knowledge from the image embeddings of the original image to those of the reconstructed image in the event branch. To do so, we introduce the minimization of the Frobenius norm ($\|\cdot\|_F$) of the matrix of differences between real and reconstructed images encoded by the trainable MLP:

$$\mathcal{L}_t = \|\text{MLP}_{\mathbf{I}}(\mathcal{V}(\mathbf{I})) - \text{MLP}_{\mathbf{E}}(\mathcal{V}(\hat{\mathbf{I}}))\|_F, \quad (4.2)$$

where $\text{MLP}_{\mathbf{I}}(\cdot)$ is the frozen MLP of the image branch and $\text{MLP}_{\mathbf{E}}(\cdot)$ is the MLP of the event branch. In other words, we leverage the information encoded in the image embeddings of the original image to guide the learning of the image embeddings for the reconstructed image. The Frobenius norm serves as a metric to quantify the dissimilarity between these embeddings, enabling the model to refine its representation of image information and enhance the consistency between the two image modalities.

4.3.3 Feature Distillation

To provide further guidance from the image branch to the event one, we minimize the Frobenius norm of the matrix differences of the outputs of each layer in the transformer decoder of Mask2Former between the original image and reconstructed image:

$$\mathcal{L}_f = \frac{1}{L} \sum_{i=1}^L \|\mathbf{D}_i - \hat{\mathbf{D}}_i\|_F, \quad (4.3)$$

where L represents the total number of transformed decoder layers, and $\mathbf{D}$ and $\hat{\mathbf{D}}$ are the output matrices of each layer in the image branch and events branch, respectively.

4.3.4 Mask Re-weighting

While maintaining E2VID in a frozen state, instances arise where sub-optimal reconstructions occur. In response, we impose a classification on inadequately reconstructed regions. However, this strategy potentially compromises the model’s performance in areas where the reconstruction is accurate and sensible. We introduce a dissimilarity network to discern the differences between the grayscale image and its counterpart reconstructed from events. Illustrated in Figure 4.3, the dissimilarity network comprises two convolutional layers. The initial convolutional layer shares weights for grayscale and reconstructed images and is followed by a rectified linear unit (ReLU) activation layer. The subsequent convolutional layer is followed by a sigmoid activation function $\sigma(\cdot)$. The original grayscale and reconstructed images are fed into the first convolutional block, and the squared error between

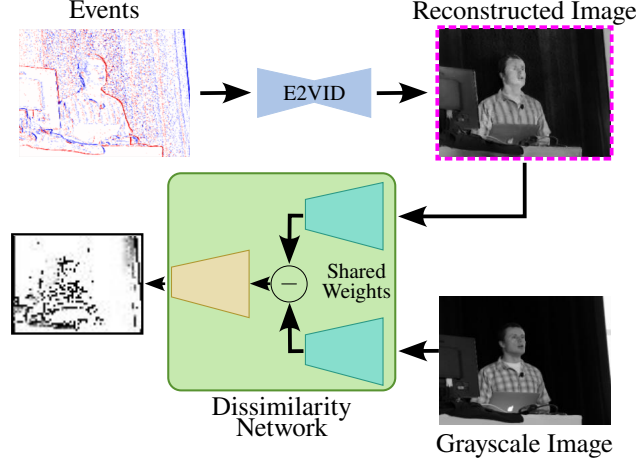


Figure 4.3: Dissimilarity network takes original grayscale and reconstructed images as input, and it outputs an error map to reweight the mask loss. It can be seen that E2VID is unable to reconstruct the stripes and hence considered a high error area by the dissimilarity network

their respective outputs is computed. This resultant output is fed into the second convolutional layer to obtain a reweighting map. It is evident from the error map depicted in Figure 4.3 that a notable discrepancy exists between the grayscale image and the reconstructed image, particularly concerning the stripes on the shirt in this example. Consequently, this discrepancy in the error map indicates reduced importance attributed to that specific area. Mathematically, this process can be expressed as:

$$\begin{aligned} \mathbf{M}_{gray} &= \text{ReLU}(\text{conv}_1(\mathbf{I})), \\ \mathbf{M}_{rec} &= \text{ReLU}(\text{conv}_1(\hat{\mathbf{I}})), \\ \mathbf{M} &= \sigma(\text{conv}_2(\mathbf{M}_{gray} - \mathbf{M}_{rec})^2), \end{aligned} \quad (4.4)$$

where $\mathbf{I}$ is the grayscale original image and $\hat{\mathbf{I}}$ is the reconstructed image from E2VID. We use $\mathbf{M}$ to re-weight our distillation loss at the level of segmentation maps $\mathcal{L}_m$. This loss for the i th stream is given by:

$$\mathcal{L}_m = \mathbf{M} \odot \mathcal{L}_{CE}(\mathbf{Y}_i, \hat{\mathbf{Y}}_i), \quad (4.5)$$

where $\mathbf{Y}_i \in \mathbb{R}^{H \times W \times |C^{seen}|}$ is the output segmentation map of the image branch, $\hat{\mathbf{Y}}_i \in \mathbb{R}^{H \times W \times |C^{seen}|}$ is the output segmentation map of the event branch, $\mathcal{L}_{CE}$ is the cross-entropy loss, and $\odot$ is the point-wise product between matrices. We illustrate a sample output of the dissimilarity network in Figure 4.4. It can be seen that OVOSE successfully ignores the areas where the error is high, for example, the person and the elephant's trunk.

4.3.5 Category Label Supervision

As we have access to category labels in the training set, we follow [149] to compute a category label loss $\mathcal{L}_c$. To this end, we compute the probability of a mask belonging to one of the training categories using a cross-entropy loss with a learnable temperature parameter as in [149].

The total loss of OVOSE is given by:

$$\mathcal{L}_{final} = \mathcal{L}_t + \mathcal{L}_f + \lambda_m \mathcal{L}_m + \lambda_c \mathcal{L}_c \quad (4.6)$$

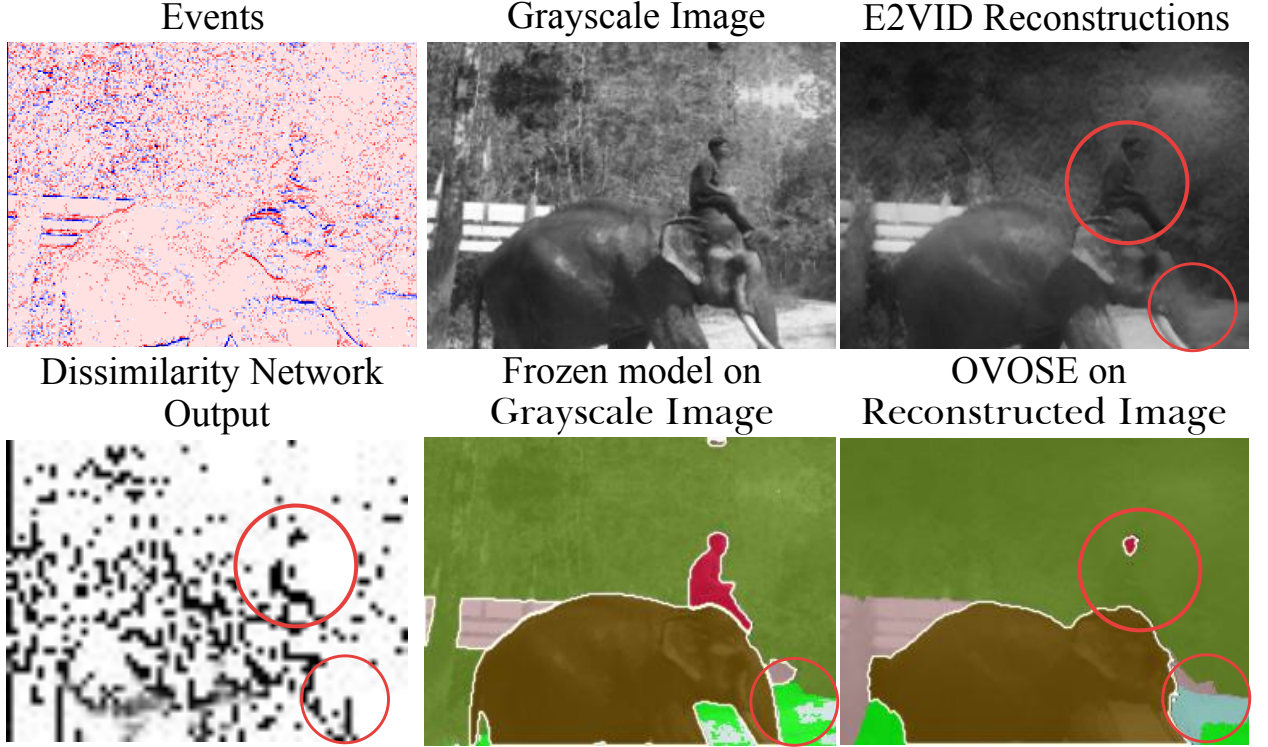


Figure 4.4: The impact of reweighting the mask loss, influenced by the dissimilarity between the grayscale and reconstructed images. Areas such as the person and the elephant’s trunk exhibit suboptimal reconstruction, leading to their exclusion in the reweighting process.

where $\lambda_m = 5.0$ and $\lambda_c = 2.0$ are regularization parameters.

4.4 Experiments and Results

In this section, we present an overview of the baseline methods for comparison, details on the training and test data, and a comprehensive analysis of both quantitative and qualitative results. Subsequently, we delve into ablation studies to validate the components of OVOSE.

4.4.1 Experimental Framework

Baseline methods. As there is no open-vocabulary semantic segmentation method for events, we benchmark OVOSE against leading UDA methods for semantic segmentation in event-based data. However, this comparison is unfair with OVOSE since UDA methods: (i) know the set of unseen classes, (ii) have access to one or multiple labeled source datasets, and (iii) have access to the unlabeled testing dataset for adaptation. We also compare our algorithm with straightforward adaptations of open-vocabulary semantic segmentation methods in regular images. This adaptation consists of reconstructing a grayscale image from the stream of events with the E2VID model (similar to our event branch) and using this as input to the open-vocabulary model. More precisely, OVOSE is compared with the UDA methods E2VID [110], EV-transfer [93], VID2E [38] and ESS [128]. For the open-vocabulary methods, we compare our algorithm against E2VID+OpenSeed [160] and E2VID+ODISE [149]. We use the mean Intersection over Union (mIoU) and pixel accuracy metrics for the quantitative evaluations.

Method	Type	Problem Formulation	Training Data	Input	Acc (%) $\uparrow$	mIoU (%) $\uparrow$
EV-Transfer [93]	UDA	Closed-Set	Cityscapes+DSEC	events+frames	60.50	23.20
E2VID [110]	UDA	Closed-Set	Cityscapes+DSEC	events+frames	76.67	40.70
ESS [128]	UDA	Closed-Set	Cityscapes+DSEC	events+frames	84.04	44.87
E2VID+OpenSeed [160]	Translation	Open-Set	MS-COCO	events	65.25	32.82
E2VID+ODISE [78]	Translation	Open-Set	MS-COCO	events	81.24	43.61
OVOSE (Ours)	Distillation	Open-Set	EV-COCO	events	85.67	48.44

Table 4.1: Results on DSEC Semantic in UDA and open-vocabulary setting. OVOSE not only outperforms all the UDA methods even though they are trained in closed-set settings but also translated open-vocabulary methods.

Method	Type	Problem Formulation	Training Data	Input	Acc (%) $\uparrow$	mIoU (%) $\uparrow$
EV-Transfer [93]	UDA	Closed-Set	Cityscapes+DDD17	events+frames	47.37	14.91
E2VID [110]	UDA	Closed-Set	Cityscapes+DDD17	events+frames	83.24	44.77
VID2E [38]	UDA	Closed-Set	Cityscapes+DDD17	events+frames	85.93	45.48
ESS [128]	UDA	Closed-Set	Cityscapes+DDD17	events+frames	87.86	52.46
E2VID+OpenSeed [160]	Translation	Open-Set	MS-COCO	events	33.25	17.95
E2VID+ODISE [78]	Translation	Open-Set	MS-COCO	events	84.6	48.12
OVOSE (Ours)	Distillation	Open-Set	EV-COCO	events	88.8	53.39

Table 4.2: Results on DDD17 in UDA and open-vocabulary setting. OVOSE outperforms all the UDA methods trained in closed-set settings but also translated open vocabulary methods

Training data. As described in the problem statement, OVOSE requires a training dataset $\mathcal{X}_{train}$ where we have the stream of events and their corresponding grayscale images. To address this need, we leverage synthetic training data as introduced by [110]. This synthetic dataset was generated using the event simulator ESIM [109], which simulated MS-COCO images [80]. The dataset comprises 1,000 sequences, each spanning 2 seconds, with grayscale images and events of dimensions 240×180 . For our training purposes, we resize the images and events to dimensions 256×192 , ensuring compatibility and optimization for our network. Following [149], we use MS-COCO classes for category label supervision.

Evaluation datasets. We evaluate the open-vocabulary performance of OVOSE on two popular event camera-based self-driving datasets:

- *DAVIS Driving Dataset* (DDD17) is a dataset for semantic segmentation in autonomous driving. It includes 12 hours of driving data recorded with a DAVIS camera, providing synchronized events and grayscale frames. [3] pre-trained a Xception network to generate semantic pseudo-labels for the DAVIS frames. Due to the low resolution, the labels were consolidated into six classes: flat (road and pavement), background (construction and sky), object (pole, pole group, traffic light, traffic sign), vegetation, human, and vehicle.
- *DSEC-Semantic* was created [40]. DSEC comprises 53 driving sequences captured in diverse urban and rural settings in Switzerland, using automotive-grade standard cameras and high-resolution event cameras. It consists of high-resolution images 1440×1080 , synchronized events 640×480 , and semantic labels [128] are generated using a state-of-the-art semantic segmentation method [131]. The fine-grained labels for 19 classes are consolidated into 11 classes for practical use: background, building, fence, person, pole, road, sidewalk, vegetation, car, wall, and traffic sign. The dataset includes 8082 labeled frames for training and 2809 for testing.

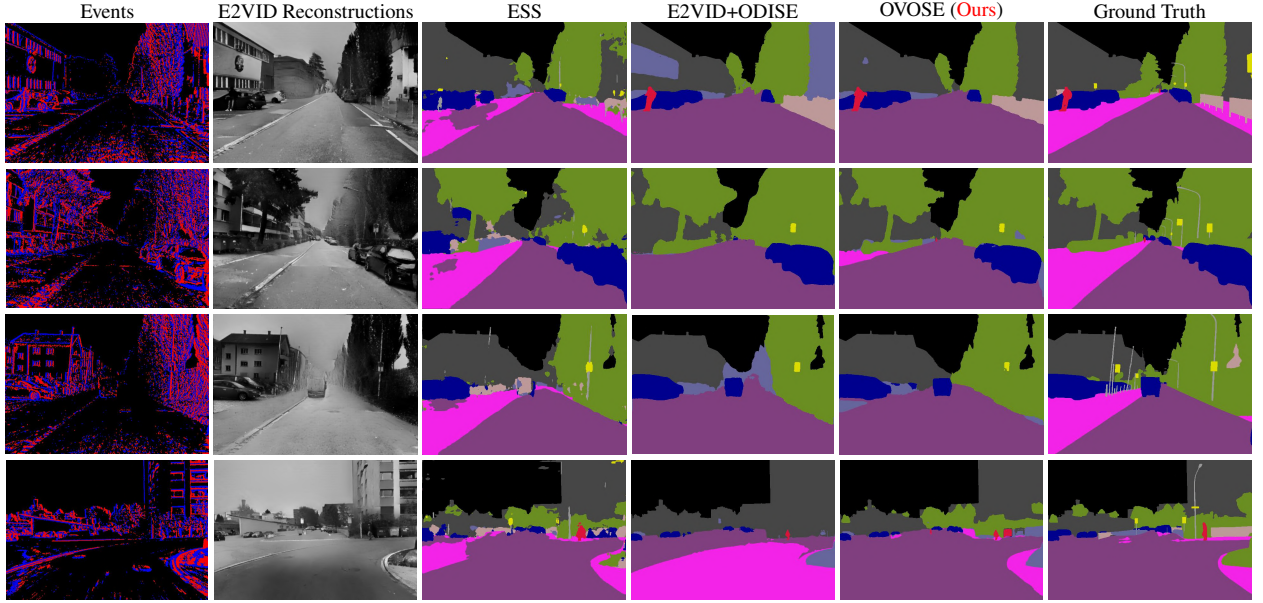


Figure 4.5: Qualitative samples from ESS in UDA closed-set, E2VID+ODISE, and OVOSE in open vocabulary setting. As compared to ESS and E2VID+ODISE, OVOSE produce accurate and less noisy predictions even though it is trained on a synthetic dataset.

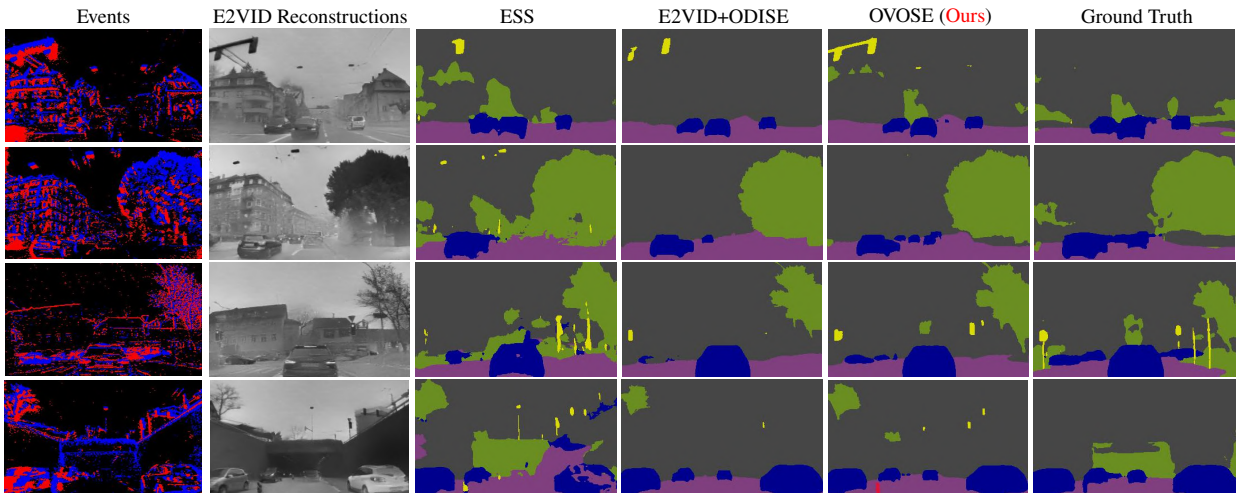


Figure 4.6: Qualitative samples from DDD17 in UDA closed-set, E2VID+ODISE, and OVOSE in open vocabulary setting. ESS works well in predicting poles and pavement, however OVOSE does better overall prediction, especially vehicles, persons, vegetation, and construction.

Implementation details. We keep the image branch frozen and fine-tune the MLP and Mask2Former of the event branch. We use convolutions with 3×3 kernel size with a stride of 2 and padding 1 for the dissimilarity network. We set the learning rate as 1×10^{-5} . We train OVOSE using Adam optimizer with a batch size of 4 and trained for 20 epochs on Nvidia Ampere GPU with 48GB of RAM. We initialize the weights of text and image encoder from [106], text-to-image diffusion UNet from [108], and MLP and Mask2former from [149]. For qualitative results we use the open source code and weights provided by [128] and [149]. For OpenSeed [160], we use their provided open-source code and weights. Upon acceptance, we will release the code and final model weights of OVOSE.

4.4.2 Results

Semantic segmentation on DSEC-Semantic. Figure 4.5 shows a qualitative comparison of OVOSE against ESS and ODISE. We obtain the qualitative results for the ESS in Figure 4.5 by utilizing the official model provided by the authors. Even though the ESS method is trained on real events in a closed-set setting, its overall semantic segmentation across the entire image appears noisy. Notably, it fails to segment vehicles in several instances accurately. For E2VID+ODISE in an open vocabulary setting, it misclassifies parts of the road and buildings. In contrast, OVOSE, trained on synthetic events and operating within an open vocabulary setting, excels in recognizing traffic signs and delivers superior overall semantic segmentation across the entire image. This is particularly evident in its ability to discern intricate details and provide accurate segmentations, showcasing its robustness and adaptability in the driving scenario of DSEC.

Table 4.1 presents the comparison of OVOSE against the baseline methods on the DSEC-Semantic dataset. Even though the UDA methods are trained in the closed-set setting using real events and urban street datasets similar to DSEC, OVOSE surpasses their performance by a significant margin. Notably, our model improves the state-of-the-art UDA method ESS [128] by a substantial 3.57% in the mIoU metric. Similarly, OVOSE outperforms E2VID+ODISE by 4.83% in the mIoU metric, showcasing the superior performance of OVOSE for semantic segmentation in event-based data.

Semantic segmentation on DDD17. Figure 4.6 presents a qualitative comparison between OVOSE, ESS, and E2VID+ODISE. ESS effectively segments poles and other event-specific objects that may not be available in the ground truth. However, it struggles with inaccuracies and noise when segmenting cars. In contrast, OVOSE operating in an open vocabulary setting showcases the ability to segment traffic signs even when unlabeled in the ground truth images. Notably, OVOSE excels in differentiating between classes, particularly with vehicles, and demonstrates more accurate and noise-resistant semantic segmentation compared to the ESS. OVOSE also outperforms E2VID+ODISE by providing clearer object recognition and segmentation. This comparison emphasizes the need for knowledge distillation and OVOSE’s capacity to deliver precise semantic segmentation even in scenarios with incomplete or noisy ground truth annotations.

When it comes to quantitative evaluation in the DDD17 dataset, the presence of noisy ground truth labels, as highlighted by [128], poses challenges. Additionally, the previous work by Ev-SegNet [3] merged several labels, further complicating the evaluation process. To address this, we consider the original classes pre-merge for predictions, subsequently merging them to facilitate a comparison with the ground truth. Table 4.2 summarizes the quantitative comparison of OVOSE with the base-

Method	Acc (%) $\uparrow$	mIoU (%) $\uparrow$
Baseline	81.2	43.61
Distillation	85.0	47.01
Distillation+Reweight CS	85.1	48.01
Distillation+Reweight SD	85.1	47.32
Distillation+Reweight DN	85.6	48.44

Table 4.3: Ablation of distillation strategies of ABAS on DSEC Semantic dataset.

Method	Labels	Acc (%) $\uparrow$	mIoU (%) $\uparrow$
EV-SegNet [3]	events	88.61	51.76
ESS [128]	events	89.25	51.57
ESS [128]	events+frames	89.37	53.29
OVOSE (ours)	events+frames	89.45	58.62

Table 4.4: Comparison of OVOSE with state-of-the-art methods on DSEC Semantic dataset in a supervised setting. OVOSE outperforms state-of-the-art ESS[128] by 5.33% mIoU

line methods on the DDD17 dataset. OVOSE significantly improves the UDA state-of-the-art ESS method [128] by 0.93% in the mIoU metric. Notably, our algorithm outperforms E2VID+OpenSeed and E2VID+ODISE by large margins, underscoring the significance of knowledge distillation and mask reweighting in semantic segmentation for event cameras.

DSEC Supervised Results. We compare OVOSE against the previous methods EV-SegNet [3] and ESS [128] in a supervised setting. To that end, we train OVOSE in a supervised closed-set setting on the DSEC-Semantic dataset [128]. We report the results in Table 4.4, where labels are either event labels from DSEC, frame labels from the Cityscapes [24] dataset, or both. For OVOSE, frame labels means EV-COCO labels. OVOSE outperforms ESS by a large margin of 5.33% mIoU in Table 4.4.

We also show qualitative results in the supervised setting in Figure 4.7. Comparing ESS and OVOSE, we notice that OVOSE segments pavement and cars better than previous works. Effectively, the performance of OVOSE is better across all classes consistently.

Similarly, OVOSE’s increase in mIoU in Table 4.4 indicates its exceptional ability to accurately segment smaller objects like poles and traffic signs, which is noticeable from Figure 4.7.

The mIoU metric gives equal importance to small objects, making it more sensitive to improvements in segmenting these elements. Therefore, our model’s success in precisely segmenting these small objects significantly contributes to the substantial mIoU increase.

4.4.3 Ablation Study

We conduct an ablation study to validate the reweighting schemes and knowledge distillation techniques in OVOSE. To that end, we test the following settings:

- Baseline: We evaluate OVOSE without training, this is equivalent to E2VID+ODISE.
- Distillation: We evaluate OVOSE without reweighting.
- Distillation and reweighting based on Cosine Similarity (CS): We compute the cosine similarity between the original and reconstructed image, and use this value to reweight the loss function.

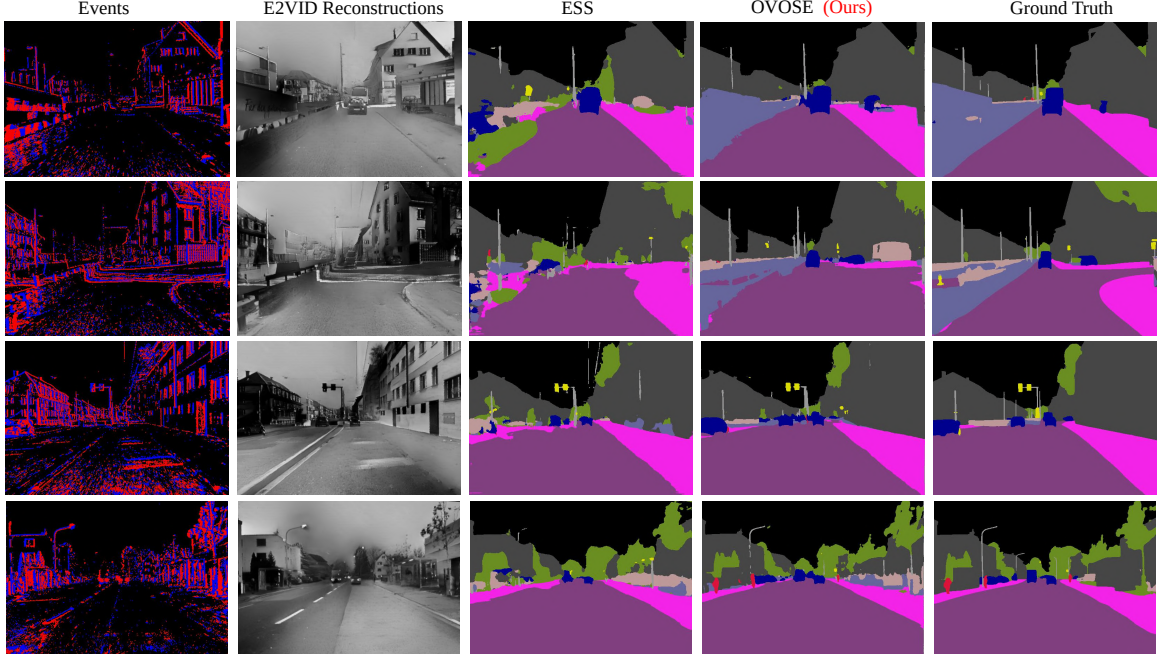


Figure 4.7: Qualitative comparison of ABAS with ESS[128] on DSEC-Semantic dataset in supervised setting utilizing semantic labels. ABAS performs more accurately and close to ground truth semantic segmentation as compared to ESS.

- Distillation and reweighting using output image from Stable Diffusion (SD): We use the image outputs of stable diffusion from the image and event branch, and compute a squared difference between these to reweight the binary cross-entropy on masks.
- Distillation and reweight with Dissimilarity Network (DN): We employ the dissimilarity network to reweight the binary cross-entropy mask loss. This corresponds to OVOSE.

Table 4.3 shows the results of the ablation study. Distilling knowledge from the image branch to the event branch resulted in a notable increase in accuracy by 3.8% and mIoU by 3.4%, showcasing the effectiveness of this approach. Moreover, the introduction of the dissimilarity network for reweighting the loss function gives an additional improvement of 1.43% in the mIoU metric. The results in Table 4.3 underscore the significance of employing the dissimilarity network as a reweighting mechanism, further enhancing the segmentation model’s performance.

4.5 Discussion

In this work, we introduced OVOSE, the first open-vocabulary semantic segmentation algorithm designed for event-based data. Comprising grayscale image and event branches, each equipped with a pre-trained foundation model, our approach leverages synthetic data for Knowledge Distillation (KD) from regular images to enhance semantic segmentation in events. OVOSE employs distillation at multiple stages of the foundation model, enhancing its effectiveness with a mask reweighting strategy through a dissimilarity network. We evaluate OVOSE in the DDD17 and DSEC-Semantic datasets and compare it against with existing methods in UDA close-set semantic segmentation and foundation models adapted to the event domain with E2VID. Our algorithm outperforms all these

models, offering a promising avenue for research in open-vocabulary semantic segmentation tailored for event cameras.

Chapter 5

Final Remarks

In this thesis, we have addressed three distinct areas of research: 1) Classification, 2) Regression, and 3) Segmentation. Each area contributes significantly to the advancement of state-of-the-art techniques within their respective domains.

Firstly, we have tackled the issue of Unsupervised Domain Adaptation (UDA) in visual recognition, aiming to overcome the limitations of manually adapted methods and the inapplicability of existing Neural Architecture Search (NAS) approaches. Our proposed Adversarial Branch Architecture Search for UDA (ABAS) introduces innovative solutions to address these challenges. By employing a data-driven ensemble approach for model selection and searching for an auxiliary adversarial branch attached to a pre-trained backbone, ABAS offers a promising framework for achieving robust domain adaptation without the need for manual intervention or labeled target data.

Secondly, in the realm of regression, we have delved into the relatively unexplored territory of predicting the future poses of interacting individuals. Through our research, we have identified the significance of leveraging body-body motion correlations for improved performance in this task. Our findings reveal the efficacy of certain single-person practices when adapted for 2-body collaborative motion forecasting, including the utilization of frequency input representations, space-time separable interaction adjacencies, and fully-learnable interaction encoding mechanisms. Additionally, we have introduced a novel initialization procedure for spatial interaction parameters, enhancing both the performance and stability of the forecasting model.

Lastly, our work extends to the domain of event-based computer vision, particularly focusing on open vocabulary semantic segmentation for event cameras. By introducing OVOSE, the first Open-Vocabulary Semantic Segmentation algorithm for Event cameras, we have addressed the challenges posed by limited event-based data and the absence of large-scale segmentation benchmarks. Through the integration of synthetic event data and knowledge distillation techniques, OVOSE demonstrates superior performance in open-vocabulary semantic segmentation, showcasing its potential for various applications in autonomous driving.

Overall, our contributions across these three research areas represent significant advancements toward achieving state-of-the-art performance in unsupervised domain adaptation, collaborative human pose forecasting, and event-based semantic segmentation in visual recognition. Through empirical evaluations and theoretical analyses, we have provided valuable insights and methodologies that pave the way for further advancements in these fields.

Chapter 6

Future Work

In this section, we outline several promising directions for future research building upon the foundation laid by our work. These directions aim to address both theoretical advancements and practical applications, to further the field of Automated machine learning for unsupervised domain adaptation, collaborative human pose forecasting, and segmentation in event cameras. From exploring alternative architectures to delving into the interpretability of generated models, each direction offers unique opportunities for advancement and innovation.

1. **Further explore the adversarial branch architecture:** Extend ABAS to more complex scenarios with multiple domains or larger datasets. Investigate alternative branching strategies and loss functions for improved adaptation capability.
2. **Theoretical analysis of convergence and generalization:** Develop deeper understanding of ABAS's convergence properties and its generalization ability to unseen data.
3. **Investigate interpretability of the discovered architectures:** Analyze the characteristics and decision-making mechanisms of the architectures generated by ABAS to gain insights into domain adaptation strategies.
4. **Extend to other learning paradigms beyond UDA:** Explore the applicability of ABAS to other related learning settings like multi-source learning or domain generalization.
5. **Generalize beyond two-person interactions:** Extend the approach to handle larger groups of interacting people.
6. **Explore alternative interaction encoding methods:** Investigate the potential of attention-based or hierarchical approaches for 2-body collaboration, considering their limitations mentioned in the paper.
7. **Incorporate contextual information:** Explore how to incorporate additional context (e.g., scene layout, objects) to improve forecasting accuracy.
8. **Improve synthetic data generation:** Develop more realistic and diverse synthetic event data with richer scene information and varying lighting conditions.
9. **Enhance knowledge distillation techniques:** Explore advanced knowledge distillation strategies that further improve OVOSE's performance and efficiency.
10. **Apply to real-world applications:** Implement and evaluate OVOSE in practical applications like autonomous driving systems, object tracking, and anomaly detection.

While our current work represents a significant step forward in computer vision research, there remains ample room for exploration and refinement. By pursuing these future directions, we envision a deeper understanding and the development of more robust and adaptable solutions. Through continued collaboration and investigation, we are optimistic about the potential impact of these endeavors on both academia and industry.

Bibliography

- [1] I. Akhter, Y. Sheikh, S. Khan, and T. Kanade. Nonrigid structure from motion in trajectory space. In *Advances in Neural Information Processing Systems*, volume 21, 2008.
- [2] E. Aksan, M. Kaufmann, P. Cao, and O. Hilliges. A spatio-temporal transformer for 3d human motion prediction. In *2021 International Conference on 3D Vision (3DV)*, 2021.
- [3] I. Alonso and A. C. Murillo. EV-SegNet: Semantic segmentation for event-based cameras. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019.
- [4] M. Andriluka, U. Iqbal, A. Milan, E. Insafutdinov, L. Pishchulin, J. Gall, and B. Schiele. Pose-track: A benchmark for human pose estimation and tracking. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5167–5176, 2018.
- [5] J. Atwood and D. Towsley. Diffusion-convolutional neural networks. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, 2016.
- [6] S. Bai, J. Z. Kolter, and V. Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.
- [7] T. Baker and R. E. Nelson. Creating something from nothing: Resource construction through entrepreneurial bricolage. *Administrative science quarterly*, 50(3):329–366, 2005.
- [8] A. Benzine, B. Luvison, Q. C. Pham, and C. Achard. Deep, robust and single shot 3d multiperson human pose estimation from monocular images. In *2019 IEEE International Conference on Image Processing (ICIP)*, 2019.
- [9] J. Binas, D. Neil, S.-C. Liu, and T. Delbrück. Ddd17: End-to-end davis driving dataset. *ArXiv*, abs/1711.01458, 2017.
- [10] M. Bucher, T.-H. Vu, M. Cord, and P. Pérez. Zero-shot semantic segmentation. *Advances in Neural Information Processing Systems*, 32, 2019.
- [11] Y. Cai, L. Huang, Y. Wang, T.-J. Cham, J. Cai, J. Yuan, J. Liu, X. Yang, Y. Zhu, X. Shen, D. Liu, J. Liu, and N. M. Thalmann. Learning progressive joint propagation for human motion prediction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.
- [12] T. Caliński and J. Harabasz. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27, 1974.

- [13] F. M. Carlucci, A. D’Innocente, S. Bucci, B. Caputo, and T. Tommasi. Domain generalization by solving jigsaw puzzles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2229–2238, 2019.
- [14] F. M. Carlucci, P. Russo, T. Tommasi, and B. Caputo. Hallucinating agnostic images to generalize across domains. In *ICCV Workshops*, pages 3227–3234, 2019.
- [15] D. Chen, J.-P. Mei, C. Wang, Y. Feng, and C. Chen. Online knowledge distillation with diverse peers. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 3430–3437, 2020.
- [16] G. Chen, H. Cao, J. Conradt, H. Tang, F. Rohrbein, and A. Knoll. Event-based neuromorphic vision for autonomous driving: A paradigm shift for bio-inspired visual sensing and perception. *IEEE Signal Processing Magazine*, 37(4):34–49, 2020.
- [17] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, 2018.
- [18] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Computer Vision – ECCV 2018: 15th European Conference, Munich, Germany, September 8–14, 2018, Proceedings, Part VII*, page 833–851, Berlin, Heidelberg, 2018. Springer-Verlag.
- [19] M. Chen, S. Zhao, H. Liu, and D. Cai. Adversarial-learned loss for domain adaptation. In *AAAI*, pages 3521–3528, 2020.
- [20] P. Chen, W. Guan, and P. Lu. Esvio: Event-based stereo visual inertial odometry. *IEEE Robotics and Automation Letters*, 8(6):3661–3668, 2023.
- [21] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1290–1299, 2022.
- [22] H.-K. Chiu, E. Adeli, B. Wang, D.-A. Huang, and J. C. Niebles. Action-agnostic human pose forecasting. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2019.
- [23] H. Cho, H. Kim, Y. Chae, and K.-J. Yoon. Label-free event-based object recognition via joint learning with image reconstruction from events. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 19866–19877, October 2023.
- [24] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016.
- [25] R. Dabral, N. Gundavarapu, R. Mitra, A. Sharma, G. Ramakrishnan, and A. Jain. Dynamic multiscale graph neural networks for 3d skeleton-based human motion prediction. In *Multi-person 3d human pose estimation from monocular images. In 2019 International Conference on 3D Vision (3DV)*, 2019.

- [26] L. Dang, Y. Nie, C. Long, Q. Zhang, and G. Li. Msr-gcn: Multi-scale residual graph convolution networks for human motion prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.
- [27] Z. Ding, J. Wang, and Z. Tu. Open-vocabulary panoptic segmentation with maskclip. *arXiv preprint arXiv:2208.08984*, 2022.
- [28] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning*, pages 647–655. PMLR, 2014.
- [29] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.
- [30] Y. Du, W. Wang, and L. Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [31] N. F. Duarte, M. Raković, J. Tasevski, M. I. Coco, A. Billard, and J. Santos-Victor. Action anticipation: Reading the intentions of humans and robots. *IEEE Robotics and Automation Letters*, 3(4):4132–4139, 2018.
- [32] T. Elsken, J. H. Metzen, F. Hutter, et al. Neural architecture search: A survey. *J. Mach. Learn. Res.*, 20(55):1–21, 2019.
- [33] S. Falkner, A. Klein, and F. Hutter. Bohb: Robust and efficient hyperparameter optimization at scale. In *International Conference on Machine Learning*, pages 1437–1446. PMLR, 2018.
- [34] M. Fieraru, M. Zanfir, E. Oneata, A.-I. Popa, V. Olaru, and C. Sminchisescu. Three-dimensional reconstruction of human interactions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.
- [35] K. Fragkiadaki, S. Levine, P. Felsen, and J. Malik. Recurrent network models for human dynamics. In *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015.
- [36] G. Gallego, T. Delbrück, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. J. Davison, J. Conradt, K. Daniilidis, et al. Event-based vision: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(1):154–180, 2020.
- [37] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.
- [38] D. Gehrig, M. Gehrig, J. Hidalgo-Carrió, and D. Scaramuzza. Video to events: Recycling video datasets for event cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3586–3595, 2020.

- [39] D. Gehrig, A. Loquercio, K. G. Derpanis, and D. Scaramuzza. End-to-end learning of representations for asynchronous event-based data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5633–5643, 2019.
- [40] M. Gehrig, W. Aarents, D. Gehrig, and D. Scaramuzza. Dsec: A stereo event camera dataset for driving scenarios. *IEEE Robotics and Automation Letters*, 6:4947–4954, 2021.
- [41] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin. Convolutional sequence to sequence learning. In *International conference on machine learning*, pages 1243–1252. PMLR, 2017.
- [42] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9. PMLR, 2010.
- [43] Y. Grandvalet, Y. Bengio, et al. Semi-supervised learning by entropy minimization. In *CAP*, pages 281–296, 2005.
- [44] I. Gulrajani and D. Lopez-Paz. In search of lost domain generalization. In *International Conference on Learning Representations*, 2021.
- [45] W. Guo, X. Bie, X. Alameda-Pineda, and F. Moreno-Noguer. Multi-person extreme motion prediction. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.
- [46] W. Guo, E. Corona, F. Moreno-Noguer, and X. Alameda-Pineda. Pi-net: Pose interacting network for multi-person monocular 3d pose estimation. In *In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021.
- [47] W. Guo, Y. Du, X. Shen, V. Lepetit, A.-P. Xavier, and M.-N. Francesc. Back to mlp: A simple baseline for human motion prediction. *arXiv preprint arXiv:2207.01567*, 2022.
- [48] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1026–1034, 2015.
- [49] 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.
- [50] S. Herath, M. T. Harandi, and F. M. Porikli. Learning an invariant hilbert space for domain adaptation. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3956–3965, 2016.
- [51] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.
- [52] W. Hong, Z. Wang, M. Yang, and J. Yuan. Conditional generative adversarial network for structured domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1335–1344, 2018.

-
- [53] Y. Huang, H. Bi, Z. Li, T. Mao, and Z. Wang. Stgat: Modeling spatial-temporal interactions for human trajectory prediction. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.
 - [54] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning*, 2015.
 - [55] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339, 2014.
 - [56] D. Jian and M. Rostami. Unsupervised domain adaptation for training event-based networks using contrastive learning and uncorrelated conditioning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 18721–18731, October 2023.
 - [57] Y. Jiang, C. Zhao, Z. Dou, and L. Pang. Neural architecture refinement: A practical way for avoiding overfitting in nas. *arXiv preprint arXiv:1905.02341*, 2019.
 - [58] J. Kim, J. Bae, G. Park, D. Zhang, and Y. M. Kim. N-imagenet: Towards robust, fine-grained object recognition with event cameras. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2146–2156, 2021.
 - [59] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR*, 2015.
 - [60] T. Kipf, E. Fetaya, K.-C. Wang, M. Welling, and R. Zemel. Neural relational inference for interacting systems. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.
 - [61] T. Kipf, E. Fetaya, K.-C. Wang, M. Welling, and R. Zemel. Neural relational inference for interacting systems. In *International Conference on Machine Learning*, pages 2688–2697. PMLR, 2018.
 - [62] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.
 - [63] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. B. Girshick. Segment anything. *ArXiv*, abs/2304.02643, 2023.
 - [64] H. S. Koppula and A. Saxena. Anticipating human activities for reactive robotic response. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2071–2071, 2013.
 - [65] P. Kothari, S. Kreiss, and A. Alahi. Human trajectory forecasting in crowds: A deep learning perspective. *IEEE Transactions on Intelligent Transportation Systems*, 23(7):7386–7400, 2022.
 - [66] P. Krähenbühl, C. Doersch, J. Donahue, and T. Darrell. Data-dependent initializations of convolutional neural networks. *CoRR*, abs/1511.06856, 2016.
-

- [67] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.
- [68] B. Li, K. Q. Weinberger, S. Belongie, V. Koltun, and R. Ranftl. Language-driven semantic segmentation. In *International Conference on Learning Representations*, 2022.
- [69] C. Li, Z. Zhang, W. S. Lee, and G. H. Lee. Convolutional sequence to sequence model for human dynamics. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.
- [70] D. Li, Y. Yang, Y.-Z. Song, and T. M. Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pages 5542–5550, 2017.
- [71] J. Li, E. Chen, Z. Ding, L. Zhu, K. Lu, and Z. Huang. Cycle-consistent conditional adversarial transfer networks. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 747–755, 2019.
- [72] J. Li, F. Yang, M. Tomizuka, and C. Choi. Evolvegraph: Multi-agent trajectory prediction with dynamic relational reasoning. *arXiv: Computer Vision and Pattern Recognition*, 2020.
- [73] L. Li, K. G. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar. Hyperband: Bandit-based configuration evaluation for hyperparameter optimization. In *ICLR (Poster)*, 2017.
- [74] M. Li, S. Chen, X. Chen, Y. Zhang, Y. Wang, and Q. Tian. Symbiotic graph neural networks for 3d skeleton-based human action recognition and motion prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44:3316–3333, 2022.
- [75] M. Li, S. Chen, Y. Zhao, Y. Zhang, Y. Wang, and Q. Tian. Dynamic multiscale graph neural networks for 3d skeleton based human motion prediction. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.
- [76] Q. Li, Z. Han, and X.-m. Wu. Deeper insights into graph convolutional networks for semi-supervised learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018.
- [77] Y. Li and X. Peng. Network architecture search for domain adaptation. *arXiv preprint arXiv:2008.05706*, 2020.
- [78] F. Liang, B. Wu, X. Dai, K. Li, Y. Zhao, H. Zhang, P. Zhang, P. Vajda, and D. Marculescu. Open-vocabulary semantic segmentation with mask-adapted clip. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7061–7070, 2022.
- [79] J. Liang, R. He, Z. Sun, and T. Tan. Distant supervised centroid shift: A simple and efficient approach to visual domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2975–2984, 2019.

- [80] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.
- [81] H. Liu, M. Long, J. Wang, and M. Jordan. Transferable adversarial training: A general approach to adapting deep classifiers. In *International Conference on Machine Learning*, pages 4013–4022. PMLR, 2019.
- [82] H. Liu, K. Simonyan, and Y. Yang. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*, 2018.
- [83] M. Long, Z. CAO, J. Wang, and M. I. Jordan. Conditional adversarial domain adaptation. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31, pages 1640–1650. Curran Associates, Inc., 2018.
- [84] M. Long, Z. Cao, J. Wang, and M. I. Jordan. Conditional adversarial domain adaptation. In *Advances in Neural Information Processing Systems*, pages 1640–1650, 2018.
- [85] M. Long, H. Zhu, J. Wang, and M. I. Jordan. Deep transfer learning with joint adaptation networks. In *International conference on machine learning*, pages 2208–2217. PMLR, 2017.
- [86] M. Lucic, K. Kurach, M. Michalski, S. Gelly, and O. Bousquet. Are gans created equal? a large-scale study. In *Advances in neural information processing systems*, pages 700–709, 2018.
- [87] W. Luo, B. Yang, and R. Urtasun. Fast and furious: Real time end-to-end 3d detection, tracking and motion forecasting with a single convolutional net. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3569–3577, 2018.
- [88] Y. Luo, L. Zheng, T. Guan, J. Yu, and Y. Yang. Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2507–2516, 2019.
- [89] T. Ma, Y. Nie, C. Long, Q. Zhang, and G. Li. Progressively generating better initial guesses towards next stages for high-quality human motion prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.
- [90] W. Mao, M. Liu, and M. Salzmann. History repeats itself: Human motion prediction via motion attention. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.
- [91] W. Mao, M. Liu, M. Salzmann, and H. Li. Learning trajectory dependencies for human motion prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.
- [92] W. Mao, M. Liu, M. Salzmann, and H. Li. Multi-level motion attention for human motion prediction. *International Journal of Computer Vision*, 129(9):2513–2535, 2021.

-
- [93] N. Messikommer, D. Gehrig, M. Gehrig, and D. Scaramuzza. Bridging the gap between events and frames through unsupervised domain adaptation. *IEEE Robotics and Automation Letters*, 7(2):3515–3522, 2022.
 - [94] N. Messikommer, S. Georgoulis, D. Gehrig, S. Tulyakov, J. Erbach, A. Bochicchio, Y. Li, and D. Scaramuzza. Multi-bracket high dynamic range imaging with event cameras. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 546–556, Los Alamitos, CA, USA, jun 2022. IEEE Computer Society.
 - [95] D. Mishkin and J. Matas. All you need is a good init. In *4th International Conference on Learning Representations, ICLR*, 2016.
 - [96] A. Mitrokhin, C. Fermüller, C. Parameshwara, and Y. Aloimonos. Event-based moving object detection and tracking. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–9. IEEE, 2018.
 - [97] A. A. Mohamed, D. Zhu, W. Vu, M. Elhoseiny, and C. G. Claudel. Social-implicit: Rethinking trajectory prediction evaluation and the effectiveness of implicit maximum likelihood estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.
 - [98] A. Monti, A. Bertugli, S. Calderara, and R. Cucchiara. Dag-net: Double attentive graph neural network for trajectory forecasting. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 2551–2558, 2021.
 - [99] A. Nichol, P. Dhariwal, A. Ramesh, P. Shyam, P. Mishkin, B. McGrew, I. Sutskever, and M. Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.
 - [100] M. Nomura and Y. Saito. Multi-source unsupervised hyperparameter optimization. *arXiv preprint arXiv:2006.10600*, 2020.
 - [101] S. Park and N. Kwak. Feed: Feature-level ensemble for knowledge distillation. *arXiv preprint arXiv:1909.10754*, 2019.
 - [102] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
 - [103] Z. Pei, Z. Cao, M. Long, and J. Wang. Multi-adversarial domain adaptation. In S. A. McIlraith and K. Q. Weinberger, editors, *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 3934–3941. AAAI Press, 2018.
 - [104] Y. Peng, Y. Zhang, P. Xiao, X. Sun, and F. Wu. Better and faster: adaptive event conversion for event-based object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 2056–2064, 2023.
-

- [105] P. O. Pinheiro. Unsupervised domain adaptation with similarity learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8004–8013, 2018.
- [106] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.
- [107] M. R. U. Rahman, L. Scofano, E. D. Matteis, A. Flaborea, A. Sampieri, and F. Galasso. Best practices for 2-body pose forecasting. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 3614–3624, 2023.
- [108] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.
- [109] H. Rebecq, D. Gehrig, and D. Scaramuzza. Esim: an open event camera simulator. In *Conference on Robot Learning*, 2018.
- [110] H. Rebecq, R. Ranftl, V. Koltun, and D. Scaramuzza. High speed and high dynamic range video with an event camera. *IEEE Trans. Pattern Anal. Mach. Intell. (T-PAMI)*, 2019.
- [111] P. Ren, Y. Xiao, X. Chang, P.-Y. Huang, Z. Li, X. Chen, and X. Wang. A comprehensive survey of neural architecture search: Challenges and solutions. *arXiv preprint arXiv:2006.02903*, 2020.
- [112] L. Robbiano, M. R. Ur Rahman, F. Galasso, B. Caputo, and F. M. Carlucci. Adversarial branch architecture search for unsupervised domain adaptation. In *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1008–1018, 2022.
- [113] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.
- [114] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014.
- [115] P. J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.
- [116] S. Roy, A. Siarohin, E. Sangineto, S. R. Buló, N. Sebe, and E. Ricci. Unsupervised domain adaptation using feature-whitening and consensus loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9471–9480, 2019.
- [117] B. Ru, P. Esperanca, and F. Carlucci. Neural architecture generator optimization. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2020.
- [118] P. Russo, F. M. Carlucci, T. Tommasi, and B. Caputo. From source to target and back: symmetric bi-directional adaptive gan. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8099–8108, 2018.

- [119] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *European conference on computer vision*, pages 213–226. Springer, 2010.
- [120] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. L. Denton, K. Ghasemipour, R. Gontijo Lopes, B. Karagol Ayan, T. Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022.
- [121] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3723–3732, 2018.
- [122] A. Sampieri, G. M. D’Amely di Melendugno, A. Avogaro, F. Cunico, F. Setti, G. Skenderi, M. Cristani, and F. Galasso. Pose forecasting in industrial human-robot collaboration. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.
- [123] L. Shi, L. Wang, C. Long, S. Zhou, M. Zhou, Z. Niu, and G. Hua. Sgcnn:sparse graph convolution network for pedestrian trajectory prediction. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.
- [124] J. Snoek, H. Larochelle, and R. P. Adams. Practical bayesian optimization of machine learning algorithms. *arXiv preprint arXiv:1206.2944*, 2012.
- [125] T. Sofianos, A. Sampieri, L. Franco, and F. Galasso. Space-time-separable graph convolutional network for pose forecasting. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.
- [126] M. Sugiyama, M. Krauledat, and K.-R. Müller. Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, 8(5), 2007.
- [127] Y. Sun, E. Tzeng, T. Darrell, and A. A. Efros. Unsupervised domain adaptation through self-supervision. *arXiv preprint arXiv:1909.11825*, 2019.
- [128] Z. Sun, N. Messikommer, D. Gehrig, and D. Scaramuzza. Ess: Learning event-based semantic segmentation from still images. In *European Conference on Computer Vision*, 2022.
- [129] M. Tan, B. Chen, R. Pang, V. Vasudevan, M. Sandler, A. Howard, and Q. V. Le. Mnasnet: Platform-aware neural architecture search for mobile. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2820–2828, 2019.
- [130] M. Tan and Q. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019.
- [131] A. Tao, K. Sapra, and B. Catanzaro. Hierarchical multi-scale attention for semantic segmentation. *ArXiv*, abs/2005.10821, 2020.
- [132] Z. Tong, Y. Liang, C. Sun, D. S. Rosenblum, and A. Lim. Directed graph convolutional network, 2020.

- [133] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko. Simultaneous deep transfer across domains and tasks. In *Proceedings of the IEEE international conference on computer vision*, pages 4068–4076, 2015.
- [134] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7167–7176, 2017.
- [135] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio. Graph Attention Networks. *International Conference on Learning Representations*, 2018.
- [136] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5018–5027, 2017.
- [137] B. Wang, E. Adeli, H.-K. Chiu, D.-A. Huang, and J. C. Niebles. Imitation learning for human pose prediction. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.
- [138] J. Wang, H. Xu, M. Narasimhan, and X. Wang. Multi-person 3d motion prediction with multi-range transformers, 2021.
- [139] L. Wang, Y. Chae, and K.-J. Yoon. Dual transfer learning for event-based end-task prediction via pluggable event to image translation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2135–2145, 2021.
- [140] L. Wang, Y. Chae, S.-H. Yoon, T.-K. Kim, and K.-J. Yoon. Evdistill: Asynchronous events to end-task learning via bidirectional reconstruction-guided cross-modal knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 608–619, 2021.
- [141] L. Wang, J. Wu, S.-L. Huang, L. Zheng, X. Xu, L. Zhang, and J. Huang. An efficient approach to informative feature extraction from multimodal data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5281–5288, 2019.
- [142] Y. Wang, B. Du, Y. Shen, K. Wu, G. Zhao, J. Sun, and H. Wen. Ev-gait: Event-based robust gait recognition using dynamic vision sensors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6358–6367, 2019.
- [143] C. White, W. Neiswanger, and Y. Savani. Bananas: Bayesian optimization with neural architectures for neural architecture search. *arXiv preprint arXiv:1910.11858*, 2019.
- [144] X. Wu, Q. Zhou, Z. Yang, C. Zhao, L. J. Latecki, et al. Entropy minimization vs. diversity maximization for domain adaptation. *arXiv preprint arXiv:2002.01690*, 2020.
- [145] Y. Wu, D. Inkpen, and A. El-Roby. Dual mixup regularized learning for adversarial domain adaptation. In *European Conference on Computer Vision*, pages 540–555. Springer, 2020.

-
- [146] Y. Xian, S. Choudhury, Y. He, B. Schiele, and Z. Akata. Semantic projection network for zero-and few-label semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8256–8265, 2019.
 - [147] G. Xu, Z. Liu, X. Li, and C. C. Loy. Knowledge distillation meets self-supervision. In *European Conference on Computer Vision*, pages 588–604. Springer, 2020.
 - [148] J. Xu, S. De Mello, S. Liu, W. Byeon, T. Breuel, J. Kautz, and X. Wang. Groupvit: Semantic segmentation emerges from text supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18134–18144, 2022.
 - [149] J. Xu, S. Liu, A. Vahdat, W. Byeon, X. Wang, and S. De Mello. Open-vocabulary panoptic segmentation with text-to-image diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2955–2966, 2023.
 - [150] J. Xu, L. Xiao, and A. M. López. Self-supervised domain adaptation for computer vision tasks. *IEEE Access*, 7:156694–156706, 2019.
 - [151] M. Xu, J. Zhang, B. Ni, T. Li, C. Wang, Q. Tian, and W. Zhang. Adversarial domain adaptation with domain mixup. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 6502–6509, 2020.
 - [152] M. Xu, Z. Zhang, F. Wei, H. Hu, and X. Bai. San: Side adapter network for open-vocabulary semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45:15546–15561, 2023.
 - [153] S. Yan, Y. Xiong, and D. Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI*, 2018.
 - [154] A. Yang, P. M. Esperança, and F. M. Carlucci. NAS evaluation is frustratingly hard. In *International Conference on Learning Representations (ICLR)*, 2020.
 - [155] Y. Yang, L. Pan, and L. Liu. Event camera data pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10699–10709, October 2023.
 - [156] C. Ying, A. Klein, E. Christiansen, E. Real, K. Murphy, and F. Hutter. Nas-bench-101: Towards reproducible neural architecture search. In *International Conference on Machine Learning*, pages 7105–7114. PMLR, 2019.
 - [157] K. You, X. Wang, M. Long, and M. Jordan. Towards accurate model selection in deep unsupervised domain adaptation. In *International Conference on Machine Learning*, pages 7124–7133. PMLR, 2019.
 - [158] Y. Yuan, X. Weng, Y. Ou, and K. Kitani. Agentformer: Agent-aware transformers for socio-temporal multi-agent forecasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.
 - [159] A. Zela, A. Klein, S. Falkner, and F. Hutter. Towards automated deep learning: Efficient joint neural architecture and hyperparameter search. *arXiv preprint arXiv:1807.06906*, 2018.
-

- [160] H. Zhang, F. Li, X. Zou, S. Liu, C. Li, J. Yang, and L. Zhang. A simple framework for open-vocabulary segmentation and detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1020–1031, 2023.
- [161] S. Zhang, X. Liu, and J. Xiao. On geometric features for skeleton-based action recognition using multilayer lstm networks. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 148–157, 2017.
- [162] W. Zhang, W. Ouyang, W. Li, and D. Xu. Collaborative and adversarial network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3801–3809, 2018.
- [163] X. Zhang, Y. He, N. Brugnone, M. Perlmutter, and M. Hirn. Magnet: A neural network for directed graphs. In *Advances in Neural Information Processing Systems*, 2021.
- [164] H. Zhao, X. Puig, B. Zhou, S. Fidler, and A. Torralba. Open vocabulary scene parsing. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2002–2010, 2017.
- [165] L. Zhao, X. Peng, Y. Chen, M. Kapadia, and D. N. Metaxas. Knowledge as priors: Cross-modal knowledge generalization for datasets without superior knowledge. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6527–6536, 2020.
- [166] S. Zhao, X. Yue, S. Zhang, B. Li, H. Zhao, B. Wu, R. Krishna, J. E. Gonzalez, A. L. Sangiovanni-Vincentelli, S. A. Seshia, et al. A review of single-source deep unsupervised visual domain adaptation. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.
- [167] T. Zhou, S. Wang, and J. Bilmes. Time-consistent self-supervision for semi-supervised learning. In *International Conference on Machine Learning*, pages 11523–11533. PMLR, 2020.
- [168] A. Z. Zhu, L. Yuan, K. Chaney, and K. Daniilidis. Unsupervised event-based learning of optical flow, depth, and egomotion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 989–997, 2019.
- [169] B. Zoph and Q. V. Le. Neural architecture search with reinforcement learning. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.
- [170] Y. Zou, Z. Yu, B. Vijaya Kumar, and J. Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *Proceedings of the European conference on computer vision (ECCV)*, pages 289–305, 2018.
- [171] N. Zubić, D. Gehrig, M. Gehrig, and D. Scaramuzza. From chaos comes order: Ordering event representations for object recognition and detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12846–12856, October 2023.

Appendix A

Appendix

We provide additional analysis, results and implementation details, to further support the claims of the paper and illustrate specific insights. In detail, we present here:

- A more in-depth description of how the Ensemble Model Selection (EMS) module is trained, as well as further implementation details, plots and experimental results regarding it and the different metric it builds on for target accuracy prediction [sec. A.1].

A.1 Ensemble Model Selection (EMS)

EMS is a crucial component of the ABAS pipeline: in the UDA setting the target labels are not available, and NAS cannot be applied to UDA without a way to assess the quality of the proposed solutions. To be more explicit, EMS uses the metrics discussed in section 3.3 to provide feedback on how good a sample is.

A.1.1 Training EMS

In this work we build an ensemble of weak predictors by training a linear regressor on top of label-free metrics which weakly correlate with the target accuracy. To do so, we randomly sampled 200 configurations on each dataset and trained them while collecting 100 snapshots for each run, thus ending with a dataset of 20,000 points. At each snapshot we collected the 6 metrics described in sec. 3.3 thus building a dataset of size 2000×6 on which to train our regressor; note that the pseudo-label metric is only used to assess the best snapshot in a single run, not to compare across runs. All input features were standardized to zero mean and unit standard deviation. To model the regressor we used a linear least-squares model as implemented in Scikit-Learn [102]. As target labels (and thus the corresponding accuracy) is not available in the context of UDA, we trained the regressor on a different dataset and transferred it to the one of interest. Specifically, on Office31 we used the regressor trained on Office-Home and vice-versa. The regressor for PACS was trained on Office31.

A.1.2 Unsupervised performance estimators

Figures A.1, A.2 and Table A.1 (in aggregated form) show how well the predictors, presented in section 3.3 of the main paper, perform. EMS achieves consistently better correlation on average

and future work could investigate if more complex model ensembles could achieve even better performance. Note that since the regressors were only trained on Office31 and Office-Home, Table 2.1 and the corresponding figures are focused on those two datasets.

A.1.3 Implementation of the Diversity metric

The original Diversity [144] is defined as:

$$H(\hat{q}(\mathcal{T})) = - \sum_{k=1}^K \hat{q}_k \log(\hat{q}_k) \quad (\text{A.1})$$

where K is the number of classes. As we use the Diversity as a predictor for estimating performance across settings with different values of K , we simply divide the metric by K as to get an average over the classes:

$$H(\hat{q}(\mathcal{T})) = -\frac{1}{K} \sum_{k=1}^K \hat{q}_k \log(\hat{q}_k) \quad (\text{A.2})$$

Metric	Average corr. with Target Acc.		
	Office31	Office-Home	PACS
Entropy	0.76	0.61	0.57
Diversity	0.76	0.77	0.76
Pseudo-Labels	0.71	0.61	0.68
Source Accuracy	0.42	0.20	0.48
Silhouette	0.70	0.72	0.74
Calinski-Harabasz	0.29	0.03	-0.13
EMS	0.87	0.81	0.80

Table A.1: This table shows how the different metrics introduced in section 3.3 of the main paper correlate with the target accuracy. Note that all metrics, with the exception of source accuracy are computed on the target.

ACKNOWLEDGEMENTS

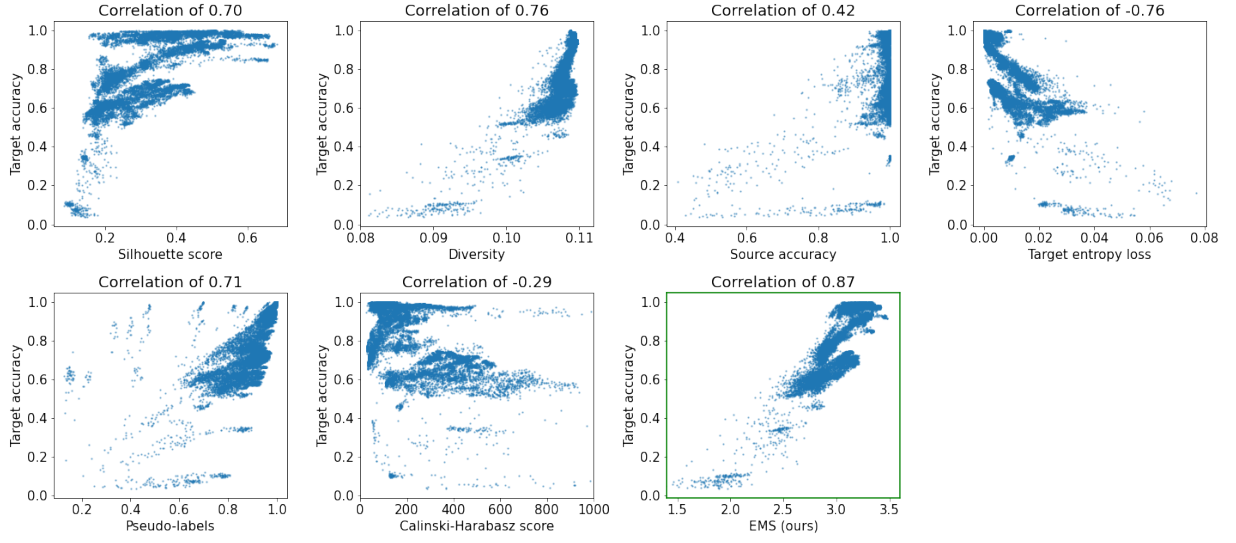


Figure A.1: Correlation between the metrics introduced in section 3.3 and the target accuracy, as computed on Office31.

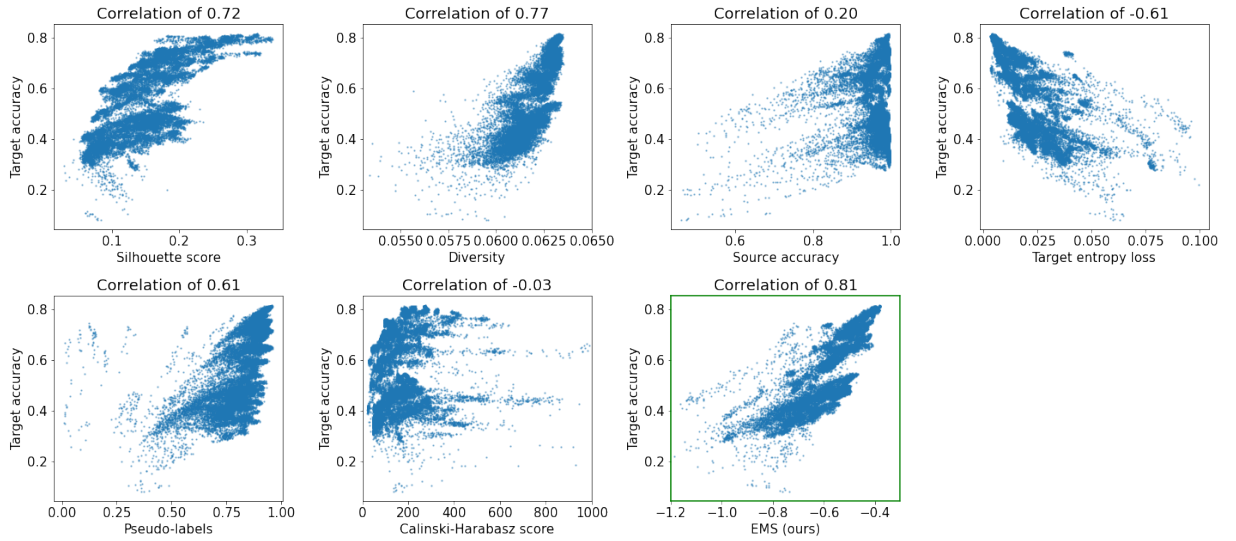


Figure A.2: Correlation between the metrics introduced in section 3.3 and the target accuracy, as computed on Office-Home.

Acknowledgements

I extend my deepest gratitude to my PhD supervisor, Prof. Fabio Galasso, whose guidance, wisdom, and unwavering support have been instrumental throughout this journey. His mentorship has been invaluable, shaping both my academic growth and personal development.

I am indebted to the collaborative efforts of Alessio Sampieri, Alessandro Flaborea, Barbara Caputo, Edoardo De Matteis, Fabio Maria Carlucci, Indro Spinelli, Jhony H. Giraldo, Luca Robbiano, Luca Scofano, Stéphane Lathuilière. Their contributions have enriched the research process and your insights and contributions have been crucial to my work.

I would like to express my sincere appreciation to my colleagues from PinLab for their engaging discussions. With special thanks to Guido Maria D'Amely di Melendugno, for insightful discussions and constructive feedback.

A heartfelt thanks to Guido Maria D'Amely di Melendugno, Chiara Passatore, and Luca Fregonese for their unwavering support and kindness during my transition to Rome. Their generosity and hospitality have eased my adaptation to a new environment and enriched my experience immeasurably.

To my father (Muhammad Latif), mother (Anwar Begum), brothers (Shabbir Hussain, Qadeer Hussain, Touseef ur Rehman), and sisters (Shaista Naz, Shamaila Naz), their unwavering support, encouragement, and understanding have been the bedrock of my journey. Their love and belief in me have sustained me through the challenges and triumphs of pursuing this academic endeavor. I am forever grateful for their presence in my life.

Lastly, I am deeply thankful for the unwavering support of my wife Saba Latif, who has been my pillar of strength throughout this journey.

This work would not have been possible without the collective contributions, guidance, and encouragement of all those mentioned above and the countless others who have supported me along the way.

Muhammad Rameez Ur Rahman, Rome, Feb 2024
