A Visual Analytics Conceptual Framework for Explorable and Steerable Partial Dependence Analysis

Article *in* IEEE Transactions on Visualization and Computer Graphics · April 2023

DOI: 10.1109/TVCG.2023.3263739

4 authors:

Marco Angelini
Sapienza University of Rome
85 PUBLICATIONS **850** CITATIONS
SEE PROFILE

Graziano Blasilli
Sapienza University of Rome
15 PUBLICATIONS **118** CITATIONS
SEE PROFILE

Simone Lenti
Sapienza University of Rome
21 PUBLICATIONS **186** CITATIONS
SEE PROFILE

Giuseppe Santucci
Sapienza University of Rome
183 PUBLICATIONS **2,689** CITATIONS
SEE PROFILE

This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2023.3263739

A Visual Analytics Conceptual Framework for Explorable and Steerable Partial Dependence Analysis

Marco Angelini, Graziano Blasilli, Simone Lenti, and Giuseppe Santucci

Abstract—Machine learning techniques are a driving force for research in various fields, from credit card fraud detection to stock analysis. Recently, a growing interest in increasing human involvement has emerged, with the primary goal of improving the interpretability of machine learning models. Among different techniques, Partial Dependence Plots (PDP) represent one of the main model-agnostic approaches for interpreting how the features influence the prediction of a machine learning model. However, its limitations (i.e., visual interpretation, aggregation of heterogeneous effects, inaccuracy, and computability) could complicate or misdirect the analysis. Moreover, the resulting combinatorial space can be challenging to explore both computationally and cognitively when analyzing the effects of more features at the same time. This paper proposes a conceptual framework that enables effective analysis workflows, mitigating state-of-the-art limitations. The proposed framework allows for exploring and refining computed partial dependences, observing incrementally accurate results, and steering the computation of new partial dependences on user-selected subspaces of the combinatorial and intractable space. With this approach, the user can save both computational and cognitive costs, in contrast with the standard monolithic approach that computes all the possible combinations of features on all their domains in batch. The framework is the result of a careful design process involving experts' knowledge during its validation and informed the development of a prototype, W4SP (available at https://aware-diag-sapienza.github.io/W4SP/), that demonstrates its applicability traversing its different paths. A case study shows the advantages of the proposed approach.

Index Terms—Machine Learning, Partial Dependence Plot, Visual Analytics.

1 Introduction

Predictive models are gaining increasing popularity, satisfying the growing request for machine learning (ML) algorithms that quickly and accurately predict or group data items. A growing research effort focused on the interpretability of such models, positioning human beings within the loop (see, e.g., Floricel et al. [1]). Among the different techniques supporting this research, the Partial Dependence (PD) analysis and its most used visual counterpart, Partial Dependence Plots (PDP) [2], have the goal of explaining how the features affect the output of a machine learning model.

This paper aims to mitigate the four state-of-the-art main drawbacks associated with such an analysis (see Section 2.2 for more details). Indeed, the PD analysis is strongly tied with ad hoc visualizations, and the first issue comes from the inherent difficulty of visualizing the partial dependence of three or more features: while simple visualizations exist for presenting the relationship between one or two features and the output of machine learning model, visualizing such a relationship for three or more features is still a challenging task. A second issue is associated with the aggregated nature of PD: it basically represents means that might hide opposite trends. A third issue arises from the potential inaccuracies caused by the wrong assumptions of independence among the features and inadequate feature sampling. Finally, exploring all the PDs corresponds to exploring the powerset of the features, making an exhaustive analysis impossible in most cases. An additional issue comes from the users' cognitive limits, as reported by Zhao et al. [3]: human beings' visual short-term memory is limited by the amount of information retained, making feasible the visual PD analysis in the range of about three to seven objects [4].

This paper proposes a Visual Analytics (VA) conceptual framework for PD analysis, highlighting the main manual and automatic activities. Such a framework enables effective analysis workflows and provides a concrete guide for exploring partial dependencies, supporting the incremental estimation of influences among features, exploring PDs at different levels of accuracy, and allowing for steering the computation of new PDs to user-defined subsets of the combinatorial analysis space. To illustrate the most significant framework steps, we propose demonstrative solutions covering both analytics and visual aspects. Finally, to foster the framework adoption, we have developed a demonstrative prototype, W4SP, challenging the advantages of its usage in a case study based on a real dataset.

In summary, this paper, considering the four state-of-the-art main PD issues and human beings' cognitive limits, proposes a conceptual framework to foster the analysis of PDs. It helps the user in the exploration of a combinatorial space, providing insights into sub-areas, and supporting hypotheses validation tasks on potential dependencies among features or sub-intervals of them. Overall, it mitigates the state-of-the-art identified issues associated with PD analysis.

The framework is validated through *(i)* a user study involving 11 researchers in machine learning and eXplainable AI, assessing its capability of mitigating the state-of-

• *Authors are with Sapienza Università di Roma, Rome, Italy. E-mail: {angelini, blasilli, lenti, santucci}@diag.uniroma1.it*

the-art identified problems, and *(ii)* a case study using a demonstrative visual analytics prototype, W4SP, that shows the advantages the framework provides in analyzing PDs.

2 Background

A black-box supervised machine learning model is a model which has already been trained and is ready to generate predictions. Common supervised machine learning models are regression models and classifier models. Formally, such kind of model is a function $\mathbf{M}(\cdot)$ that takes in input a matrix $X = [x_1,\ x_2,\ \cdots,\ x_n]$ of n instances (observations and data entries are synonyms) and generates a vector of predictions $Y = \mathbf{M}(X)$ where $Y = [y_1,\ y_2,\ \cdots,\ y_n]$. Each data entry x_i is composed of m features. Different solutions exist in the field of model-agnostic methods [5] for interpretable machine learning, which are commonly appreciated for their flexibility and reusability [6]. Model-agnostic methods can be further distinguished into global and local methods: while the former describes how features affect the prediction on average, the latter aims to explain the influence on the prediction of individual instances. *Partial Dependence Plot* (PDP) [2] is likely the most adopted global method. Among local methods, *Individual Conditional Expectation* (ICE) [7], *Local interpretable model-agnostic explanations* (LIME) [8], and *Shapley Additive Explanations* (SHAP) [9] aim to describe how the instance's prediction changes when features change. Local models, like LIME, are very promising. However, according to Molnar [6], these methods are still in development and could present issues that should be carefully considered during their usage.

2.1 Partial Dependence

Introduced by Friedman [2], the Partial Dependence (PD) measures the marginal effect that one or more features have on the predicted outcome of a machine learning model. It is one of the most used model-agnostic methods [6], [10], presenting a variety of implementations [11], [12], extensions [7], [13], and applications [14], [15], [16], [17], [18], [19], [20]. The PD function is computed using a trained model, a dataset X (in general, the training set [6]), and a subset $T = \{f_1,\ f_2, \cdots\}$ of target features, that is, the features we are interested in. Subset R contains the remaining features. The cardinality of T represents the order of the PD: first-order when T contains one feature, and so on.

The partial dependence of feature f at a particular value v_f is computed by forcing all the data entries in X to assume the value v_f for the feature f, and then averaging the predictions, see Fig. 1. In general, the PD is represented as a curve, a *Partial Dependence Plot* (PDP), computed for a set of grid points spread over the domain of the features, see Fig. 2. Several approaches can be used to set up the grid: take all the unique values in the training dataset, take k equispaced values on each feature domain, or take k values based on the percentiles. Formally, the first-order partial dependence of feature f is $PD_f(v) = \frac{1}{n}\sum_{x \in X} \mathbf{M}(x)$ where $\forall x_{i,f} : x_{i,f} = v$. The curve of a single data entry is called Individual Conditional Expectation [7]; the PDP is the average of all the individual curves, see Fig. 2. It is intuitive to interpret a partial dependence; it shows how the average prediction changes when features' values change. It expresses causality between the features and the prediction, showing the model outcome as a function of the features [21].

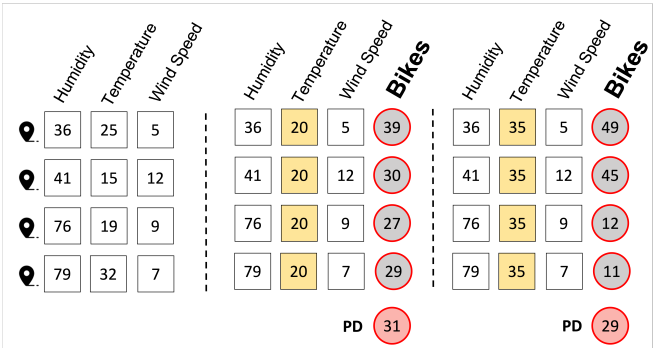


Fig. 1: Computing PD for an ML model that predicts the number of bikes rented daily in a city based on three features: *humidity*, *temperature*, and *wind speed*. Each tuple (left) represents a city. In the center, we set *temperature* to 20 for all the tuples, and the prediction on the transformed dataset shows $PD = 31$ (i.e., the average prediction of the tuples). Then we set *temperature* to 35, and the average prediction remains almost similar $PD = 29$, while the tuples predictions show high variability.

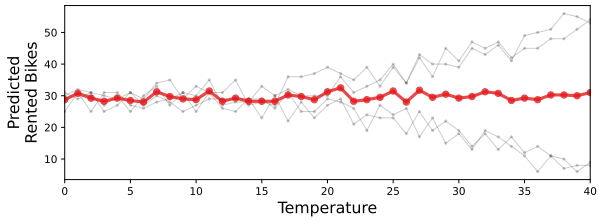


Fig. 2: Individual Conditional Expectation (ICE) and Partial Dependence Plot (PDP) for the feature *temperature* in the scenario of Fig. 1. Due to its almost flat PDP (red line), *temperature* does not appear to affect the model output. The PDP is hiding heterogeneous effects: when *temperature* exceeds 20 degrees, the ICE curves (gray) show two different behaviors.

2.2 Partial Dependence Issues

As discussed in the Introduction, a PD analysis is hindered by drawbacks that make its usage less effective. In this Section, we discuss them more formally and provide some examples.

Issue IS1: visual interpretation. This problem, see, e.g., [2], [6], [15], [16], relates to existing limitations in visualizing the partial dependence of more than two features. While first-order PDs are usually represented using a line chart, and second-order PDs are typically rendered using heatmaps, for higher-order PDs no visualization technique scales well enough. Moreover, the cognitive effort required for interpreting the effects of the feature(s) rises fast, e.g., Molnar [6] states that the realistic maximum number of features in a partial dependence function is two. This is not a fault of PDs, but of the bi-dimensional representations (paper or screen) and the humans' inability to deal with more than three dimensions.

Issue IS2: aggregation of heterogeneous effects. This issue, see, e.g., [6], [7], [22], [23], [24], arises from the PD computation. Due to its averaging steps, the final trend of a PD can mask the effects of subsets of the dataset that show opposite behaviors. Indeed, a flat PDP curve can indicate that *(i)* either the feature does not influence the prediction or *(ii)* subsets of the dataset show opposite trends and cancel each other when computing the average. For example, in Figure 2, the plotting of single tuples shows two subsets of cities: one

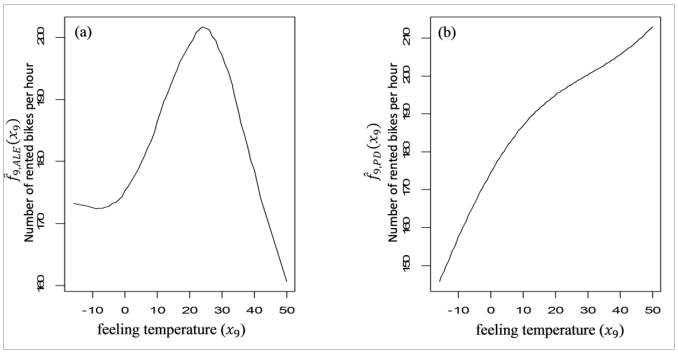


Fig. 3: The PDP on the right shows an unrealistic number of predicted rented bikes for very high $feeling\ temperatures$. The source of the error is that $feeling\ temperature$ is strongly correlated with other features (e.g., temperature and humidity). Taking into account these correlations produces the more realistic Accumulated Local Effects (ALE) plot on the left: the plot has a maximum at 26°C and decreases for very higher temperatures. The picture is from [25] that introduces the ALE technique.

in which high temperatures lead to a prediction of a high number of rented bikes and another one with an opposite trend. The two subsets balance each other, and the final PDP is a flat line, suggesting no influence of temperature and hiding the behavior of the two contrasting subsets.

Issue IS3: inaccuracy. A PD computation could be less or more accurate, see, e.g., [6], [14], [16], [17], [25], [26], being the sources of inaccuracy *(i)* the wrong assumption of independence among considered features, or *(ii)* the sampling step used to compute the partial dependence, or *(iii)* the choice of the sampling points as stated by Krause et al. [17]. Additionally, a higher-order PD could not confirm the lower-order behavior due to the effects that the new considered features bring in. As an example of issue *(i)* is visible in Figure 3 that shows, on the right, a quite unrealistic prediction of rented bikes for very high feeling temperatures.

Issue IS4: computability. Exploring all the possible PDs orders implies browsing the powerset of the features, making it infeasible in most cases. In general, also considering PDs with orders higher than two can become very costly pretty quickly. This is a strong limitation that usually, paired with issue IS1, limits the analysis to just two features at a time.

3 Related work

Understanding the behavior of a machine learning model and allowing the interpretation of its results are research topics on which the eXplainable Artificial Intelligence (XAI) [27], [28] area has focused in recent years. Two main directions can be followed in conducting those analyses: model-agnostic [29], [30] and model-specific approaches [31], [32]. PDP is one of the most used model-agnostic techniques in XAI. Interestingly, their degree of application varies among the XAI contributions, where they usually are simply plotted to provide static evidence concerning model behavior. For example, Chipman et al. [33] propose a new machine learning model called BART and relies on first-order PD to explain such a model when used on a simulated dataset. Green and Kern [34] use PDP to understand the relationship between predictors and the conditional average treatment effect for a voter mobilization experiment, while Berk and Bleich [35] exploit it to model predictor-response relationships in the criminal justice field.

3.1 Existing mitigations for PD issues

As discussed in Section 2.2, the usage of PDP is hindered by a set of drawbacks that make its usage less effective. Starting from these issues, we describe several works that tried to mitigate one or more of them, classifying them by the predominant issue on which they focus.

Visual interpretation. Only a few works coped with the visual interpretation problem because, at low PD orders (specifically at first order), its effects are limited even if still present. The works [3], [19], [20] limit the analysis to first-order PD and use the classic line chart based representation.

Differently, Krause et al. [17] propose a visual analytics approach that shows a partial dependence as a 1D heatmap, in which the color encodes the predicted outcome of the model. Finally, Collaris and van Wijk [36] propose local and global versions of a novel plot called contribution-value plots. While engaging in mitigating the visual interpretation issue, this contribution addresses somehow unrelated issues, as stated by the authors ("*We argue PDP and LCV plots serve a different, and complementary, purpose*"). Additionally, differently from our approach, the proposed plots require an extensive pre-computation that does not allow them to be used in real-time, suffering from additional computation requirements due to plot rendering.

Aggregation of heterogeneous effects. To mitigate the aggregation of heterogeneous effects of observations, Moosbauer et al. [14] model an uncertainty estimate for it and drive sub-grouping by this estimate. We encompass this approach as one of the steps of our conceptual framework hinting at the development of a new research direction on finding more reliable PDP estimates by analyzing subgroups of observations. We contribute to this approach by introducing a novel grouping criterium based on PD trends instead of data tuples characteristics. Grömping [22] presents a different approach for grouping based on correlated features, proposing a stratified version of PDs useful for the visualization of interactions among correlated features.

Inaccuracy. Apley et al. [37] report the potential inaccuracy of PD plots. The authors state that PD plots can produce erroneous results if, for example, some predictors are strongly correlated. To mitigate this problem, they propose Accumulated Local Effects (ALE) plots, a variation of PD plots that mitigates their inaccuracy and aggregation of heterogeneous effects, allowing the identification of accurate local areas. The conceptual framework follows a similar approach, helping the analyst to identify accurate/inaccurate areas, complementing it with the possibility of computing higher-order PDs to confirm or deny their accuracy.

Computability. Wexler et al. [19] propose "What-if Tool", a visual analytics environment that allows for probing and assessing machine learning models, supporting local and global partial dependence analyses. However, they only consider first-order PD (computability issue). We overcome the first limitation by allowing to compute higher-order PD. Due to this, we cope with the visual interpretation issue

by providing an encoding that clearly communicates the discrete nature of the analysis. Kwon et al. [20] follow a similar approach (partial dependence curves). It suffers from the same problems, but it exploits PD as an explanation means for diagnostic risk prediction targeted at medical personnel instead of machine learning architects. It has as an additional constraint a lower cognitive threshold that leads to focus only on single features.

Finally, focusing on combinations of mitigations, some Visual Analytics solutions coped with computability and visual interpretation problems. Krause et al. [17], propose Prospector, a visual analytics system that provides an interactive partial dependence analysis of features for machine learning predictive models. In addition, the system supports localized inspection of data points to understand how and why specific data points are predicted as they are. The user can change the feature values and understand how the actual prediction changes. The system also suggests what feature value should be changed to obtain a certain desired outcome. However, Prospector can only model changes along one feature at a time, identifying computability issues for higher-order PDs that hinder interactivity. Differently from this work, we propose a conceptual framework to allow the computation of higher-order PD by steering the analysis in sub-areas of interest, exploiting lower-order PD information that we help assess the accuracy. We follow a similar visual solution (color heatmaps) for the partial dependence visualization, adapting its behavior to higher-order PDs. Similar considerations are valid for iForest by Zhao et al. [3]. iForest is a visual analytics solution for interpreting random forests in which the authors strongly use PDPs represented as line charts joined by a second aligned chart plotting data distribution. However, this work too is focused on single-feature importance and first-order PDs.

3.2 Existing conceptual frameworks for PD analysis

The previously discussed issues, particularly IS3 (Inaccuracy) and IS4 (Computability), are strongly related to how a PD analysis is conducted in the state-of-the-art. Not many works coped with the problem of designing frameworks to mitigate them, focusing more on the latter.

Molnar framework. The classic approach to conducting partial dependence analysis, introduced by Molnar [6], reports a framework with a monolithic workflow, in which the analyst *(i)* fits an ML model and *(ii)* computes first-order PDs sequentially. Then as a batch, *(iii)* evaluates the results (identifying interesting combinations of features), and *(iv)* proceeds to compute second-order PDs (the higher the order, the more the needed time due to the combinatorial number of dependencies). The suggested workflow ends at the second-order PDs due to computational and cognitive effort limitations. Eventually, all the obtained results are presented in the same order (e.g., all the interesting features are presented at first-order first and at second-order later, while a mix of them is not available).

Tamagnini framework. Tamagnini et al. [38] propose a conceptual framework, a workflow, and a visual analytics interface to enable analysts to understand the causes behind predictions of binary classifiers by interactively exploring a set of instance-level explanations. Rivelo is based on a technique that creates artificial instances derived from observed values to examine the features' influence on the output. The user is able to select features, select the explanation, inspect the descriptor vectors, and explore related raw data as composing steps of the workflow. By switching back and forth among these steps, the user can understand the influence of a feature on the output (partial dependence). Even if this workflow allows analyses more interactive than the ones proposed by Molnar, it still considers only first-order PDs and their disaggregated instances. It does not consider the inaccuracy of the hypothesized dependence or compute higher-order PDs.

Both Molnar and Tamagnini proposals use classic analytical steps in their framework while proposing novel ways to navigate these steps through their proposed workflow. For this reason, we will refer to their proposals as workflows in the rest of the paper, identifying their novelty upfront using the most representative term. On the contrary, our proposed conceptual framework also introduces novelty in the composing steps. It exploits a progressive analysis at different PDs order for *(i)* confirming a PD accuracy at the smallest order possible (helping visual interpretation due to less complex visual representation needed), *(ii)* identifying areas of inaccuracy *(iii)* limiting the computation of higher-order PDs only for those inaccurate areas, where the higher-order can be higher than the state-of-the-art two features. Our framework is based on *(i)* prioritizing the feature analysis, *(ii)* increasing the PD computed order, and *(iii)* supporting an incremental computation of PDs, managing the trade-off between the analyzed data subspace(s), the number of considered features, and the quality gain from computing a higher-order PD.

4 Exploring, Interpreting, and Steering Partial Dependence

This section presents the proposed conceptual framework for explorable and steerable partial dependence analysis. We first identify the main tasks of a PD analysis and then describe the design of the eight steps composing the framework and their relations to existing PD workflows and PD issues.

To design the proposed conceptual framework, we collected the analyst tasks, extracting them from the two state-of-the-art existing workflows (Section 3.2) and considering the goals of the PD analysis itself. We extracted three main tasks reported below showinh how they mitigate one or more of the four PD analysis issues (IS1, IS2, IS3, IS4).

Task T1: identify interesting features that potentially can impact the model output. This task is extracted as an intermediate goal of every PD analysis (the final result is to find a minimum set of features) and is also present in the Tamagnini workflow. The Molnar workflow supports this task statically by simply hinting at the effects of correlation. In contrast, the Tamagnini workflow puts the user in charge of selecting a subset of interesting features. If supported correctly, it allows mitigating IS4 (computability) by reducing the number of features to investigate. The dual task is to give less priority to features for which there are strong hints that they do not affect the model output. For both cases, human feedback

is deemed necessary because it allows forming the feature subset considering multiple criteria or strategies instead of a single one and reducing its cardinality, also projecting domain expertise that purely automatic approaches cannot capture.

Task T2: identify the strength of a partial dependence (how much a set of features affects the model output). Again, this task is one of the goals of the existing workflows, and it concerns the interpretation of the visual representation of the n-th-order PD. The higher the order, the more cognitive effort is required to interpret the result, and for this reason, the state-of-the-art solutions do not go above the second order. Due to computability limitations, the inclusion of human-in-the-loop can help focus the available computation resources on interesting sub-intervals and accept a PD analysis result at different levels of accuracy for different combinations of features, effectively allowing to raise the order of PD analyses above the second order. Given this capability, this task must rely on visual encodings capable of conveying the relations among more than two features, allowing the engaged analyst to make sense of the visualized partial dependence. If correctly supported, this task allows mitigating issue IS1.

Task T3: evaluate the accuracy of the partial dependence, in terms of how confident the analyst is of the result of an n-th order partial dependence. Molnar and Tamagnini workflows that incorporate human feedback at the conclusion of the analysis already support this task. However, they allow looking only at the same order PD analysis, where the higher-order ones require a complete recomputation for the first and are only analyzed at the first-order for the second. In the Molnar workflow, the knowledge of a second-order PD can inform the analyst of the goodness of the previously computed first-order PD, but at the cost of a long batch computation (the human is not inserted in the analysis loop). The Tamagnini workflow supports this task by allowing the analyst to reason on the first-order PD, eventually looking at additional information on raw data or single instances. Including human feedback during the analysis can better implement this task. If correctly supported, this task allows mitigating issues IS2 and IS3.

4.1 Conceptual Framework

The extracted tasks are also related to the Multi-Level Typology of Abstract Visualization Tasks from Brehmer and Munzner [39]: T1 and T2 to the query/identify category and T3 to the consume/discover/generate/verify. Differently from the existing workflows, we propose a conceptual framework to explore PD analyses and incrementally construct the results, allowing the final results to be composed of different orders PDs for different features (e.g., the model output o is mainly affected by three features, f_1, f_2, f_3, and the effect of f_1 is well described at the first order PD, while f_2, f_3 are well described by their joint-effect trough a second-order PD). To obtain this result, the conceptual framework must allow discerning the level of accuracy of the computed PDs. The analyst chooses the accuracy level by narrowing the exploration space (combinatorial in its full extent) by exploiting three main actions:

- excluding features for which their PD is already accurate (pruning the portion of the exploration space resulting from considering their higher-order PDs);
- prioritizing features (e.g., considering together correlated ones and focusing the analysis first on the most promising ones);
- narrowing observations cardinality by focusing on homogeneous sub-groups (e.g., considering only sub-intervals of the domain of a feature, where the accuracy is low, for computation of a higher-order PD).

The resulting conceptual framework is more general and versatile than the existing workflows, which still allows replicating their phases but adds the possibility to scale to higher-order PDs through user intervention during the process. The full conceptual framework is composed of eight steps, visible in Fig. 4. None of these steps are mandatory, leaving the possibility to work on reduced versions of the conceptual framework, supporting different workflows every time, even during the same execution that iterates more time on different steps. In the following, we describe every single step and link it with the effects it has on supporting Tasks T1-T3 and mitigating issues IS1-IS4.

Step S1: feature(s) prioritization. When the analysis is not pre-oriented on specific features, identifying the most significant ones can be challenging due to their potentially high number. This issue calls for proper prioritization strategies to sort the features. This step is mandatory for supporting task T1. The proposed conceptual framework proposes implementing it by ranking first interesting features that could affect the model output. It also supports the possibility of filtering out from the analysis features that occupy low positions in the resulting ranked order. This step provides benefits for issue IS4 (computability) due to the reduced number of features to consider in the PD analysis. At the end, a set of candidate features are selected to continue the analysis. The nature of this step is mainly automatic, allowing the user to select different prioritization strategies.

Step S2: feature(s) behavior interpretation. After identifying interesting features, the analyst focuses on interpreting their PDs, looking for confirmation of their influence on the model output. At the beginning of the analysis, a visual representation of all the first-order PDs for the selected features is available due to the low computational cost. The distribution of the dataset used for computing the PD can be projected on the PDP to inform the analysis, avoiding overestimating regions with sparse data and underestimating dense ones. Alternatively, distribution information can be used to define the representative feature values to compute the PD; the sampling rate of the values along the domain thus suggests the distribution of the data. In any case, the data distribution should be considered to correctly interpret the PDP and identify possible issues in the dataset (e.g., the dataset distribution is not representative of the real one).
This step is mainly human-based, creating the first VA cycle joint with Step S1. It supports task T2 and mitigates the issue IS1 (visual interpretation) if supported by an effective visual encoding. After this step, the analyst can form hypotheses, that still need to be validated, on the effects of single features (at first-order) or multiple features (at higher orders) on the model output.

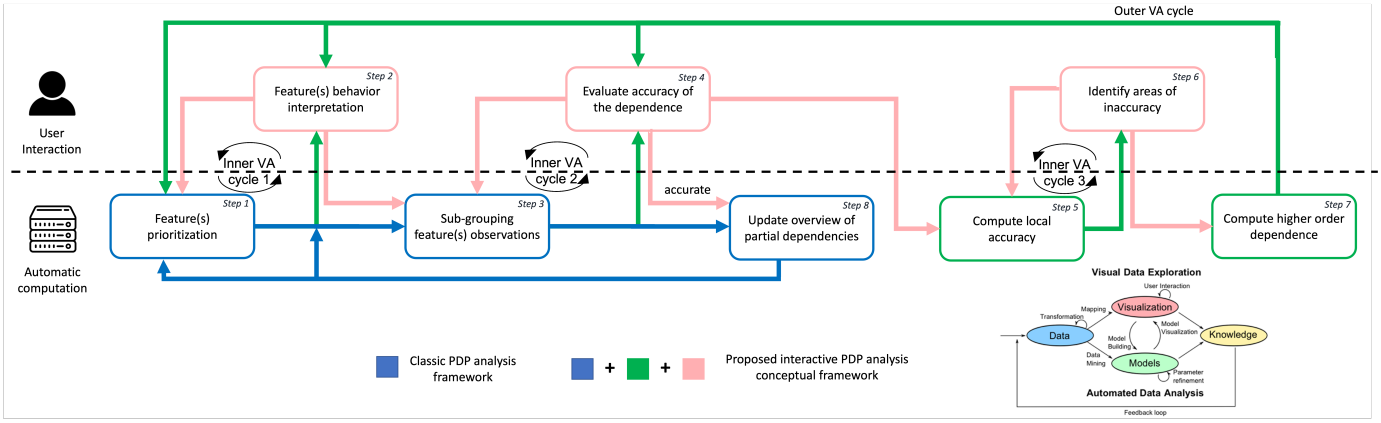


Fig. 4: The conceptual framework for explorable and steerable partial dependence analysis. The steps covered by existing workflows are reported in blue, while the novel steps are represented in green (automatic) or red (human-based), following the color scheme of the VA model.

Step S3: sub-grouping feature(s) observations. This step allows the analyst to subgroup instance-based contributions to the PD to understand better how much a PD represents the general behavior of the feature(s) in influencing the model output. Different strategies can be applied for the sub-grouping, e.g., based on data characteristics or PD trends. In Section 5 we propose a novel way of sub-grouping based on the instances' PD trend observed at S2, allowing to group instances that show a similar trend in the model output. This step supports task T2 and partially task T3. Concerting T3, this step allows to understand if the instance-based contributions reflect the averaged behavior of the PD. For example, a low presence of sub-groups means that a dominant group covering most of the data distribution exists. Conversely, in the case of many (highly populated) different groups, their contributions could be lost due to the elision of contrasting trends.
This step is mainly automatic and allows for mitigating the issue IS2 (aggregation of heterogeneous effects). In the end, the analyst can identify features whose PD is a good representer of the influence and features for which a finer analysis of sub-groups is needed.

Step S4: evaluate the accuracy of a partial dependence. This step is fundamental to steer the analysis narrowing the exploration space (reducing the features to consider). If a PD is considered accurate, the current order of the PD can be considered a good approximation of the higher orders for that feature. This is a new step with respect to existing workflows and supports the iterative and steerable nature of the conceptual framework. At the first-order PD analysis, it would be unusual for an analyst to be able to 100% confirm (by validating it) a feature influence. To confirm it, the PD of a feature should present a high predominant sub-group of instances (or a unique group), and none of the remaining features should be correlated with the analyzed feature. This does not mean that the first-order PD computed is inaccurate but that the analyst may still need to catch all the information needed to confirm its accuracy. To do that, our conceptual framework suggests computing higher-order PDs and using them as a means of validation. Thus, the analyst must proceed to steps S5, S6, and S7. Suppose a higher-order PD (e.g., a second-order PD) confirms the behavior of the lower-order PD (e.g., first-order). In that case, the lower-order is promoted as part of the final result, contributing to S8 (see feature f_1 in Fig. 5) due to it being a good descriptor of the feature influence and a less demanding one to interpret for the analyst.
This step is mainly human-based, forming the second VA cycle joint with step S3. It supports Task T3 and mitigates issue IS3 (inaccuracy).

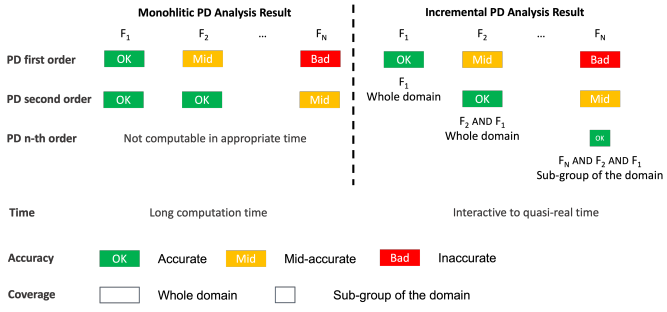


Fig. 5: Comparison between monolithic non-interactive PD analysis (left) and our proposed steerable PD analysis (right) on quality of results, computation, and response times.

Step S8: update overview of PDs. This step envisions the construction of a PD analysis result from one to several iterations over the conceptual framework, depending on the accuracy level evaluated in S4. An example of the incremental construction of the result is visible in the right part of Fig. 5. In the example, it is visible that for feature f_1, the first-order PD is sufficient to describe its influence on the model output (full green rectangle). For feature f_2, instead, it is not enough (full yellow rectangle). Thus a second-order PD_{f_1,f_2} is computed, following S6 and S7. Its accuracy is re-evaluated with a second execution of S4 (eventually priorly executing again also steps S1, S2, and S3 at analyst discretion), and this second-order PD is promoted as part of the result. This step is necessary due to the incremental analysis that our proposed conceptual framework allows, and no other workflow presents this behavior. This step supports tasks T2 and T3 and generally contains the incrementally generated analysis result. It mitigates issue IS1 (visual interpretation) because it minimizes the complexity of visual representation for the analysis results (promoting low-order PD where they are good representatives of the feature influence on the model output), and indirectly IS3 (inaccuracy) and

IS4 (computability), showing the level of accuracy of the different-orders PD composing the result and the subset of higher-than-one order PDs computed (with respect to classic workflows that compute all of them).

Step S5: compute local accuracy. The conceptual framework suggests proceeding to this step if the outcome of S4 is negative for a certain subset of features. For local accuracy, we mean the capability to assess accuracy level for different domain intervals and not just at single feature grain. The conceptual framework mandates computing higher-order PD for the non-conclusive analyzed features and using them to compute a fine-grained local accuracy based on intervals of the feature domain. It relies on sub-grouping results at iteration one, computing its local breakdown on the feature domain. It can exploit higher-order PDs for iterations higher than one and present the result of their comparison. If a higher-order PD including the analyzed feature(s) is present (e.g., we have PD_{f_1} and PD_{f_1,f_2}, as reported in the example of Fig. 4) it is possible to exploit them in comparative analysis. First, we project the second-order PD_{f_1,f_2} on f_1 obtaining $\Pi_{f_1,f_2}(f_1)$. Then, we compare this projected PD with PD_{f_1}, computing where they are similar and where they diverge. This step is mainly automatic and supports task T3.

Step S6: identify areas of inaccuracy. This step aims to allow a finer analysis that distinguishes sub-intervals of the feature(s) domain(s) where the PD analysis is accurate from sub-intervals where it is not. It allows focusing the following computation of higher-order PDs only on the combination of identified inaccurate sub-intervals, reducing the computational cost and allowing scaling better than existing approaches. In doing so, it exploits the results computed in S5, providing them with interpretable visualizations. In the case of iteration one, the analyst can consider inaccurate domain sub-intervals with high divergence in groups behavior (effectively applying a local analysis of what is done globally in S3). Based on the comparison of lower-order and higher-order PDs, the analyst promotes the first-order PD of f_1 as a good descriptor for subsequent iterations if they behave similarly; otherwise, she extracts the subintervals where they diverge for further analysis. Our conceptual framework introduces this step, and no other existing workflow presents it due to their monolithic way of analyzing PDs. Even the Tamagnini workflow, based on visual analytics, considers only the features number reduction (that our conceptual framework implements in S1 and S3) and does not consider a reduction of the feature domains extent. This is a human-based step, forming the third VA cycle joint with S5. It supports task T3, mitigates issue IS3 (inaccuracy), and through its sub-intervals identification, it mitigates also issue IS4 (computability).

Step S7: compute and evaluate higher-order PD. This step computes the higher-order PDs. It represents the computational step that allows to cycle in the conceptual framework (by providing carefully selected higher-order PDs, allowing the analyst to study them at whatever step and using them for evaluating accuracy in S4 and S6) and incrementally constructing the final analysis result.
Due to the described incremental construction of the analysis result and to mitigate issue IS4 as much as possible, the default behavior of this step is to rise by one the order of the new computed PDs every time this step is executed. It is not forbidden to directly move to even higher orders (e.g., moving from first-order PDs to third-order PDs). The method may, however, result in inefficiency both in terms of resource waste (e.g., maybe second-order PDs did not require rising to third-order) and in terms of latency (i.e., while the conceptual framework reduces the exploration space for higher-order PDs, it computes it precisely for a single PD without any approximations, and it requires time for higher-order PDs). As illustrated in Fig. 5 for feature f_n, its first-order PD was entirely inaccurate (full red rectangle), and even its second-order PDs were not resolutive (yellow rectangle), despite accurate areas in half of its domain. For this reason, the third-order PD_{f_1,f_2,f_n} is computed just for half of the f_n domain, resulting accurate. This means that for half of its domain, the second-order PD is an accurate descriptor of f_n influence, while for the other half, the analyst must rely on the third-order PD (green half-rectangle). This step is mainly automatic, indirectly supporting all the tasks, and it helps mitigate issue IS4 (computability).

The conceptual framework can be iteratively traversed until the analyst is confident of the incrementally constructed analysis result during S8. Interestingly, it remains backward compatible with the originating Molnar and Tamagnini workflows. The former can be modeled by considering just S1, S3, and a monolithic version of S8 that always computes all the combinations of a higher-than-one order PD. The latter can be modeled by considering just S1, S2, S3, and S8, given that it does not consider working with higher-order PDs.

4.2 Steering Partial Dependence Analysis

The proposed conceptual framework allows an analyst to steer the PD analysis interactively through the different decisions she can take in steps S2, S4, and S6. They all affect the feature set and features' domain intervals considered in the PD analysis. Overall, they constitute the outer VA cycle intending to steer the PD analysis toward the appropriate trade-off between accuracy and computability. Different choices for alternative parameterizations for those two coordinates can alter the final results and support different insights generation. Additionally, all these analyses can contribute to the incremental generation of the final result of the PD analysis represented in Fig. 5. This overall steerability is then referred to the whole PD analysis and not to a single step of the conceptual framework. All the human-based steps (S2, S4, S6) contribute several parameterizations that then affect the computational steps (S1, S3, S5, S7) according to the definition of computational steering provided by Mulder et al. [40] and Van Liere et al. [41]. Ultimately, the steering effect will be visible in step S8, where the current partial dependence analysis update will be visible. Managing an updatable overview at different levels of accuracy is a new feature not available in any of the existing workflows.

5 Instantiating the Conceptual Framework

To make the proposed conceptual framework operative, design decisions are needed for some of its composing steps. In this section, we briefly describe some of them,

remarking that the conceptual framework is general, so other choices are possible. For S1, we leverage existing strategies from the literature, like the correlation index or the *Partial Dependence Feature Importance* metric (PFI) [13]. Following that, we present the design decisions for the subgrouping strategy (supporting S3), the comparison of PDs (supporting S4), the quantification of similarity between PDs that can be used to identify inaccurate areas (supporting S6), and the visual encoding for the human-based steps of the conceptual framework (supporting S2 primarily but also S6), which is kept at the end because it encompasses the results from several stages. Eventually, the section presents W4SP (Versatile Visual analysis for Steerable Partial Dependence), a demonstrative prototype supporting the conceptual framework, demonstrating its actionability in a real environment.

5.1 Sub-Trends Analysis of PD – STRAP

To instantiate the sub-grouping strategy of S3, we introduce *Sub-TRend Analysis of Partial dependence* (STRAP) leveraging the idea of decomposing a partial dependence curve by grouping single instances curves that present similar behaviors. Each curve, called a subPD, represents a sub-behavior of the model that could not be captured by PD. The creation of subPD curves follows a bottom-up approach. First, single instances curves are computed for the whole dataset; then, they are grouped into several disjoint sub-groups by considering their curve trends similarity. Finally, a subPD curve is created by averaging the predictions of the sub-group observations. Therefore, the PD can be expressed as a weighted sum of its subPDs, where weights are equal to the cardinality of each relative sub-group.

The computation of subPD curves depends on the definition of the relative sub-groups of observations. Several factors can be considered while defining such sub-groups; among them, the type of prediction (e.g., numerical, binary) and the trends shapes play the most important roles. In the following, we present an algorithm for defining sub-groups in the case of a binary classifier and numerical features. We assume that the model generates a prediction that is either 0 (negative) or 1 (positive), and the domain of the feature is numerical. Each feature domain is divided into k uniform bins (k can be different among features), and observation curves are computed over the grid values $\{v_1,\ v_2,\ \cdots,\ v_k\}$. The choice of k is up to the user, while common approaches are the rule of thumbs $k = \lceil\sqrt{n}\rceil$, Sturge's formula [42], Scott's formula [43], or a kernel-density estimation approach [44].

We have identified five sub-trends representing the subPDs of a partial dependence decomposition. Observations belong to a sub-group according to the following criteria:

- **Negative** – Constant curve equal to zero.
- **Positive** – Constant curve equal to one.
- **Increasing** – The curve assumes zero for $[v_1,\ \cdots,\ v_i]$ and one for $[v_{i+1},\ \cdots,\ v_k]$.
- **Decreasing** – The curve assumes one for $[v_1,\ \cdots,\ v_i]$ and zero for $[v_{i+1},\ \cdots,\ v_k]$.
- **Alternating** – The curve does not fit into any previous criteria.

The cardinality of each sub-group represents the magnitude of the sub-trend: the higher cardinality, the more representative the subPD.

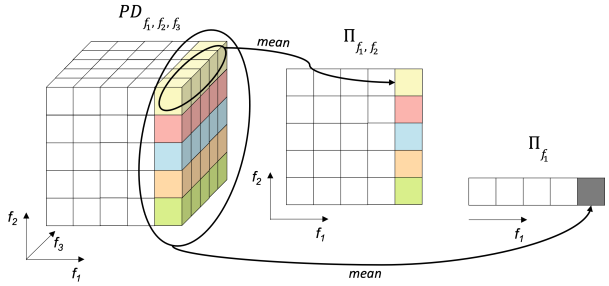


Fig. 6: Projection of a third-order PD_{f_1,f_2,f_3} in a second-order Π_{f_1,f_2} and first-order Π_{f_1}. The projection is the mean of the projected values. Each projection has an associated error quantifiable by using the standard deviations of the projected values: the lower the standard deviation, the more precise the projection.

5.2 Partial Dependence Projection

To allow the comparison between higher-order and lower-order PDs needed for step S4 of the conceptual framework, we propose to project a n^{th}-PD into several low-order projected PD (Π). The properties of this projection technique are to be scalable for increasing order of computed PDs while limiting the cognitive effort required for interpreting the effects of the feature(s) on the model output and managing the comparison between PDs. Fig. 6 provides the intuition of how to compute the first-order Π_{f_1} and the second-order Π_{f_1,f_2} projections from the third-order PD_{f_1,f_2,f_3}. The projection Π_{f_1,f_2} at (u,v) is the mean of the partial dependence values having $f_1 = u$ and $f_2 = v$, i.e., $\Pi_{f_1,f_2}(u,v) = \frac{1}{k}\sum_i^k PD_{f_1,f_2,f_3}(u,v,i)$.

Projection Accuracy. The projection of a partial dependence considerably reduces the exploration space, see Fig. 6, from k^n to k^m elements to analyze when projecting a n-PD to a m-Π. The relative reduction is $(k^n - k^m)/k^n$. The accuracy of the projection, i.e., how much the mean is representative of the projected values, can be evaluated using the standard deviation (σ). A low standard deviation indicates that the projection is representative. As shown in figure Fig. 6, the five yellow values of the 3D-PD are projected into a single value by computing their mean, while the standard deviation indicates the accuracy. For a whole projection, the mean of all the standard deviations (σ_Π) indicates its accuracy.

Quantitative experiment. We conducted a quantitative experiment to quantify, while projecting, the exploration space reduction and the standard deviations in the case study scenario of Section 6, a binary classifier predicting diabetes risk given eight features and $k = 20$. We computed all the 3D-PD and 2D-PD and their possible projections, reporting the results in Table 1. For example, each of the 28 2D-PDs can be projected to two 1D-Π. The exploration space reduction is 95%, while the average σ_Π of all the projections is 0.128 with a confidence interval of ± 0.034. When projecting a 3D-PD into 1D-Π, the exploration space is highly reduced (99.75%), while σ_Π increased. The influence of σ_Π on the projection accuracy depends on the specific model output. For example, in the current scenario, if a projection assumes values ≤ 0.2 or ≥ 0.8, a standard deviation of 0.197 does not affect its accuracy because both the PD and Π show a high classification probability to the negative or positive class. Anyway, the presented conceptual framework relies on the

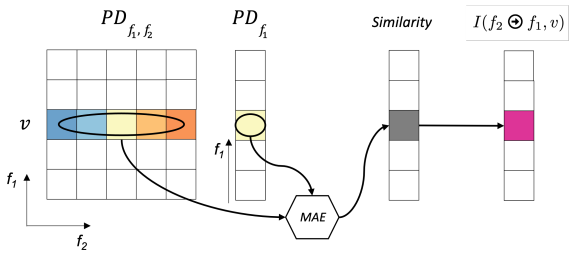


Fig. 7: Similarity between PD_{f_1,f_2} and PD_{f_1}. The mean absolute error (MAE) is used to compare the five second-order PD values, and the first-order one fixed a grid value v. The similarity also allows computing the influence that f_2 has on f_1. The PD_{f_1,f_2} shows an increase of prediction (PD encoding ranges in) when f_2 increases. At the same time, PD_{f_1} shows a medium value (yellow). The two PDs are dissimilar, allowing us to assume that f_2 influences f_1.

human in the loop. Thus, the user, provided with all the necessary information, can understand if the projection is accurate or not and move through the appropriate next steps.

Projection	Count	Exploration Space			σ_Π	
		from	*to*	*reduction %*	μ	CI
2D → 1D	56	400	20	95.00	.128	± .034
3D → 1D	168	8000	20	99.75	.197	± .020
3D → 2D	168	8000	400	95.00	.131	± .018

TABLE 1: Exploration space reduction and average standard deviation and its confidence interval, of all the projections of 2D and 3D partial dependence of the scenario presented in Section 6: a binary classifier based on eight features with $k = 20$ grid points each.

5.3 Different-order PD similarity

To allow the identification of areas of inaccuracy for S6, we introduce a way to quantify the similarity between two different-order PDs. In this way, it is possible to compare an n^{th}-PD with any higher-order one, identifying if the lower-order PD is a good descriptor of the higher-order one, i.e., behaves similarly. The dissimilarity between PD_{f_1} and PD_{f_1,f_2} at grid value v is the mean absolute error (MAE) [45] among the two PDs fixing $f_1 = v$: $\frac{1}{k}\sum_i^k |PD_{f_1,f_2}(v,i) - PD_{f_1}(v)|$. To empathize small values of dissimilarity, it is possible to use a logarithmic-based variation: $\frac{1}{k}\sum_i^k \log(1 + |PD_{f_1,f_2}(v,i) - PD_{f_1}(v)|)$.

As shown in Fig. 7, the (dis)similarity among the PDs also allows for estimating the interaction between two features. The more the two features interact, the stronger the prediction is affected by both. We say that f_2 influences ($\textcircled{\rightarrow}$) feature f_1 if PD_{f_1} and PD_{f_1,f_2} are dissimilar. To quantify this influence, we use the MAE normalized in $[0,\ 1]$, when the model output ranges in $[p_{min},\ p_{max}]$.

$$I(f_2 \textcircled{\rightarrow} f_1, v) = 1 - \frac{1}{k}\sum_i^k \frac{|PD_{f_1,f_2}(v,i) - PD_{f_1}(v)|}{\max[PD_{f_1}(v) - p_{min},\ p_{max} - PD_{f_1}(v)]}$$

If both $(f_2 \textcircled{\rightarrow} f_1)$ and $(f_1 \textcircled{\rightarrow} f_2)$ are high, the mutual interaction between the two features is high, and the prediction is strongly affected by both of them. On the contrary, if $(f_2 \textcircled{\rightarrow} f_1)$ is high and $(f_1 \textcircled{\rightarrow} f_2)$ is low, there is a scarce interaction between the two features, and what we see in the second-order partial dependence depends only on f_2, effectively confirming the first-order PD behavior of f_2.

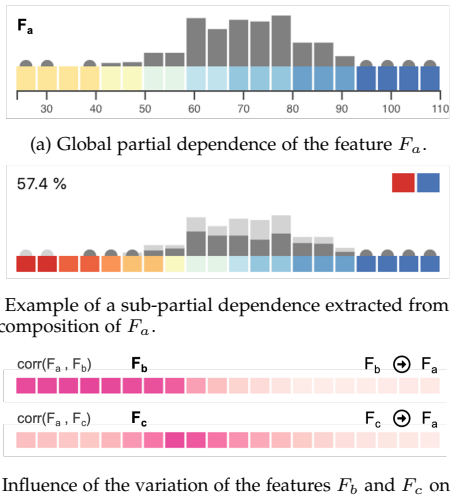
5.4 Designing Visual Encoding

Different factors influence the interpretation of a PD, both when analyzing it individually and as part of a workflow. Usually, the interpretation focuses more on its behavior than its exact values across the domain [46]. For this reason, it is essential that its representation privileges expressiveness in conveying its trend, eventually overlooking precision. The reliability of the partial dependence is strongly affected by the distribution of the observations; areas with few or no observations are generally less reliable because the tuples generated by the dependence calculation are more likely to be significantly unrelated to the real observations. The significance of these tuples is moreover affected by the relation between the feature under analysis and the others. If the feature is correlated with one or more of the others, the probability of including very unlikely or even impossible tuples considerably increases. Considering these elements and cases where the analysis is not pre-oriented to a single feature, it is usually needed to manage multiple features, identify the interesting ones for further analysis, or consider different subsets of them together for mutual influence. In this scenario, it is therefore required to represent multiple dependencies simultaneously, even more, when examining more than one for each feature (e.g., results from sub-grouping) to identify heterogeneous effects. When it is needed to compute higher-order PDs considering two or more dimensions simultaneously, computational factors must also be taken into account; having very dense binning on feature domains, for example, increases the cost considerably.

For those reasons, we propose a small-multiples [47] approach to provide an overview of the PDs and their sub-groups. A first-order PD is usually visualized in literature as a line chart by interpolating its values over the domain. While this encoding effectively conveys the behavior of the dependence, its readability is strongly affected by the choice of the aspect ratio [48], which would be strongly conditioned by the need to show it as a small-multiple. Furthermore, the interpolation hides the binning step leading to a possible overestimation of the accuracy of a dependence computed on few values. Moreover, second-order PDs are usually encoded as two-dimensional heatmaps with an optional overlaid representation of the data distribution, leading to a sharp change in the visual encoding. We propose an alternative visual encoding for first-order PDs based on a one-dimensional heatmap. The color encodes the value of the PD; in the case of a binary classifier, we suggest using a diverging and color-blind safe color scale (e.g., blue-yellow-red,) that implicitly expresses the segmentation into two categories (above and below the threshold). This encoding explicitly represents the binning step and is coupled with a bar chart representing the data distribution. The bar chart is designed to check the height of each bar; if it is below a configurable perceptual threshold, the bar is substituted by a half-circle allowing to distinguish values with no observations from those with few ones. Fig. 8a shows the resulting encoding.

The encoding can be adapted to sub-groups, regardless of the sub-grouping criteria, allowing for easier comparison between the full PD and sub-groups. Furthermore, it enables a preattentive identification of the sub-trends, e.g., clearly distinguishing positive and negative trends. The PD encoding is

This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2023.3263739



(a) Global partial dependence of the feature F_a.

(b) Example of a sub-partial dependence extracted from the decomposition of F_a.

(c) Influence of the variation of the features F_b and F_c on the dependence of the classification from F_a.

Fig. 8: Visual encoding of the PD of feature F_a. The global PD (a) is aligned with the PD of one of its sub-trend (b) and the influences of features F_b and F_c on the global PD (c). The heatmap represents the PD using a diverging color scale ( ). The gray bar chart represents the data distribution, using a half-circle when the value is below a configurable perceptual threshold, allowing to distinguish between zero and small values. In the case of sub-partial dependence (b), the bar chart shows both the distribution of the subset (dark gray) and the whole dataset (light gray). In (c), a sequential color scale ( ) represents F_b and F_c influence.

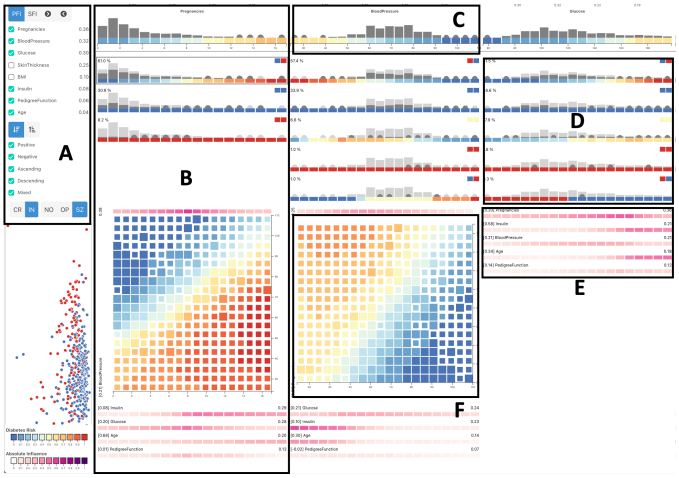


Fig. 9: The W4SP prototype. The controls tab (**A**) allows for sorting features. Each column (**B**) is relative to a feature and shows its first-order PD (**C**) and its decomposition in sub-PDs (**D**). Feature influences are reported at the bottom of each column (**E**). Second-order PDs (**F**) can be plotted on-demand, along with projections of higher-order PDs.

enriched with a gray bar chart representing the distribution of the data in the current group ( ) and in the whole set ( ), enabling the visual comparison of their difference. To provide a measure of the significance of each subgroup, the percentage of data is presented on the top-left, while a small glyph on the top-right quickly recalls the trend (see Fig. 8b).

We exploit the one-dimensional heatmap of the proposed small multiple to encode the different-order PD similarity (used to evaluate accuracy both globally and locally on specific bins). It encodes the influence that the second feature has for each value of the original one using a sequential color scale ( ). Given the visually encoded binning in the heatmap, this means allows for confirming first-order PD accuracy for the whole feature domain or identifying intervals of the domain where the second-order PD shows high variability, requiring further investigation.

This design proposal allows computing PD with more than two features. However, for cognitive effort considerations, we will always visualize the results with maximum two-dimensional visualizations [4] exploiting the projection technique introduced previously. To do that, we enhance the proposed small-multiple, allowing to visualize a bidimensional-heatmap on demand, clicking on the one-dimensional one that visualizes the PD accuracy. Third-order PDs can be computed to evaluate how the contribution of the third feature alters the result, similarly to what is done for the previous step but then re-projected on the bi-dimensional or one-dimensional heatmaps.

5.5 The W4SP prototype

W4SP (Versatile Visual analysis for Steerable Partial Dependence) is a demonstrative visual analytics prototype that supports the proposed conceptual framework and demonstrates its actionability in a real environment. W4SP, see Fig. 9, is composed of the main view, paired with the *Controls Tab* that allows to manage the analysis (Fig. 9.A).

The main view is divided into columns, one for each feature under analysis. Many features call for narrowing down the analysis to the most interesting ones. For this reason, the system allows for sorting them using several criteria, e.g., correlation or PFI, so restricting the analysis to the top ones (S1).

Each column (Fig. 9.B) is related to a feature; columns are sorted according to the selected strategy. On top of each column, the first-order PD (Fig. 9.C) is encoded according to the small-multiple design discussed in Section 5.4 (S2). The different sub-groups of a PD are extracted by using STRAP, and its *sub*PDs (Fig. 9.D) are shown below the global (same encoding), allowing the analyst to focus on the sub-groups (S3). The *Controls Tab* allows sorting the *sub*PDs according to cardinality or trend. The user can deselect groups from the *Controls Tab* to focus the analysis only on specific ones.

The bottom part of a column (Fig. 9.E) allows for evaluating the impact of other features on the predictions. The influence of the other features with respect to the current one, see Section 5.2, is shown as an overall numerical value (right) and a one-dimensional heatmap, as defined in Section 5.4. Those features are sortable according to the two directions of influence. The two features' correlation is shown on the left (S4 and S6). Clicking on a feature triggers the visualization of the second-order partial dependence using a bi-dimensional heatmap (Fig. 9.F). The user can project the data distribution on each heatmap, selecting one of two different encodings, opacity-based or size-based (S2 for higher-order PDs computed during S7). To limit cognitive effort, the user can inspect them on demand: clicking on a heatmap visualizes the PD Projection of the other features sorted by influence. W4SP is available at https://aware-diag-sapienza.github.io/W4SP/, while the paper's appendix presents additional examples of its usage.

6 Case Study

The case study in Fig. 10 demonstrates a workflow of analysis built on the proposed conceptual framework using W4SP. It relies on the PIMA Indians Diabetes dataset [49] and a binary classifier predicting the diabetes risk given eight features: *Age, BMI, BloodPressure, Glucose, Insulin, PedigreeFunction, Pregnancies, and SkinTickness*. The user, aiming at understanding how the features affect the prediction, analyzes the PDs along ten phases.

Phase 1. Since the analysis is not biased towards specific features, the user sorts them using the PFI metric (S1) that identifies *Pregnancies* and *BloodPressure* as the most influencing and *Age* as the least influencing, see Fig. 10.1.

Phase 2. Inspecting *Pregnancies* (S2) highlights an ascending trend while the skewness of the distribution suggests descending confidence of the estimation. *BloodPressure*, conversely, shows a descending trend, and the tuples are more evenly distributed on the domain with a prevalence in the central portion (Fig. 10.2). The almost flat PDPs of *Age* and *PedigreeFunction* suggest their influence is negligible.

Phase 3. A deeper analysis of the sub-groups of *Age* (S3) shows that the flatness of the dependence hides three distinct sub-trends that comprise 37.2% of the observations, as visible in Fig. 10.3. This step suggests that both individually and in relation to the other features, *Age* should not be considered aggregating all the tuples (suffering from IS2) but distinguishing them through the single sub-groups.

Phase 4. The user evaluates the accuracy of the two most influencing features using the available information (S4). *Pregnancies* has only one non-constant sub-group confirming its first-order PDP as a good descriptor beyond the considerations about the confidence in the domain. *BloodPressure* exhibits a similar behavior. At the same time, it has two sub-trends deviating from the global trend (see Fig. 10.4), their incidence (less than 7% of the tuples), and their difference, especially in the portion of the domain with fewer tuples (at the beginning of the domain), makes them almost irrelevant.

Phase 5. The analysis can proceed by calculating all the second-order PDPs containing *Pregnancies* and *BloodPressure* (S7) to confirm the hypothesized single influences.

Phase 6. W4SP, by sorting the second-order PDPs by influence score, allows the user to confirm *Pregnancies* and *BloodPressure* as the top-most influencing features, since they appear at the top of their relative 2D-PDPs lists, mutually influencing each other more than other features. In addition, the system suggests *Glucose* and *Insulin* as the second-most influencing features for both of them (S2bis). The user can now evaluate the accuracy of the first-order PDs, considering the information collected at the higher order (S4bis). She identifies a small area in both *Pregnancies* and *BloodPressure* domains influenced by *Insulin* (S6), see Fig. 10.6. On the contrary, the area influenced by *Glucose* in both *Pregnancies* and *BloodPressure* is wider.

Phase 7. The areas of influence suggest the user to consider the higher-order interactions between the identified features, computing the third-order PDs *Pregnancies-BloodPressure-Glucose* and *Pregnancies-BloodPressure-Insulin* (S7bis).

Phase 8. The influence of *Glucose*, computed using the third-order PD, is projected on the second-order PD *Pregnancies-BloodPressure* (S2tris). The user identifies a small area on the 2D-domain (following the main diagonal) in which *Glucose* greatly influences the classification (see Fig. 10.8), becoming determinant for the tuples that have combinations of values of *Pregnancies* and *BloodPressure* in that area (S4tris, S6bis).

Phase 9. A similar approach is used to evaluate the influence of *Insulin* (S2quater). The area of influence projected on the second-order PD *Pregnancies-BloodPressure* is wider (see Fig. 10.9), highlighting the influence of *Insulin* in the whole area below the secondary diagonal (S4quater, S6tris).

Phase 10. After three repetitions of S8, the user identified the 2D-PD of *Pregnancies-BloodPressure* as a good descriptor of the model behavior, which can be approximated by the 1D-PD of *Pregnancies*. Although the previous phases highlighted considerable influence posed by *Glucose* and *Insulin* over the 2D domain *Pregnancies-BloodPressure*, a review of the data distribution suggests the majority of the area beneath the main diagonal is sparse or empty, suggesting a low confidence interval and allowing the influence area to be reduced. This could push the user to continue the analysis by computing the 4D-PD *Pregnancies-BloodPressure-Glucose-Insulin* but only in the limited area (black square in Fig. 10.10, left), which is less than 10% of the domain. The user concludes that the *Pregnancies* PD, the *Pregnancies-BloodPressure* 2D-PD, and 3D-PD considering *Insulin* in the limited area are a good descriptor of the model behavior.

Advantages for ML users. Using this workflow, the user computed and analyzed 8 (out of 8) 1D-PDs, 13 (out of 28) 2D-PDs, and 2 (out of 56) 3D-PDs (93% of computation time reduction, 2 minutes instead of 32 minutes in our tests). The workflow significantly reduced the cognitive load and the computation time with respect to the monolithic state-of-the-art workflows.

The case study is user-agnostic and directed towards any ML user category [50], [51], such as practitioners (e.g., Architects, Trainers) or end users. Users can benefit from the insights gained in *Phase 10* in a variety of ways, depending on their role and goals: both practitioners (e.g., machine learning experts, data scientists) and end users (e.g., clinicians, domain experts) will benefit from understanding how features affect predictions. Practitioners may want to validate and improve the classifier; for example, understanding which features affect classification more and in what way can provide useful information for tuning hyperparameters. Clinicians and domain experts can use insights to evaluate model accuracy or learn new things. By analyzing the insights produced by the PD analysis, these users can determine whether the model is accurate and responds correctly to variations in a feature. Furthermore, those users could even gain some additional knowledge, such as discovering that under certain conditions, the risk of diabetes decreases.

In the paper's appendix, we also present two usage scenarios addressed to specific user roles. The first usage scenario presents an example of how a practitioner can validate a binary classifier. The second scenario supports clinicians dealing with localized inspections to better understand a

This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2023.3263739

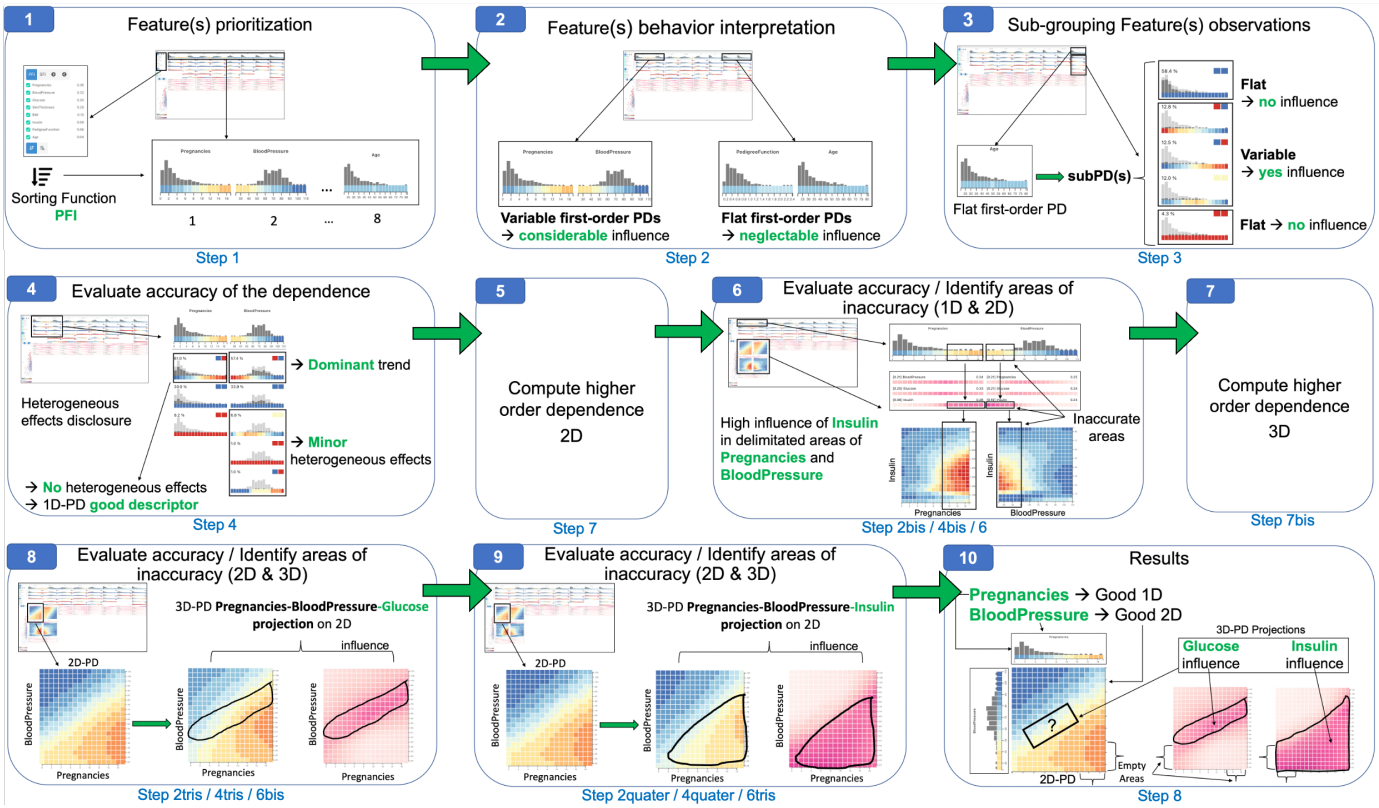


Fig. 10: Illustration of the case study demonstrating a workflow of analysis built on the proposed conceptual framework using W4SP.

patient's clinical condition.

7 Conceptual Framework Validation

We conducted a user study with researchers in machine learning and eXplainable AI to validate the proposed conceptual framework. We used the workflow presented in Section 6 as a representative instance of the conceptual framework in mitigating the four issues of PD analysis and the efficacy of each composing step in providing benefits for the analyst. Section 7.1 presents the method used to conduct the user study, while Section 7.2 discusses the collected results.

7.1 Method

We contacted 11 researchers (6 males and 5 females, mean age 26) actively working in the fields of machine learning and eXplaninable AI as users for validating the conceptual framework. We organized two sessions lasting approximately 2:45 hours (first session: five participants; second session: six participants). A think-aloud method [52] was followed to make all the participants aware of doubts or additional details asked by others. Participants were first asked to provide their expertise in data visualization and Machine Learning using a five levels Likert scale (Expert, Fully knowledgeable, Knowledgeable, Passing knowledgeable, No knowledge). All the participants were fully knowledgeable (5 participants) or knowledgeable (5 participants) about Machine Learning, except one with passing knowledge. Their experience with Data Visualization was various: five participants had passing knowledge, three participants were knowledgeable, two participants had no knowledge, and only one was fully knowledgeable. The users were then exposed to the following activities:

Introduction to Partial Dependence Analysis. In this phase, lasting 15 minutes, the participants were exposed to the general theory of PD analysis, the practices used to conduct this analysis in the state-of-the-art, and the issues that affect the PD analysis. The presented material covered examples and exposed how to use state-of-the-art solutions, both from the interpretation of visual encoding and analysis flow.

Partial Dependence Analysis with classic workflow. In this phase, lasting approximately 1 hour, the participants were tasked to solve a PD analysis using the Molnar workflow and state-of-the-art tools, visual representations provided by literature (e.g., Python Scikit-learn [53]) on the Diabetes dataset [49]. This phase aims to extract quantitative information about the difficulties of conducting PD analysis with classic workflows, relating them to PD analysis issues. For this reason, results were evaluated with Correctness and Easiness metrics.

Tasks provided to them were organized in order of increasing complexity, splitting the classic workflow into three phases (analysis of first-order PDPs, analysis of second-order PDPs, and analysis of a specific third-order PDP). At the end of each phase, we explained the correct expected result for each phase and the final result of the analysis and started a discussion with participants on what have they done and their rationale. At the end of this phase, we verified that each participant understood the correct result of the analysis and what could have potentially impacted her different choices, relating them to issues IS1-4.

Partial Dependence Analysis with the proposed workflow.

We simulated the different phases of the analysis conducted with the proposed workflow (the material was prepared in advance). Having collected quantitative data about the PD analysis difficulty, this phase aims to evaluate how much the proposed conceptual framework steps are perceived as useful and effective in mitigating those problems. We exposed the analysis one step of the conceptual framework at a time, evidencing the outcome of the execution, asking participants what the results communicated to them, and providing differences with respect to the classic Molnar workflow. Due to the participants already knowing the outcome of the analysis at this stage, we no longer used the Correctness metric. We instead used the Utility metric to evaluate user impressions on how much each step of the new conceptual framework would prove helpful in conducting the analysis (global evaluation). We complemented it with the Effectiveness metric to evaluate how much each step is perceived as effective individually (local evaluation). At the end of each step (the complete sequence is: S1, S2, S3, S4, S5, S6, S7, S4bis, S8, S5bis, S6bis, S7bis, S4tris, S8bis, for a total of 14 steps) we asked participants to answer two questions: *(i)* how much they think the presented step would have helped them in solving the analysis, and *(ii)* how did they consider effective the presented analysis step. This phase lasted 1:15 hours, dedicating, on average, 5-10 minutes per step (depending on the step complexity and repetition).

At the end of each activity, we asked participants to complete a questionnaire to collect their feedback. The results of this activity are discussed in the following section.

Final discussion. We dedicated the last 15 minutes to live Q&A on the presented approaches, collecting participants' impressions and comments.

7.2 Results

Classic workflow. The first task was the identification of the two features that most influence the classification according to the classic workflow and using their state-of-the-art first-order PDPs; 7 out of 11 participants were able to correctly identify the two features while the others were able to identify only one of the two (see Fig. 11, 1D-PDP). Three participants summarized their approach to the task by dividing it into two phases: first, they looked at the shape of the plots and then observed the interval between the maximum and minimum values to make the final choice. More than half of the participants consider the independent influence of the two features insufficient to describe the primary influence on the classification for previous knowledge of the domain or for the observed trends of other features that could influence the two selected ones. Afterward, the participants had to evaluate if they confirmed their analysis after including the second-order PDPs. For this task, the results were more varied, with only 4 participants confirming that they came to the same conclusions, 5 to some of them, and 2 to significantly different conclusions (see Fig. 11, 2D-PDP). Some participants conducted their analysis alternating between the first and the second-order plots, while others focused only on the latter. All the second group participants stated that if they had to perform the task again, they would consider first-order plots more carefully, even for the second-order analysis. One participant found it particularly difficult to consider both the mutual influence of the features and the distribution of the observations over their domains. More than half of the participants generally expressed an excessive cognitive load leading to a loss of confidence in the choices. The feedback also confirms this on the perceived ease of performing the tasks. At the same time, 7 participants found the first task easy, and only one participant found the second task easy, with 6 of them finding it difficult or very difficult (see Fig. 11).

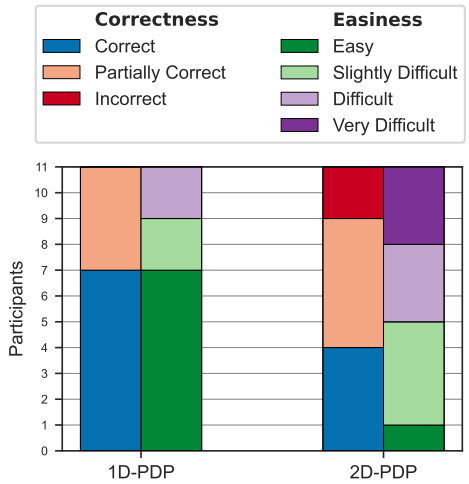


Fig. 11: Correctness and perceived easiness of the 1D and 2D Partial Dependence Analysis performed according to the classic workflow.

Conceptual framework steps utility. The participants evaluated how the proposed steps supported them in performing the previously carried-out analysis. The level of agreement is generally high for all the steps; for each, at least 9 participants responded that it would help them partially or significantly (see Fig. 12). It is worth noting that the step with the lowest agreement was S3 (sub-grouping feature(s) observations); this might be influenced by the fact that the feature on which the contribution of heterogeneous effects was most evident was not one of the most influential; the effect on the conducted analysis was less evident. One participant observed that to evaluate the influence of other features, sorting them using a metric computed on the higher order was not sufficient (S8), but that projection of influence onto the domain of the dimension of interest was needed (S7b).

Conceptual framework steps effectiveness. All the steps are considered effective, at least as how much they were considered helpful in the conducted analysis (suggesting that the participants did not consider them useful just for the presented use case). All the steps were considered effective or partially effective by at least 10 out of 11 participants (see Fig. 12). While all the participants considered S1 at least partially effective, only five of them considered it completely effective. This is supported by the participants' feedback, with many of them pointing out that it can be helpful but that the order of influence could vary depending on considerations arising from the next steps. Conversely, S4 (evaluate the accuracy of the partial dependence) is the step with the highest consensus, with 10 participants considering it as entirely effective for conducting an informed PD analysis.

This article has been accepted for publication in IEEE Transactions on Visualization and Computer Graphics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2023.3263739

This assessment is supported by the accuracy at higher orders (second-order from S8 and S7b, third-order from S8b) that has been considered fully effective by 8 out of 11 participants.

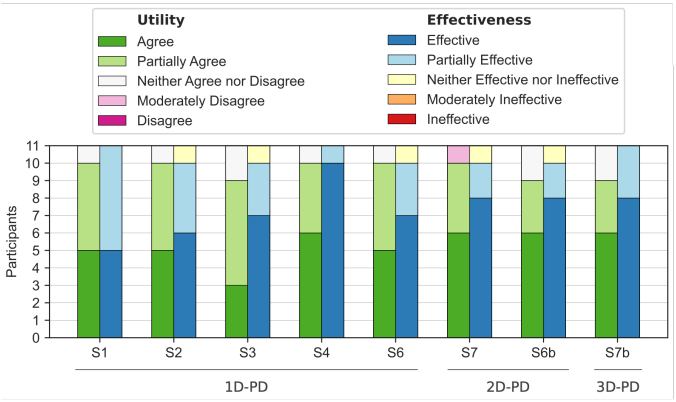


Fig. 12: Utility and effectiveness of the conceptual framework steps.

Mitigating PD Issues. Finally, participants rated the proposed workflow's effectiveness in mitigating the presented issues (Fig. 13). All participants stated that the workflow effectively mitigated all four issues except for some doubts that emerged about IS1. In comparison to other issues, issue IS1 (visual interpretation) received a lower consensus. Meanwhile, one participant commented, *"I find this approach very useful, especially in relation to IS1 and IS4"*, while another said, despite being convinced of the approach direction, he did not have enough evidence to determine whether it was effective in supporting the visual interpretation (IS1) since he had *"[. . .] limited experience with partial dependence analyses and data visualization"*. According to three participants, projecting information from a higher order to a lower order reduces cognitive load, leading to a simpler analysis. The consensus among participants was greater for IS2 (aggregation of heterogeneous effects) and IS3 (inaccuracy). Regarding these issues, one participant found the sub-partial dependencies effective in highlighting different sub-behaviors while another one commented that the projection of the influence from a higher to a lower order dependency to evaluate the accuracy is compelling but that additional means could be explored for it beyond those presented. The issue IS4 (computability) achieved the highest results. Three participants found the ability to reduce the exploration space iteratively as the most convincing result: *"Especially as far as the computational aspects are concerned, the presented workflow is very effective"*, *"I find this approach very useful, especially for what concerns the computability issue"*, *"I think overall this workflow can greatly improve PD Analysis, especially regarding the computability issue"*. In the end, participants were asked whether the workflow provided overall better support for analysis than the state-of-the-art solutions they used. Fig. 13 shows that nine participants agreed and only two partially agreed.

Overall, the validation confirmed the utility and effectiveness of the proposed conceptual framework in mitigating PD issues qualitatively. At the same time, we notice that more quantitative comparative analysis between implemented instances of the proposed conceptual framework and the classic approaches is needed to quantify this approach's benefits further.

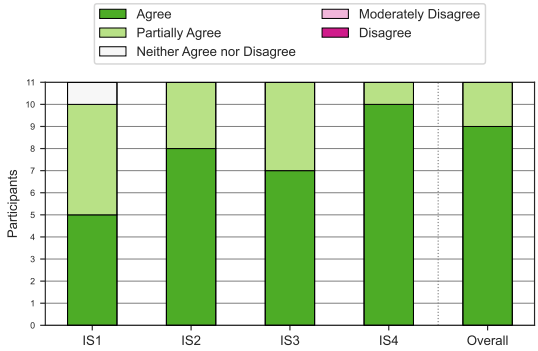


Fig. 13: Conceptual framework effectiveness in mitigating the PD issues.

8 Discussion and Challenges

In this section, we report considerations about potential elements that can be further investigated in future research.

Model and task dependence. We built our conceptual framework to be independent of the model and task (e.g., classification, regression) at hand: it never relies on the intrinsic characteristics of the model (treating it as a black box) or the task. At the same time, the generality of the modeled steps allows for exploiting model-dependent information. While it represents a future research direction to understand how and where the conceptual framework could fit different models, we provide an example for decision trees. In this case, considering a classification task, knowing the split points of the tree for all the features involved could inform feature prioritization strategies at step S1.

Data type dependence. Similar considerations to the previous point apply to data dependence. We tested an instance of our conceptual framework using tabular data, but in principle, this does not constitute a limitation. Using image, video, or multi-modal data is a characteristic of the system at hand (a combination of model, downstream task, and data). The techniques used in steps S1, S3, and S5 can be adapted to other data types while keeping their nature and goal intact.

PD Projection accuracy. In steps S5 and S6, the PD projection plays an essential role in allowing to steer the PD analysis towards inaccurate areas, to compute higher order PDs just for them. This analysis aims to reach an acceptable accuracy level for this estimation. While we outlined in Section 5.2 how to get an acceptable accuracy level, this analysis is at this moment valid just for the binary classification tasks (that was at the base of the proposed case study and evaluation activities). A challenge for further research is how to adapt it to additional tasks (e.g., multi-class classification, regression). Promising approaches that we started investigating could leverage the correlation among features or the dataset density in the feature domain intervals.

Scalability. The proposed conceptual framework, through *(i)* the insertion of human decision points during the analysis and not just at the end, *(ii)* reduction of features to consider, *(iii)* reduction of features domain to consider when computing higher-order PDs and *(iv)* management of a final result composed of different order PDs (instead of the SOTA that produce a result at the same order of PD for all the features considered) allows raising the number of features considered

above two, and nearer the cognitive limit of seven. Considering the recent scalability model in visualization proposed by Richer et al. [54], the proposed framework achieves this goal by reducing the problem size S (through the actions *(i)* - *(iv)*) and the effort E (through higher interactivity and easier-to-interpret visual encodings). While scalability consideration remains valid for a higher number of features, the conceptual framework provides a better granularity on where to focus the optimization, expanding the optimization possibilities instead of the monolithic computation of an n-order PD. Optimization methods to further reduce the computation cost for each step while keeping the analysis interactive are part of future research directions, potentially looking at progressive visual analytics solutions [55], [56], [57].

9 Conclusions

This paper presented a VA conceptual framework for exploring and steering a PD analysis. It mitigates state-of-the-art issues, allowing the evaluation of higher-order PDs incrementally. We instantiated the conceptual framework proposing a set of tailored analytical and visual designs. We evaluated the conceptual framework through a user study confirming its effectiveness both with respect to the existing approaches and in mitigating the collected PD issues. We made actionable the conceptual framework contributing a demonstrative prototype, W4SP; finally, a case study demonstrates its applicability to a real dataset. We identified two possible areas of improvement. The approach accommodates up to 40 features (using a 4k monitor and disregarding cognitive limitations), exploiting the small-multiples visual encoding. It is however possible to reduce the space needed for each of them by using techniques for lower resolution visualization, such as interactive expansion, similar to Table-lens [58]. Designing more sophisticated guidance solutions [59] to further reduce the user cognitive burden is another research direction. E.g., visually aggregating same-order PDs sharing a common set of features to approximate a higher-order PD, will lower the cognitive burden and computational cost.

References

[1] C. Floricel, N. Nipu, M. Biggs, A. Wentzel, G. Canahuate, L. Van Dijk, A. Mohamed, C. Fuller, and G. Marai, "Thalis: Human-machine analysis of longitudinal symptoms in cancer therapy," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 1, pp. 151–161, 2022.

[2] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *The Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001.

[3] X. Zhao, Y. Wu, D. L. Lee, and W. Cui, "iforest: Interpreting random forests via visual analytics," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 1, pp. 407–416, 2019.

[4] T. A. Sørensen and S. Kyllingsbæk, "Short-term storage capacity for visual objects depends on expertise," *Acta psychologica*, vol. 140, no. 2, pp. 158–163, 2012.

[5] M. T. Ribeiro, S. Singh, and C. Guestrin, "Model-agnostic interpretability of machine learning," 2016.

[6] C. Molnar, "Interpretable machine learning," 2022. [Online]. Available: https://christophm.github.io/interpretable-ml-book

[7] A. Goldstein, A. Kapelner, J. Bleich, and E. Pitkin, "Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation," *Journal of Computational and Graphical Statistics*, vol. 24, no. 1, pp. 44–65, 2015.

[8] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should i trust you?": Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16. New York, NY, USA: ACM, 2016, p. 1135–1144.

[9] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, p. 4768–4777.

[10] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: data mining, inference and prediction*. Springer, 2009.

[11] B. M. Greenwell, "pdp: An R Package for Constructing Partial Dependence Plots," *The R Journal*, vol. 9, no. 1, pp. 421–436, 2017.

[12] 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*, vol. 12, pp. 2825–2830, 2011.

[13] B. M. Greenwell, B. C. Boehmke, and A. J. McCarthy, "A simple and effective model-based variable importance measure," 2018.

[14] J. Moosbauer, J. Herbinger, G. Casalicchio, M. Lindauer, and B. Bischl, "Explaining hyperparameter optimization via partial dependence plots," 2021.

[15] J. H. Friedman and J. J. Meulman, "Multiple additive regression trees with application in epidemiology," *Statistics in Medicine*, vol. 22, no. 9, pp. 1365–1381, 2003.

[16] D. R. Cutler, T. C. Edwards Jr., K. H. Beard, A. Cutler, K. T. Hess, J. Gibson, and J. J. Lawler, "Random forests for classification in ecology," *Ecology*, vol. 88, no. 11, pp. 2783–2792, 2007.

[17] J. Krause, A. Perer, and K. Ng, "Interacting with predictions: Visual inspection of black-box machine learning models," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, ser. CHI '16. New York, NY, USA: AACM, 2016, p. 5686–5697.

[18] C. Wang and K.-L. Ma, "HypperSteer: Hypothetical Steering and Data Perturbation in Sequence Prediction with Deep Learning," *arXiv:2011.02149 [cs]*, Nov. 2020. [Online]. Available: http://arxiv.org/abs/2011.02149

[19] J. Wexler, M. Pushkarna, T. Bolukbasi, M. Wattenberg, F. Viegas, and J. Wilson, "The What-If Tool: Interactive Probing of Machine Learning Models," *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–1, 2019.

[20] B. C. Kwon, P. Chakraborty, J. Codella, A. Dhurandhar, D. Sow, and K. Ng, "Visually Exploring Contrastive Explanation for Diagnostic Risk Prediction on Electronic Health Records," 2020.

[21] Q. Zhao and T. Hastie, "Causal interpretations of black-box models," *Journal of Business & Economic Statistics*, vol. 39, no. 1, 2021.

[22] U. Grömping, "Model-agnostic effects plots for interpreting machine learning models," *Reports in Mathematics, Physics and Chemistry, Department II, Beuth University of Applied Sciences Berlin Report*, vol. 1, 2020.

[23] C. Molnar, G. König, B. Bischl, and G. Casalicchio, "Model-agnostic feature importance and effects with dependent features – a conditional subgroup approach," 2021.

[24] M. Britton, "Vine: Visualizing statistical interactions in black box models," 2019.

[25] D. W. Apley and J. Zhu, "Visualizing the effects of predictor variables in black box supervised learning models," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 82, no. 4, pp. 1059–1086, 2020.

[26] X. Zhu, Y. Li, and X. Wang, "Machine learning prediction of biochar yield and carbon contents in biochar based on biomass characteristics and pyrolysis conditions," *Bioresource Technology*, vol. 288, p. 121527, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0960852419307576

[27] B. C. Kwon, M. Choi, J. T. Kim, E. Choi, Y. B. Kim, S. Kwon, J. Sun, and J. Choo, "Retainvis: Visual analytics with interpretable and interactive recurrent neural networks on electronic medical records," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 1, pp. 299–309, Jan 2019.

[28] H. Strobelt, S. Gehrmann, H. Pfister, and A. M. Rush, "Lstmvis: A tool for visual analysis of hidden state dynamics in recurrent neural networks," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 1, pp. 667–676, Jan 2018.

[29] J. Zhang, Y. Wang, P. Molino, L. Li, and D. S. Ebert, "Manifold: A model-agnostic framework for interpretation and diagnosis of machine learning models," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 1, pp. 364–373, 2019.

[30] D. Ren, S. Amershi, B. Lee, J. Suh, and J. D. Williams, "Squares: Supporting interactive performance analysis for multiclass classifiers," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 61–70, 2017.

[31] F.-Y. Tzeng and K.-L. Ma, "Opening the black box - data driven visualization of neural networks," in *VIS 05. IEEE Visualization, 2005.*, 2005, pp. 383–390.

[32] Y. Ming, H. Qu, and E. Bertini, "Rulematrix: Visualizing and understanding classifiers with rules," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 1, pp. 342–352, 2019.

[33] H. A. Chipman, E. I. George, and R. E. McCulloch, "Bart: Bayesian additive regression trees," *Ann. Appl. Stat.*, vol. 4, no. 1, pp. 266–298, 03 2010.

[34] D. P. Green and H. L. Kern, "Modeling heterogeneous treatment effects in survey experiments with bayesian additive regression trees," *The Public Opinion Quarterly*, vol. 76, no. 3, pp. 491–511, 2012. [Online]. Available: http://www.jstor.org/stable/41684581

[35] R. A. Berk and J. Bleich, "Statistical procedures for forecasting criminal behavior," *Criminology & Public Policy*, vol. 12, no. 3, 2013.

[36] D. Collaris and J. J. van Wijk, "Machine learning interpretability through contribution-value plots," in *Proceedings of the 13th International Symposium on Visual Information Communication and Interaction*, ser. VINCI '20. New York, NY, USA: AACM, 2020.

[37] D. W. Apley and J. Zhu, "Visualizing the effects of predictor variables in black box supervised learning models," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 82, no. 4, pp. 1059–1086, 2020.

[38] P. Tamagnini, J. Krause, A. Dasgupta, and E. Bertini, "Interpreting black-box classifiers using instance-level visual explanations," in *Proceedings of the 2Nd Workshop on Human-In-the-Loop Data Analytics*, ser. HILDA'17. New York, NY, USA: ACM, 2017, pp. 6:1–6:6.

[39] M. Brehmer and T. Munzner, "A multi-level typology of abstract visualization tasks," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2376–2385, 2013.

[40] J. D. Mulder, J. J. van Wijk, and R. van Liere, "A survey of computational steering environments," *Future Generation Computer Systems*, vol. 15, no. 1, pp. 119–129, 1999. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X98000478

[41] R. van Liere, J. D. Mulder, and J. J. van Wijk, "Computational steering," *Future Generation Computer Systems*, vol. 12, no. 5, pp. 441–450, 1997, hPCN96. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X96000295

[42] H. A. Sturges, "The choice of a class interval," *Journal of the american statistical association*, vol. 21, no. 153, pp. 65–66, 1926.

[43] D. W. Scott, "On optimal and data-based histograms," *Biometrika*, vol. 66, no. 3, pp. 605–610, 1979.

[44] B. W. Silverman, *Density estimation for statistics and data analysis*. Routledge, 2018.

[45] C. J. Willmott and K. Matsuura, "Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance," *Climate Research*, vol. 30, no. 1, pp. 79–82, 2005. [Online]. Available: http://www.jstor.org/stable/24869236

[46] M. Sankaran, J. Ratnam, and N. Hanan, "Woody cover in african savannas: the role of resources, fire and herbivory," *Global Ecology and Biogeography*, vol. 17, no. 2, pp. 236–245, 2008.

[47] S. van den Elzen and J. J. van Wijk, "Small multiples, large singles: A new approach for visual data exploration," *Computer Graphics Forum*, vol. 32, no. 3pt2, pp. 191–200, 2013.

[48] J. Talbot, J. Gerth, and P. Hanrahan, "An empirical model of slope ratio comparisons," *IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis)*, 2012.

[49] R. A. Rossi and N. K. Ahmed, "The network data repository with interactive graph analytics and visualization," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, ser. AAAI'15. AAAI Press, 2015, pp. 4292–4293.

[50] H. Strobelt, S. Gehrmann, H. Pfister, and A. M. Rush, "Lstmvis: A tool for visual analysis of hidden state dynamics in recurrent neural networks," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 1, pp. 667–676, 2018.

[51] B. La Rosa, G. Blasilli, R. Bourqui, D. Auber, G. Santucci, R. Capobianco, E. Bertini, R. Giot, and M. Angelini, "State of the art of visual analytics for explainable deep learning," *Computer Graphics Forum*, vol. n/a, no. n/a, 2023.

[52] M. Van Someren, Y. F. Barnard, and J. Sandberg, "The think aloud method: a practical approach to modelling cognitive," *London: AcademicPress*, vol. 11, 1994.

[53] 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*, vol. 12, pp. 2825–2830, 2011.

[54] G. Richer, A. Pister, M. Abdelaal, J.-D. Fekete, M. Sedlmair, and D. Weiskopf, "Scalability in visualization," *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–15, 2022.

[55] M. Angelini, G. Santucci, H. Schumann, and H.-J. Schulz, "A review and characterization of progressive visual analytics," *Informatics*, vol. 5, no. 3, 2018. [Online]. Available: https://www.mdpi.com/2227-9709/5/3/31

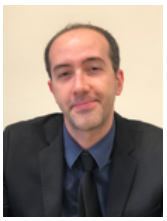
[56] J.-D. Fekete, D. Fisher, A. Nandi, and M. Sedlmair, *Progressive Data Analysis and Visualization*. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Apr. 2019. [Online]. Available: https://hal.inria.fr/hal-02090121

[57] M. Hogräfer, M. Angelini, G. Santucci, and H.-J. Schulz, "Steering-by-example for progressive visual analytics," *ACM Trans. Intell. Syst. Technol.*, vol. 13, no. 6, sep 2022. [Online]. Available: https://doi.org/10.1145/3531229

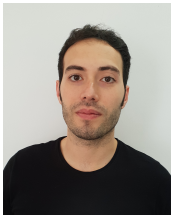
[58] R. Rao and S. K. Card, "The table lens: Merging graphical and symbolic representations in an interactive focus + context visualization for tabular information," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '94. New York, NY, USA: ACM, 1994, p. 318–322.

[59] D. Ceneda, T. Gschwandtner, T. May, S. Miksch, H.-J. Schulz, M. Streit, and C. Tominski, "Characterizing guidance in visual analytics," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 111–120, 2017.

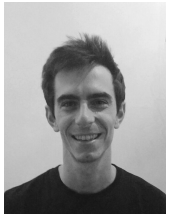
[60] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml



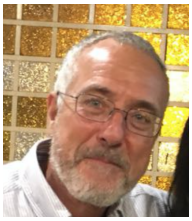
Marco Angelini, is an Assistant Professor in Engineering in Computer Science at Sapienza University of Rome, Italy. He is a member and coordinates research projects of *A.W.A.RE* group (http://aware.diag.uniroma1.it). His main research interests include Visual Analytics for predictive analysis, applied in the Cybersecurity domain, and Progressive Visual Analytics. He published 55+ papers in peer-reviewed international journals and conferences. More about him at https://sites.google.com/dis.uniroma1.it/angelini



Graziano Blasilli is a Post-Doctoral Researcher and a member of the *A.W.A.RE* research group, in Sapienza University of Rome, where he received the PhD in Engineering in Computer Science. His research interests are focused on Visual Analytics applied on the Cybersecurity domain and to the eXplainable Artificial Intelligence field. Find more at https://blasilli.com.



Simone Lenti is an Assistant Professor in Engineering in Computer Science and a member of the *A.W.A.RE* research group at Sapienza University of Rome. His research activities about Visual Analytics techniques for the Cybersecurity domain contributed to more than fifteen publications in peer-reviewed international journals and conferences and two EU-funded research projects.



Giuseppe Santucci is full professor in Engineering in Computer Science at Sapienza University of Rome. His main research activities concern visual languages, visualization, and visual analytics, focusing on both theoretical issues, like progressive visual analytics and visual quality metrics, and domain-specific applications, like information retrieval and cyber security. On such topics, he published 200+ papers in international journals and conferences.