PREDICTION VIA SHAPLEY VALUE REGRESSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Shapley values have several desirable properties for explaining black-box model predictions, which come with strong theoretical support. Traditionally, Shapley values are computed post-hoc, leading to additional computational cost at inference time. To overcome this, we introduce ViaSHAP, a novel approach that learns a function to compute Shapley values, from which the predictions can be derived directly by summation. We explore two learning approaches based on the universal approximation theorem and the Kolmogorov-Arnold representation theorem. Results from a large-scale empirical investigation are presented, in which the predictive performance of ViaSHAP is compared to state-of-the-art algorithms for tabular data, where the implementation using Kolmogorov-Arnold Networks showed a superior performance. It is also demonstrated that the explanations of ViaSHAP are accurate, and that the accuracy is controllable through the hyperparameters.

1 INTRODUCTION

The application of machine learning algorithms in some domains requires communicating the reasons behind predictions with the aim of building trust in the predictive models and, more importantly, addressing legal and ethical considerations (Lakkaraju et al., 2017; Goodman & Flaxman, 2017). Nevertheless, many state-of-the-art machine learning algorithms result in black-box models, precluding the user's ability to follow the reasoning behind the predictions. Consequently, explainable machine learning methods have gained notable attention as a means to acquire needed explainability without sacrificing performance.

Machine learning explanation methods employ a variety of strategies to produce explanations, e.g., the use of local interpretable surrogate models (Ribeiro et al., 2016), generation of counterfactual examples (Karimi et al., 2020; Dandl et al., 2020; Mothilal et al., 2020; Van Looveren & Klaise, 2021; Guo et al., 2021; Guyomard et al., 2022), selection of important features (Chen et al., 2018; Yoon et al., 2019; Jethani et al., 2021), and approximation of Shapley values (Lundberg & Lee, 2017; Lundberg et al., 2020; Frye et al., 2021; Covert & Lee, 2021; Jethani et al., 2022). Methods that generate explanations based on Shapley values are prominent since they offer a unique solution that meets a set of theoretically established, desirable properties. The computation of Shapley values can, however, be computationally expensive. Recent work has therefore focused on reducing the running time (Lundberg & Lee, 2017; Lundberg et al., 2020; Jethani et al., 2022) and enhancing the accuracy of approximations (Frye et al., 2021; Aas et al., 2021; Covert & Lee, 2021; Mitchell et al., 2022; Kolpaczki et al., 2024). However, the Shapley values are computed post-hoc, and hence entail a computational overhead, even when approximated, e.g., as in the case of FastSHAP. Generating instance-based explanations or learning a pre-trained explainer always demands further data, time, and resources. Nevertheless, to the best of our knowledge, computing Shapley values as a means to form the prediction has not yet been considered.

The main contributions of this study are:

- a novel machine learning method, ViaSHAP, that trains a model to simultaneously provide accurate predictions and Shapley values
- multiple implementations of the proposed method using the universal approximation theorem and the Kolmogorov-Arnold representation theorem, followed by a large-scale empirical investigation

In the following section, we cover fundamental concepts about the Shapley value and, along the way, introduce our notation. Section 3 describes the proposed method. In Section 4, results from a large-scale empirical investigation are presented and discussed. Section 5 provides a brief overview of the related work. Finally, in the concluding remarks, we summarize the main conclusions and outline directions for future work.

2 Preliminaries

2.1 The Shapley Value

In game theory, a game in coalitional form is a formal model for a scenario in which players form coalitions, and the game's payoff is shared between the coalition members. A coalitional game focuses on the behavior of the players and typically involves a finite set of players $N = \{1, 2, \ldots, n\}$ (Manea, 2016). A coalitional game also involves a characteristic set function $v : 2^N \to \mathbb{R}$ that assigns a payoff, a real number, to a coalition $S \subseteq N$ such that: $v(\emptyset) = 0$ (Owen, 1995.). Different concepts can be employed to distribute the payoff among the players of a coalitional game to achieve a fair and stable allocation. Such solution concepts include the Core, the Nucleolus, and the Shapley Value (Manea, 2016; Ferguson, 2018).

The Shapley Value is a solution concept that allocates payoffs to the players according to their marginal contributions across possible coalitions. The Shapley value $\phi_i(v)$ of player i in game v is given by (Shapley, 1953):

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(n - |S| - 1)!}{n!} (v(S \cup \{i\}) - v(S)).$$

The term $\left(\frac{|S|!(n-|S|-1)!}{n!}\right)$ is a combinatorial weighting factor for the different coalitions that can be formed for game v. The difference term $(v(S \cup \{i\}) - v(S))$ represents the additional value that player i contributes to the coalition S, i.e., the marginal contribution of player i.

Given a game v, an additive explanation model μ is an interpretable approximation of v which can be written as (Lundberg & Lee, 2017; Covert & Lee, 2021):

$$\mu(S) = \delta_0(v) + \sum_{i \in S} \delta_i(v) \qquad \text{with } \delta_0(v) \text{ a constant and } \delta_i(v) \text{ the payoff of player } i.$$

μ is a linear model whose weights are the payoffs of each player. Using the Shapley values as the payoffs is the only solution in the class of additive feature attribution methods that satisfies the following properties (Young, 1985):

- **Property 1** ***(Local Accuracy)****: the solution matches the prediction of the underlying model:*

$$\mu(N) = \sum_{i \in N} \phi_i(v) = v(N)$$

- **Property 2** ***(Missingness)****: Players without impact on the prediction attributed a value of zero. Let $i \in N$:*

$$\forall S \subseteq N \setminus \{i\},\ v(S) = v(S \cup \{i\}) \Rightarrow \phi_i(v) = 0$$

- **Property 3** ***(Consistency)****: The Shapley value grows or remains the same if a player's contribution grows or stays the same. Let v and v' two games over N, let $i \in N$:*

$$\forall S \subseteq N \setminus \{i\}, v(S \cup \{i\}) - v(S) \geq v'(S \cup \{i\}) - v'(S)$$
$$\Rightarrow \phi_i(v) \geq \phi_i(v')$$

2.2 SHAP

In the context of explainable machine learning, the Shapley value is commonly computed post-hoc to explain the predictions of trained machine learning models. Let f be a trained model whose inputs are defined on n features and whose output $y \in Y \subseteq \mathbb{R}$. We also define a *baseline* or *neutral* instance, noted $\mathbf{0} \in X$. For a given instance $\mathbf{x}$, the Shapley value is computed over each feature to explain the difference in output $\mathbf{x} \in X$ and the baseline. The baseline may be determined depending on the context, but common examples include the average of all examples in the training set, or one that is commonly used as a threshold (Izzo et al., 2021).

In this context, a coalitional game for S can be derived from the model, where the players are the features, and the payoff is the difference in output wrt the baseline:

$$\forall S \subseteq N,\ v_{\mathbf{x}}(S) = f(\mathbf{x}^S) - f(\mathbf{0}),$$

where $x_i^S = x_i$ if $i \in S$, $x_i^S = \mathbf{0}_i$ otherwise. In this game, a player i getting picked for coalition S means that its corresponding feature's value is x_i, otherwise it remains at its baseline value $\mathbf{0}_i$. Note that $v_{\mathbf{x}}(\emptyset) = f(\mathbf{0}) - f(\mathbf{0}) = 0$, which makes $v_{\mathbf{x}}$ a valid coalition game.

The Shapley values for this game can then be obtained as the solution of an optimization problem. The objective is to determine a set of values that accurately represent the marginal contributions of each feature while verifying properties 1 through 3. In the litterature, they were obtained by minimizing the following weighted least squares loss function (Marichal & Mathonet, 2011; Lundberg & Lee, 2017; Patel et al., 2021):

$$\mathcal{L}(v_{\mathbf{x}}, \mu_{\mathbf{x}}) = \sum_{S \subseteq N} \omega(S)\Big(v_{\mathbf{x}}(S) - \mu_{\mathbf{x}}(S)\Big)^2, \tag{1}$$

where ω is a weighting kernel, the choice of the kernel can result in a solution equivalent to the Shapley value (Covert & Lee, 2021; Covert et al., 2020). Therefore, Lundberg & Lee (2017) proposed the Shapley kernel:

$$\omega_{Shap}(S) = \frac{(n-1)}{\binom{n}{|S|} \cdot |S| \cdot (n - |S|)}. \tag{2}$$

Note that, for a d-dimensional output with $d > 1$, each output is considered as a different unidimensional model. That is, each of the d dimensions will define a different game, and thus a different set of n Shapley values. The explanation of the output is thus an $n \times d$ matrix of Shapley values, providing the contribution of each input feature to each output game. This can trivially be obtained through the same optimization process by stacking d loss functions such as in equation 1. Thus, we will consider in the following that y be unidimensional unless otherwise specified.

2.3 KernelSHAP

Computing the exact Shapley values is a demanding process as it requires evaluating all possible coalitions of feature values. There are $2^n - 1$ possible coalitions for a model with n features, each of which has to be evaluated to determine the features' marginal contributions, which renders the exact computation of Shapley values infeasible for models with a relatively large number of features. Consequently, Lundberg & Lee (2017) proposed KernelSHAP as a more feasible method to approximate the Shapley values. KernelSHAP samples a subset of coalitions instead of evaluating all possible coalitions. The explanation model is learned by solving the following optimization problem (Covert & Lee, 2021; Jethani et al., 2022):

$$\begin{aligned}\phi(v_{\mathbf{x}}) &= \underset{\phi_{\mathbf{x}} \in \mathbb{R}^n}{\arg\min} \underset{p(S)}{\mathbb{E}}\Big[\Big(v_{\mathbf{x}}(S) - \mathbf{1}_S^\top \phi_{\mathbf{x}}\Big)^2\Big] \\ &= \underset{\phi_{\mathbf{x}} \in \mathbb{R}^n}{\arg\min} \underset{p(S)}{\mathbb{E}}\Big[\Big(f(\mathbf{x}^S) - f(\mathbf{0}) - \mathbf{1}_S^\top \phi_{\mathbf{x}}\Big)^2\Big]\end{aligned} \tag{3}$$

$$\text{s.t. } \mathbf{1}^\top \phi_{\mathbf{x}} = v_{\mathbf{x}}(N) = f(\mathbf{x}) - f(\mathbf{0}), \tag{4}$$

where $\mathbf{1}_S$ is the mask corresponding to S, i.e. which takes value 1 for features in S and 0 otherwise, and the distribution $p(S)$ is proportional to the Shapley kernel (equation 2) (Covert & Lee, 2021; Jethani et al., 2022). equation 4 is referred to as the *efficiency* constraint.

2.4 FastSHAP

Although KernelSHAP provides a practical solution for the Shapley value estimation, the optimization problem 3 must be solved separately for every prediction. Additionally, KernelSHAP requires many samples to converge to accurate estimations for the Shapley values, and this problem is exacerbated with high dimensional data (Covert & Lee, 2021). Consequently, FastSHAP (Jethani et al., 2022) has been proposed to efficiently learn a parametric Shapley value function and eliminate the need to solve a separate optimization problem for each prediction. The model $\phi_{\text{fast}} : X \to \mathbb{R}^n$, parameterized by θ is then trained to produce the Shapley value for an input by minimizing the following loss function:

$$\begin{aligned} \mathcal{L}(\theta) &= \mathop{\mathbb{E}}_{p(\mathbf{x})} \mathop{\mathbb{E}}_{p(S)} \left[\left(v_{\mathbf{x}}(S) - \mathbf{1}_S^\top \phi_{\text{fast}}(\mathbf{x}; \theta) \right)^2 \right] \\ &= \mathop{\mathbb{E}}_{p(\mathbf{x})} \mathop{\mathbb{E}}_{p(S)} \left[\left(f(\mathbf{x}^S) - f(\mathbf{0}) - \mathbf{1}_S^\top \phi_{\text{fast}}(\mathbf{x}; \theta) \right)^2 \right], \end{aligned} \tag{5}$$

where $p(\mathbf{x})$ is the distribution of the input data, and $p(S)$ is proportional to the Shapley kernel defined in equation 2. In the case of a multidimensional output, a uniform sampling is done over the possible output dimensions.

The accuracy of ϕ_{fast} in approximating the Shapley value depends on the expressiveness of the model class employed as well as the data available for learning ϕ_{fast} as a post-hoc function.

3 ViaSHAP

We introduce ViaSHAP, a method that formulates predictions via Shapley values regression. In contrast to the previous approaches, the Shapley values are not computed in a post-hoc setup. Instead, the learning of Shapley values is integrated into the training of the predictive model and exploits every data example in the training data. At inference time, the Shapley values are used directly to generate the prediction. The following subsections describe how ViaSHAP is trained to predict simultaneously both accurate predictions and the corresponding explanation through Shapley values.

3.1 Predicting Shapley Values

Let $X \subseteq \mathbb{R}^n$ and $Y \subseteq \mathbb{R}^d$ respectively the input and output spaces, and $M = \{1, \cdots, d\}$ the set of output dimensions. We define a model $\mathcal{V}ia^{SHAP} : X \to Y$ which, for a given instance $\mathbf{x}$, computes both the Shapley values and the predicted output in a single process.

First, $\phi^{\mathcal{V}ia} : X \to \mathbb{R}^{n \times d}$ computes a matrix of values $\phi^{\mathcal{V}ia}(\mathbf{x}; \theta)$. Then, $\mathcal{V}ia^{SHAP}$ predicts the output vector as $\mathcal{V}ia^{SHAP}(\mathbf{x}) = \mathbf{1}^\top \phi^{\mathcal{V}ia}(\mathbf{x}; \theta)$ i.e., summing column-wise. A link function σ can be applied to accommodate a valid range of outputs $\big(\mathbf{y} = \sigma(\mathbf{1}^\top \phi^{\mathcal{V}ia}(\mathbf{x}; \theta)\big)$, e.g., the sigmoid function for binary classification or softmax for multi-class classification.

$\mathcal{V}ia^{SHAP}$ computes the Shapley values of each prediction and uses the predicted Shapley values to formulate the outcome, as illustrated in Figure 1. Similar to KernelSHAP and FastSHAP (in equation (3) and equation (5)), $\phi^{\mathcal{V}ia}$ is trained by minimizing the weighted least squares loss of the predicted Shapley values as follows:

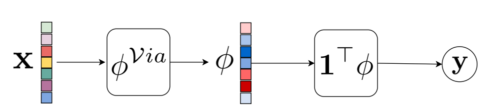

Figure 1: $\mathcal{V}ia^{SHAP}$ generates predictions by first estimating the Shapley values, whose summation produces the final outcome.

$$\mathcal{L}_{\phi}(\theta) = \sum_{\mathbf{x}\in X} \sum_{j\in M} \mathop{\mathbb{E}}_{p(S)} \Big[\Big(\mathcal{V}ia_j^{\textit{SHAP}}(\mathbf{x}^S) - \mathcal{V}ia_j^{\textit{SHAP}}(\mathbf{0}) - \mathbf{1}_S^\top \phi_j^{\mathcal{V}ia}(\mathbf{x};\theta) \Big)^2 \Big]. \tag{6}$$

Given that the ground truth Shapley values are inaccessible during training, the learning process relies solely on sampling input features, based on the principle that unselected features should be assigned a Shapley value of zero, while the prediction formulated using the selected features should be equal to the sum of their corresponding Shapley values. Since $\phi^{\mathcal{V}ia}$ and $\mathcal{V}ia^{\textit{SHAP}}$ are essentially the same model, coalition sampling for both functions is performed within the same model but at different locations. For $\mathcal{V}ia^{\textit{SHAP}}(S)$, the sampling occurs on the input features before feeding them to the model. While $\mathbf{1}_S^\top \phi^{\mathcal{V}ia}$ sampling is applied to the predicted Shapley values, given the original set of features as input to the model, as illustrated in Figure 2. In the following, we show that the solution computed by the optimized $\phi^{\mathcal{V}ia}(\mathbf{x};\theta^*)$ function maintains the desirable properties of Shapley values for each output dimension. For ease of notation, we drop the subscript j below and consider one output at a time. All proofs, unless otherwise specified, can be found in the Appendix.

Lemma 1 *$\phi^{\mathcal{V}ia}(\mathbf{x};\theta)$ satisfies the property of local accuracy wrt $\mathcal{V}ia^{\textit{SHAP}}$.*

Lemma 2 *The global minimizer model, $\phi^{\mathcal{V}ia}(\mathbf{x};\theta^*)$, of the loss function (6), assigns value zero to features that have no influence on the outcome predicted by $\mathcal{V}ia^{\textit{SHAP}}(\mathbf{x})$ in the distribution $p(S)$.*

Lemma 3 *Let two $\mathcal{V}ia^{\textit{SHAP}}$ models $\mathcal{V}$ and $\mathcal{V}'$ whose respective $\phi^{\mathcal{V}ia}$ are parameterized by θ^* and $\theta^{*'}$, which globally optimize loss function (6) over two possibly different targets y and y'. Then, given a feature $i \in N$:*

$$\forall S \subseteq N \setminus \{i\}, \mathcal{V}(\mathbf{x}^{S\cup\{i\}}) - \mathcal{V}(\mathbf{x}^S) \geq \mathcal{V}'(\mathbf{x}^{S\cup\{i\}}) - \mathcal{V}'(\mathbf{x}^S) \Rightarrow \phi_i^{\mathcal{V}ia}(\mathbf{x};\theta^*) \geq \phi_i^{\mathcal{V}ia}(\mathbf{x};\theta^{*'})$$

Theorem 1 *The global optimizer function $\phi^{\mathcal{V}ia}(\mathbf{x};\theta^*)$ computes the exact Shapley values of the predictions of $\mathcal{V}ia^{\textit{SHAP}}(\mathbf{x})$.*

Theorem 1 directly follows from Lemma 1, Lemma 2, and Lemma 3, which demonstrate that $\phi^{\mathcal{V}ia}(\mathbf{x};\theta^*)$ adheres to properties 1 through 3, as well as the fact that Shapley values provide the sole solution for assigning credit to players while satisfying the properties from 1 to 3 (Young, 1985; Lundberg & Lee, 2017).

3.2 Predictor Optimization

The parameters of $\mathcal{V}ia^{\textit{SHAP}}$ are optimized with the following dual objective: to learn an optimal function for producing the Shapley values of the predictions and to minimize the prediction loss with respect to the true target. Therefore, the prediction loss is minimized using a function suitable for the specific prediction task, e.g., binary cross-entropy for binary classification or mean squared error for regression tasks. The following presents the loss function for multinomial classification:

$$\mathcal{L}(\theta) = \sum_{\mathbf{x}\in X} \sum_{j\in M} \Big(\beta \cdot \Big(\mathop{\mathbb{E}}_{p(S)} \Big[\Big(\mathcal{V}ia_j^{\textit{SHAP}}(\mathbf{x}^S) - \mathcal{V}ia_j^{\textit{SHAP}}(\mathbf{0}) - \mathbf{1}_S^\top \phi_j^{\mathcal{V}ia}(\mathbf{x};\theta) \Big)^2 \Big] \Big) - \Big(j \log(\hat{j}) \Big) \Big), \tag{7}$$

where β is a predefined scaling hyperparameter and $\hat{j}$ is the predicted probability of class $j \in M$ by $\mathcal{V}ia^{\textit{SHAP}}$. The optimization of $\mathcal{V}ia^{\textit{SHAP}}$ is illustrated in Figure 2 and summarized in Algorithm 1.

3.3 ViaSHAP Approximator

According to the universal approximation theorem, a feedforward network with at least one hidden layer and sufficient units in the hidden layer can approximate any continuous function over a compact input set to an arbitrary degree of accuracy, given a suitable activation function (Hornik et al., 1989; Cybenko, 1989; Hornik, 1991). Consequently, neural networks and multi-layer perceptrons (MLP) can be employed to learn $\mathcal{V}ia^{\textit{SHAP}}$ for prediction tasks where there is a continuous mapping function from the input dataset to the true targets, which also applies to the true Shapley values as a continuous function.

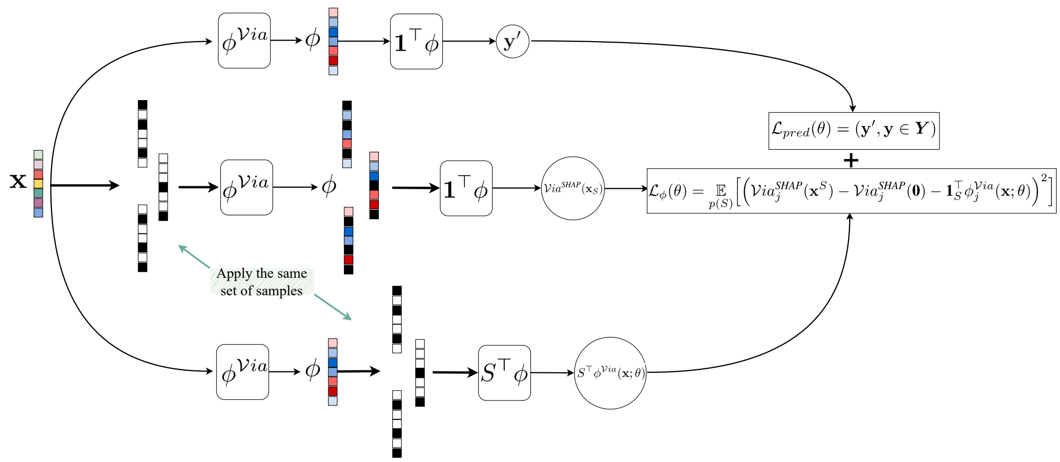

Figure 2: **The optimization of $\mathcal{V}ia^{SHAP}$** is conducted using a dual-objective loss function that aims to learn an optimal function for generating the Shapley values while minimizing the prediction loss.

Liu et al. (2024) recently proposed Kolmogorov–Arnold Networks (KAN), as an alternative approach to MLPs inspired by the Kolmogorov-Arnold representation theorem. According to the Kolmogorov-Arnold representation theorem, a multivariate continuous function on a bounded domain can represented by a finite sum of compositions of continuous univariate functions (Kolmogorov, 1956; 1957; Liu et al., 2024), as follows:

$$f(\mathbf{x}) = f(x_1, \ldots, x_n) = \sum_{q=1}^{2n+1} \Psi_q\Big(\sum_{p=1}^{n} \psi_{q,p}(x_p)\Big),$$

where $\psi_{q,p} : [0,1] \to \mathbb{R}$ is a univariate function and $\Psi_q : \mathbb{R} \to \mathbb{R}$ is a univariate continuous function. Liu et al. (2024) defined a KAN layer as a matrix of one-dimensional functions: $\Psi = \{\psi_{q,p}\}$, with $p = 1, 2, \ldots, n_{\text{in}}$ and $q = 1, 2, \ldots, n_{\text{out}}$. Where n_{in} and n_{out} represent the dimensions of the layer's input and output, respectively, and $\psi_{q,p}$ are learnable functions parameterized as splines. A KAN network is a composition of L layers stacked together; subsequently, the output of KAN on instance $\mathbf{x}$ is given by:

$$\mathbf{y} = \text{KAN}(\mathbf{x}) = \Psi_{L-1} \circ \Psi_{L-2} \circ \cdots \circ \Psi_1 \circ \Psi_0(\mathbf{x}).$$

The degree of each spline and the number of splines for each function are both hyperparameters.

4 Empirical Investigation

We evaluate both the predictive performance of $\mathcal{V}ia^{SHAP}$ and the feature importance attribution with respect to the true Shapley value. This section begins with outlining the experimental setup. Then, the predictive performance of $\mathcal{V}ia^{SHAP}$ is evaluated. Afterwards, we benchmark the similarity between the feature importance obtained by $\mathcal{V}ia^{SHAP}$ and the ground truth Shapley values. We also evaluate the predictive performance and the accuracy of Shapley values on image data. Finally, we summarize the findings of the ablation study.

4.1 Experimental Setup

We employ 25 publicly available datasets in the experiments, each divided into training, validation, and test subsets [1]. The training set is used to train the model, the validation set is used to detect overfitting and determine early stopping, and the test set is used to evaluate the model's performance. All the learning algorithms are trained using default settings without hyperparameter tuning. The training and validation sets are combined into a single training set for algorithms that do not utilize

[1]The details of the datasets are available in Table 13

a validation set for performance tracking. During data preprocessing, categorical feature categories are tokenized with numbers starting from one, reserving zero for missing values. We use standard normalization so the average value over each feature is $\mathbf{0}$ and masking the value of feature i with $\mathbf{0}$ is equivalent to setting its value to $\mathbb{E}(x_i)$. We follow the interventional approach to approximate Shapley values, as it is computationally more efficient and tend to be more "true" to the data, as suggested by Chen et al. (2020). We experimented with four different implementations of $\mathcal{V}ia^{SHAP}$, using Kolmogorov–Arnold Networks (KANs) and feedforward neural networks:

1- $\textit{KAN}^{\mathcal{V}ia}$: Based on the method proposed by Liu et al. (2024) using a computationally efficient implementation[2]. Uses spline basis functions and consists of an input layer, two hidden layers, and an output layer. Layer dimensions: Input layer maps n features to 64 dimensions, the first hidden layer to 128 dimensions, the second hidden layer to 64 dimensions, and the output layer to $n\times$ (number of classes).

2- $\textit{KAN}_{\varrho}^{\mathcal{V}ia}$: Replaces the spline basis in the original KANs with Radial Basis Functions (RBFs)[3]. The architecture matches that of $\textit{KAN}^{\mathcal{V}ia}$.

3- $\textit{MLP}^{\mathcal{V}ia}$: A multi-layer perceptron (MLP) with identical input and output dimensions as the KAN-based implementations. Incorporates batch normalization after each layer and uses ReLU activation functions.

Algorithm 1: $\mathcal{V}ia^{SHAP}$

Data: training data X, labels Y, scalar β
Result: model parameters θ
Initialize $\mathcal{V} : \mathcal{V}ia^{SHAP}(\phi^{\mathcal{V}ia}(\mathbf{x};\theta))$
while *not converged* **do**
  $\mathcal{L} \leftarrow 0$
  for *each* $\mathbf{x} \in X$ *and* $\mathbf{y} \in Y$ **do**
    sample $S \sim p(S)$
    $\mathbf{y}' \leftarrow \mathcal{V}(\mathbf{x})$
    $\mathcal{L}_{pred} \leftarrow$ *prediction loss*$(\mathbf{y}', \mathbf{y})$
    $\mathcal{L}_{\phi} \leftarrow \Big(\mathcal{V}_{\mathbf{y}}(\mathbf{x}^S) - \mathcal{V}_{\mathbf{y}}(\mathbf{0}) - \mathbf{1}_S^\top \phi_{\mathbf{y}}^{\mathcal{V}ia}(\mathbf{x};\theta)\Big)^2$
    $\mathcal{L} \xleftarrow{+} \mathcal{L}_{pred} + \beta \cdot \mathcal{L}_{\phi}$
  end
  Compute gradients $\nabla_\theta \mathcal{L}$
  Update $\theta \leftarrow \theta - \nabla_\theta \mathcal{L}$
end

4- $\textit{MLP}_{\theta}^{\mathcal{V}ia}$: Similar to $\textit{MLP}^{\mathcal{V}ia}$ but increases the number of units in the hidden layers to match the total number of parameters in the $\textit{KAN}^{\mathcal{V}ia}$ models since $\textit{KAN}^{\mathcal{V}ia}$ always results in models with more parameters than the remaining implementations. Hidden layer dimensions are adjusted based on the dataset.

The four implementations were trained with the β of equation 7 set to 10 and used 32 sampled coalitions per instance. The above hyperparameters were determined in a quasirandom manner.

For the evaluation of the predictive performance, the four ViaSHAP approximators ($\textit{KAN}^{\mathcal{V}ia}$, $\textit{KAN}_{\varrho}^{\mathcal{V}ia}$, $\textit{MLP}^{\mathcal{V}ia}$, and $\textit{MLP}_{\theta}^{\mathcal{V}ia}$) are compared against XGBoost, Random Forests, and TabNet (Arik & Pfister, 2021). All the compared algorithms are trained using the default hyperparameters settings without tuning, as it has been shown by Shwartz-Ziv & Armon (2022) that deep models typically require more extensive tuning on each tabular dataset to match the performance of tree ensemble models, e.g., XGBoost. If the model's performance varies with different random seeds, it will be trained using five different seeds, and the average result will be reported alongside the standard deviation. In binary classification tasks with imbalanced training data, the minority class in the training subset is randomly oversampled to match the size of the majority class, a common strategy to address highly imbalanced data (Koziarski et al., 2017; ao Huang et al., 2022). On the other hand, no oversampling is applied to multinomial classification datasets. The area under the ROC curve (AUC) is used for measuring predictive performance since it is invariant to classification thresholds. For multinomial classification, we compute the AUC for each class versus the rest and then weighting it by the class support. If two algorithms achieve the same AUC score, the model with a smaller standard deviation across five repetitions with different random seeds is considered better. For the explainability evaluation, we generate ground truth Shapley values by running KernelSHAP until it converges since it has been demonstrated that KernelSHAP will converge to the true Shapley value when given a sufficiently large number of data samples (Covert & Lee, 2021).[4] We measure the similarity of the approximated Shapley values by ViaSHAP to the ground truth using cosine similarity and Spearman rank (Spearman, 1904) correlation, where cosine similarity measures the alignment between two explanation vectors, while Spearman rank correlation measures the consis-

[2] https://github.com/Blealtan/efficient-kan
[3] https://github.com/ZiyaoLi/fast-kan
[4] https://github.com/iancovert/shapley-regression

tency in feature rankings. The results are presented as mean values with standard deviations across all data instances in the test set.

For image experiments, we use the CIFAR-10 dataset (Krizhevsky et al., 2014). We provide three $\mathcal{V}ia^{SHAP}$ implementations for image classification: $ResNet50^{\mathcal{V}ia}$, $ResNet18^{\mathcal{V}ia}$, and $U\text{-}Net^{\mathcal{V}ia}$ based on ResNet50, ResNet18 (He et al., 2016), and U-Net (Ronneberger et al., 2015), respectively. The accuracy of the Shapley values is estimated by measuring the effect of excluding and including the top important features on the prediction, similar to the approach followed by Jethani et al. (2022).

4.2 Predictive Performance Evaluation

We evaluated the performance of the seven algorithms ($KAN^{\mathcal{V}ia}$, $KAN_{\varrho}^{\mathcal{V}ia}$, $MLP^{\mathcal{V}ia}$, $MLP_{\theta}^{\mathcal{V}ia}$, TabNet, Random Forests, and XGBoost) across the 25 datasets, with detailed results presented in Table 1. The results show that $KAN^{\mathcal{V}ia}$ obtains the highest average rank with respect to AUC. $KAN_{\varrho}^{\mathcal{V}ia}$ came in second place, closely followed by XGBoost, with only a slight difference between them. We employed the Friedman test (Friedman, 1939) to determine whether the observed performance differences are statistically significant. We tested the null hypothesis that there is no difference in predictive performance. The Friedman test allowed the rejection of the null hypothesis, indicating that there is indeed a difference in predictive performance, as measured by AUC, at the 0.05 significance level. Subsequently, the post-hoc Nemenyi (Nemenyi, 1963) test was applied to identify which pairwise differences are significant, again at the 0.05 significance level. The results of the post-hoc test, summarized in Figure 3, indicate that the differences between $\mathcal{V}ia^{SHAP}$ using KAN implementations and the tree ensemble models, i.e., XGBoost and Random Forests, are statistically insignificant, given the sample size of 25 datasets. However, the differences in predictive performance between $KAN^{\mathcal{V}ia}$ and MLP variants ($MLP^{\mathcal{V}ia}$ and $MLP_{\theta}^{\mathcal{V}ia}$) are statistically significant. It is also noticeable that the MLP variants of $\mathcal{V}ia^{SHAP}$ underperform compared to all other competitors, even when the MLP models have an equivalent number of parameters to $KAN^{\mathcal{V}ia}$. We also evaluated the impact of incorporating Shapley loss on the predictive performance of a KAN model by comparing $KAN^{\mathcal{V}ia}$ to an identical KAN classifier trained without the Shapley loss. The results show that $KAN^{\mathcal{V}ia}$ significantly outperforms identical KAN architecture that is not optimized to compute Shapley values. The detailed results are available in Appendix G.

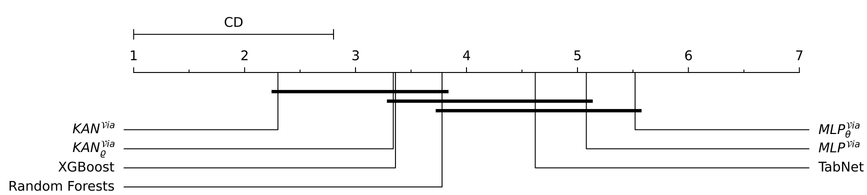

Figure 3: **The average rank** of the 7 predictors on the 25 datasets with respect to the AUC (the lower rank is better). The critical difference (CD) is the largest statistically insignificant difference.

4.3 Explainability Evaluation

The explainability of the various ViaSHAP implementations is evaluated by measuring the similarity of $\mathcal{V}ia^{SHAP}$'s Shapley values ($\phi^{\mathcal{V}ia}(\mathbf{x};\theta)$ to the ground truth Shapley values (ϕ), computed by the unbiased KernelSHAP, as discussed in Subsection 4.1, taking $\mathcal{V}ia^{SHAP}$ as the black-box model. We present results for models trained with the default values for the hyperparameters. The effect of these settings are further investigated in the ablation study.

The evaluation of the alignment between $\phi^{\mathcal{V}ia}(\mathbf{x};\theta)$ and ϕ using cosine similarity generally shows a high degree of similarity between the generated Shapley values and the ground truth as illustrated in Figure 4. The ranking of the compared variants of $\mathcal{V}ia^{SHAP}$ with respect to their cosine similarity to the ground truth Shapley values shows that $MLP_{\theta}^{\mathcal{V}ia}$ is ranked first, followed by $KAN^{\mathcal{V}ia}$, $KAN_{\varrho}^{\mathcal{V}ia}$, and $MLP^{\mathcal{V}ia}$, respectively. However, the Friedman test does not indicate any significant difference between the different approaches to compute Shapley values. At the same time, the results of ranking the four variants of $\mathcal{V}ia^{SHAP}$ based on their Spearman rank correlation with the ground truth Shapley

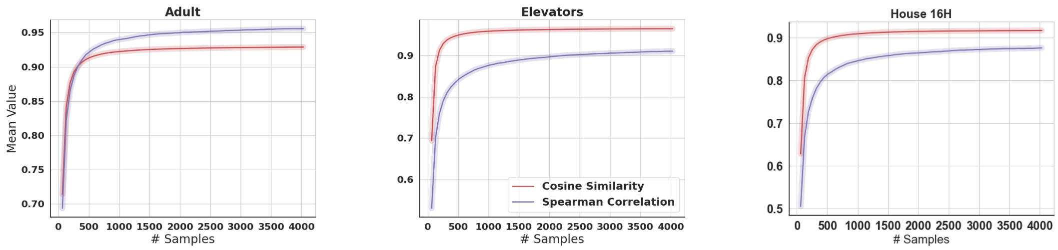

Figure 4: **The similarity between $KAN^{\mathcal{V}ia}$ and KernelSHAP's approximations.** KernelSHAP initially provides approximations that differ remarkably from the values of $\mathcal{V}ia^{SHAP}$. However, as KernelSHAP refines its approximations with more samples, the similarity to $\mathcal{V}ia^{SHAP}$'s values grows.

values reveal that $KAN^{\mathcal{V}ia}$ ranks first, followed by a tie for second place between $KAN_{\varrho}^{\mathcal{V}ia}$ and $MLP_{\theta}^{\mathcal{V}ia}$, and $MLP^{\mathcal{V}ia}$ placing last. In order to find out whether the differences are significant, the Friedman test is applied once again, which allows for the rejection of the null hypothesis, indicating that there is indeed a difference between the compared models in their $\phi^{\mathcal{V}ia}(\mathbf{x};\theta)$ correlations to the ground truth ϕ, at 0.05 significance level. The post-hoc Nemenyi test, at 0.05 level, indicates that differences between $MLP^{\mathcal{V}ia}$ and the remaining models are significant, as summarized in Figure 6. Overall, $KAN^{\mathcal{V}ia}$ is found to be a relatively stable approximator across the 25 datasets when both similarity metrics (cosine similarity and Spearman rank correlation) are considered. Detailed results can be found in Tables 2 and 3 in Appendix E. We also compare the accuracy of the Shapley values generated by $\mathcal{V}ia^{SHAP}$ to those produced by FastSHAP, with $\mathcal{V}ia^{SHAP}$ models utilized as black-boxes within FastSHAP. The results in Appendix I show that $\mathcal{V}ia^{SHAP}$ significantly outperforms FastSHAP in terms of similarity to the ground truth.

4.4 Image Experiments

We evaluated the predictive performance of $ResNet50^{\mathcal{V}ia}$, $ResNet18^{\mathcal{V}ia}$, and $U\text{-}Net^{\mathcal{V}ia}$ on the CIFAR-10 dataset. All models were trained from scratch (without transfer learning). The results, summarized in Table 4, demonstrate that $\mathcal{V}ia^{SHAP}$ can perform accurately in image classification tasks. We also compared the accuracy of the explanations obtained by $\mathcal{V}ia^{SHAP}$ implementations with those obtained by FastSHAP (where $\mathcal{V}ia^{SHAP}$ models were treated as black boxes). The results in Table 5 and Figure 7 show that $\mathcal{V}ia^{SHAP}$ consistently provides more accurate Shapley value approximations than the explanations obtained using FastSHAP. The experiment details can be found in Appendix F.

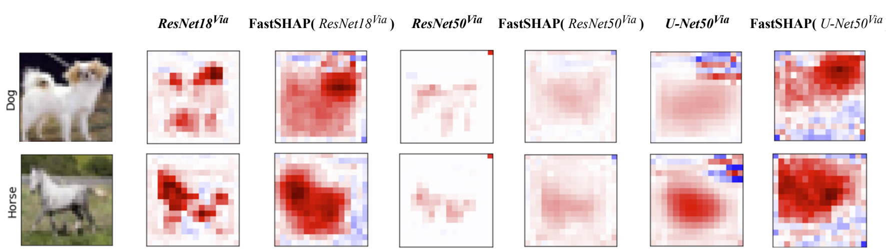

Figure 5: The explanation of the predicted class using two random images from the CIFAR-10.

4.5 Ablation Study

The ablation study was conducted after the empirical evaluation to ensure that no prior knowledge of the data or models influenced the experimental setup. In the ablation study, we assessed the impact of the scaling hyperparameter β and the number of sampled coalitions. The detailed results of the ablation study are provided in Appendix H. We began by examining the effect of β on both predictive performance and the similarity of computed Shapley values to the ground truth. The results demonstrate that predictive performance remains robust to changes in β, unless β is raised to an exceptionally large value, e.g., $\geq$200-fold. A more remarkable observation is that the similarity of the computed Shapley values to the ground truth improves as β grows. However, the model fails to learn properly with substantially large β. Afterwards, we evaluated the effect of the number of sampled coalitions per data instance on the performance of the learned models. The results suggest that the number of samples has little impact on both predictive performance and the similarity of the computed Shapley values to the ground truth compared to beta, i.e., $\mathcal{V}ia^{SHAP}$ can be effectively trained with as few as one sample per data instance. We also study the effect of a link function on both predictive performance and the accuracy of Shapley values of $\mathcal{V}ia^{SHAP}$. Finally, we examined the impact of β on the progression of training and validation loss during the training phase. The results indicate that $\mathcal{V}ia^{SHAP}$ tends to require a longer time to converge as β values increase.

5 Related Work

In addition to KernelSHAP and the real-time method FastSHAP, alternative approaches have been proposed to reduce the time required for Shapley value approximation. Methods that exploit specific properties of the explained model can provide faster computations, e.g., TreeSHAP (Lundberg et al., 2020) and DASP (Ancona et al., 2019), while others limit the scope to specific problems, e.g., image classifications or text classification (Chen et al., 2019; Teneggi et al., 2022). Additionally, directions to improve Shapley value approximation by enhancing data sampling have also been explored (Frye et al., 2021; Aas et al., 2021; Covert et al., 2021; Mitchell et al., 2022; Chen et al., 2023; Kolpaczki et al., 2024). Nevertheless, traditional methods for computing Shapley values have typically been considered post-hoc solutions for explaining predictions, requiring additional time, data, and resources to generate explanations. In contrast, $\mathcal{V}ia^{SHAP}$ computes Shapley values during inference, eliminating the need for a separate post-hoc explainer.

Research on generating explanations using pre-trained models has explored several approaches. Chen et al. (2018), Yoon et al. (2019), and Jethani et al. (2021) trained models for important features selection. Schwab & Karlen (2019) trained a model to estimate the influence of different inputs on the predicted outcome. Situ et al. (2021) proposed to distill any explanation algorithm for text classification. Pretrained explainers, similar to other post-hoc methods, require further resources for training, and the fidelity of their explanations to the underlying black-box model can vary.

Many approaches for creating explainable neural networks have been proposed in the literature. Such approaches not only generate predictions but also include an integrated component that provides explanations, which is trained alongside the predictor (Lei et al., 2016; Alvarez Melis & Jaakkola, 2018; Guo et al., 2021; Al-Shedivat et al., 2022; Sawada & Nakamura, 2022; Guyomard et al., 2022). Explainable graph neural networks (GNNs) have also been studied for graph-structured data, which typically exploit the internal properties of their models to generate explanations, e.g., the similarity between nodes (Dai & Wang, 2021), finding patterns and common graph structures(Feng et al., 2022; Zhang et al., 2022; Cui et al., 2022), or analyzing the behavior of different components of the GNN (Xuanyuan et al., 2023). Explanations generated by explainable neural networks do not correspond to Shapley values or meet the properties inherent to Shapley values, in contrast to $\mathcal{V}ia^{SHAP}$. Moreover, the explanations are offered without fidelity guarantees and do not elaborate on how exactly the predictions are computed, whereas $\mathcal{V}ia^{SHAP}$ generates predictions directly from their Shapley values.

6 Concluding Remarks

We have proposed ViaSHAP, an algorithm that computes Shapley values during inference. We evaluated the performance of ViaSHAP using implementations based on the universal approximation theorem and the Kolmogorov-Arnold representation theorem. We have presented results from a large-scale empirical investigation, in which ViaSHAP was evaluated with respect to predictive performance and the accuracy of the computed Shapley values. ViaSHAP using Kolmogorov-Arnold Networks showed superior predictive performance compared to multi-layer perceptron variants while competing favorably with state-of-the-art algorithms for tabular data XGBoost and Random Forests. ViaSHAP estimations showed a high similarity to the ground truth Shapley values, which can be controlled through the hyperparameters. One natural direction for future research is to implement ViaSHAP using state-of-the-art algorithms.Another direction is to use ViaSHAP to study possible adversarial attacks on a predictive model.

REFERENCES

Kjersti Aas, Martin Jullum, and Anders Løland. Explaining individual predictions when features are dependent: More accurate approximations to shapley values. *Artificial Intelligence*, 298:103502, 2021.

Maruan Al-Shedivat, Avinava Dubey, and Eric Xing. Contextual explanation networks. *J. Mach. Learn. Res.*, 21(1), 2022.

David Alvarez Melis and Tommi Jaakkola. Towards robust interpretability with self-explaining neural networks. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Marco Ancona, Cengiz Oztireli, and Markus Gross. Explaining deep neural networks with a polynomial time algorithm for shapley value approximation. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pp. 272–281, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

Zhan ao Huang, Yongsheng Sang, Yanan Sun, and Jiancheng Lv. A neural network learning algorithm for highly imbalanced data classification. *Information Sciences*, 612:496–513, 2022.

Sercan O. Arik and Tomas Pfister. Tabnet: Attentive interpretable tabular learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(8):6679–6687, 2021.

Hugh Chen, Joseph D. Janizek, Scott Lundberg, and Su-In Lee. True to the model or true to the data?, 2020. URL https://arxiv.org/abs/2006.16234.

Hugh Chen, Ian C. Covert, Scott M. Lundberg, and Su-In Lee. Algorithms to estimate shapley value feature attributions. *Nature Machine Intelligence*, 5(6):590–601, Jun 2023.

Jianbo Chen, Le Song, Martin Wainwright, and Michael Jordan. Learning to explain: An information-theoretic perspective on model interpretation. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pp. 883–892, 10–15 Jul 2018.

Jianbo Chen, Le Song, Martin J. Wainwright, and Michael I. Jordan. L-shapley and c-shapley: Efficient model interpretation for structured data. In *International Conference on Learning Representations*, 2019.

Ian Covert and Su-In Lee. Improving kernelshap: Practical shapley value estimation using linear regression. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130, pp. 3457–3465, April 2021.

Ian Covert, Scott M. Lundberg, and Su-In Lee. Explaining by removing: A unified framework for model explanation. *CoRR*, abs/2011.14878, 2020.

Ian Covert, Scott Lundberg, and Su-In Lee. Explaining by removing: A unified framework for model explanation. *Journal of Machine Learning Research*, 22(209):1–90, 2021.

Hejie Cui, Wei Dai, Yanqiao Zhu, Xiaoxiao Li, Lifang He, and Carl Yang. Interpretable graph neural networks for connectome-based brain disorder analysis. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2022: 25th International Conference, Singapore, September 18–22, 2022, Proceedings, Part VIII*, pp. 375–385, 2022.

G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4):303–314, Dec 1989.

Enyan Dai and Suhang Wang. Towards self-explainable graph neural network. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pp. 302–311, 2021.

Susanne Dandl, Christoph Molnar, Martin Binder, and Bernd Bischl. Multi-objective counterfactual explanations. In *Parallel Problem Solving from Nature – PPSN XVI*, pp. 448–469, 2020.

Aosong Feng, Chenyu You, Shiqiang Wang, and Leandros Tassiulas. Kergnns: Interpretable graph neural networks with graph kernels. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(6):6614–6622, Jun. 2022.

Thomas S. Ferguson. *Game Theory*. University of California at Los Angeles, 2018.

Milton Friedman. A correction: The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, 34(205):109–109, 1939.

Christopher Frye, Damien de Mijolla, Tom Begley, Laurence Cowton, Megan Stanley, and Ilya Feige. Shapley explainability on the data manifold. In *International Conference on Learning Representations*, 2021.

Bryce Goodman and Seth Flaxman. European union regulations on algorithmic decision-making and a "right to explanation". *AI Magazine*, 38(3):50–57, 2017.

Hangzhi Guo, Thanh Nguyen, and Amulya Yadav. Counternet: End-to-end training of counterfactual aware predictions. In *ICML 2021 Workshop on Algorithmic Recourse*, 2021.

Victor Guyomard, Françoise Fessant, Thomas Guyet, Tassadit Bouadi, and Alexandre Termier. Vc-net: A self-explaining model for realistic counterfactual generation. In *Machine Learning and Knowledge Discovery in Databases, ECML PKDD*, 2022.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4 (2):251–257, 1991.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

Cosimo Izzo, Aldo Lipani, Ramin Okhrati, and Francesca Medda. A baseline for shapley values in mlps: from missingness to neutrality. In *ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning.*, pp. 605–610, 10 2021. doi: 10.14428/esann/2021.ES2021-18.

Neil Jethani, Mukund Sudarshan, Yindalon Aphinyanaphongs, and Rajesh Ranganath. Have we learned to explain?: How interpretability methods can learn to encode predictions in their interpretations. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130, pp. 1459–1467, 13–15 Apr 2021.

Neil Jethani, Mukund Sudarshan, Ian Connick Covert, Su-In Lee, and Rajesh Ranganath. FastSHAP: Real-time shapley value estimation. In *International Conference on Learning Representations*, 2022.

Amir-Hossein Karimi, Gilles Barthe, Borja Balle, and Isabel Valera. Model-agnostic counterfactual explanations for consequential decisions. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108, pp. 895–905, 26–28 Aug 2020.

Andrei Nikolaevich Kolmogorov. On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition. *Doklady Akademii Nauk*, 114:953–956, 1957.

Andrey Nikolaevich Kolmogorov. On the representation of continuous functions of several variables as superpositions of continuous functions of a smaller number of variables. *Doklady Akademii Nauk*, 108(2):179–182, 1956.

Patrick Kolpaczki, Viktor Bengs, Maximilian Muschalik, and Eyke Hüllermeier. Approximating the shapley value without marginal contributions. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(12):13246–13255, Mar. 2024.

Michał Koziarski, Bartosz Krawczyk, and Michał Woźniak. Radial-based approach to imbalanced data oversampling. In *Hybrid Artificial Intelligent Systems: 12th International Conference, HAIS 2017, La Rioja, Spain, June 21-23, 2017, Proceedings 12*, pp. 318–327. Springer, 2017.

Alex Krizhevsky, Vinod Nair, Geoffrey Hinton, et al. The cifar-10 dataset. *online: http://www. cs. toronto. edu/kriz/cifar. html*, 55(5):2, 2014.

Himabindu Lakkaraju, Ece Kamar, Rich Caruana, and Jure Leskovec. Interpretable & explorable approximations of black box models. *CoRR*, abs/1707.01154, 2017.

Tao Lei, Regina Barzilay, and Tommi Jaakkola. Rationalizing neural predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 107–117, Austin, Texas, November 2016. Association for Computational Linguistics.

Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljačić, Thomas Y. Hou, and Max Tegmark. Kan: Kolmogorov-arnold networks, 2024. URL https://arxiv.org/abs/2404.19756.

Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 4768–4777, 2017.

Scott M. Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M. Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. From local explanations to global understanding with explainable ai for trees. *Nature Machine Intelligence*, 2(1):56–67, Jan 2020.

Mihai Manea. *Cooperative Games*. Massachusetts Institute of Technology, 2016.

Jean-Luc Marichal and Pierre Mathonet. Weighted banzhaf power and interaction indexes through weighted approximations of games. *European Journal of Operational Research*, 211(2):352–358, 2011.

Rory Mitchell, Joshua Cooper, Eibe Frank, and Geoffrey Holmes. Sampling permutations for shapley value estimation. *Journal of Machine Learning Research*, 23(1), jan 2022.

Ramaravind K. Mothilal, Amit Sharma, and Chenhao Tan. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pp. 607–617, 2020.

Peter Björn Nemenyi. *Distribution-free multiple comparisons*. PhD thesis, Princeton University, 1963.

Guillermo. Owen. *Game theory*. Academic Press,, New York :, 3rd ed. edition, 1995.

Neel Patel, Martin Strobel, and Yair Zick. High dimensional model explanations: An axiomatic approach. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pp. 401–411, 2021.

Marco Tulio Ribeiro, Sameer Singh, and Carlos 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*, KDD '16, pp. 1135–1144, 2016.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pp. 234–241. Springer International Publishing, 2015.

Yoshihide Sawada and Keigo Nakamura. Concept bottleneck model with additional unsupervised concepts. *IEEE Access*, 10:41758–41765, 2022.

Patrick Schwab and Walter Karlen. Cxplain: Causal explanations for model interpretation under uncertainty. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Lloyd S Shapley. A value for n-person games. In Harold W. Kuhn and Albert W. Tucker (eds.), *Contributions to the Theory of Games II*, pp. 307–317. Princeton University Press, Princeton, 1953.

Ravid Shwartz-Ziv and Amitai Armon. Tabular data: Deep learning is not all you need. *Information Fusion*, 81:84–90, 2022.

Xuelin Situ, Ingrid Zukerman, Cecile Paris, Sameen Maruf, and Gholamreza Haffari. Learning to explain: Generating stable explanations fast. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 5340–5355, August 2021.

C. Spearman. The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1):72–101, 1904.

Jacopo Teneggi, Alexandre Luster, and Jeremias Sulam. Fast hierarchical games for image explanations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

Arnaud Van Looveren and Janis Klaise. Interpretable counterfactual explanations guided by prototypes. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 650–665. Springer, 2021.

Frank Wilcoxon. Individual comparisons by ranking methods. biometrics bulletin 1, 6 (1945), 80–83. *URL http://www. jstor. org/stable/3001968*, 1945.

Han Xuanyuan, Pietro Barbiero, Dobrik Georgiev, Lucie Charlotte Magister, and Pietro Liò. Global concept-based interpretability for graph neural networks via neuron analysis. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(9):10675–10683, Jun. 2023.

Jinsung Yoon, James Jordon, and Mihaela van der Schaar. INVASE: Instance-wise variable selection using neural networks. In *International Conference on Learning Representations*, 2019.

H. P. Young. Monotonic solutions of cooperative games. *International Journal of Game Theory*, 14 (2), Jun 1985.

Zaixin Zhang, Qi Liu, Hao Wang, Chengqiang Lu, and Chee-Kong Lee. Protgnn: Towards self-explaining graph neural networks. In *AAAI*, 2022.

A PROOF OF LEMMA 1

By definition of $\mathcal{V}ia^{\text{SHAP}}$:

$$\mathcal{V}ia^{\textit{SHAP}}(\mathbf{x}) = \mathbf{1}^\top \phi^{\mathcal{V}ia}(\mathbf{x};\theta) = \sum_{i\in N} \phi_i^{\mathcal{V}ia}(\mathbf{x};\theta)$$

This is the definition of local accuracy for the game $v : S \mapsto \mathcal{V}ia^{\textit{SHAP}}(\mathbf{x}^S)$.

B PROOF OF LEMMA 2

Assume that the global minimizer $\phi^{\mathcal{V}ia}(\mathbf{x};\theta^*)$ of the loss function (6) does not satisfy the missingness property, i.e., there exists a feature i that has no impact on the prediction:

$$\mathcal{V}ia^{\textit{SHAP}}(\mathbf{x}^{S\cup\{i\}}) = \mathcal{V}ia^{\textit{SHAP}}(\mathbf{x}^S),\ \forall S \subseteq N \setminus \{i\} \tag{8}$$

However, the Shapley value ϕ_i assigned by $\phi^{\mathcal{V}ia}(\mathbf{x};\theta^*)$ is not zero ($\phi_i \neq 0$).

We recall the optimized loss function:

$$\mathcal{L}_\phi(\theta) = \sum_{\mathbf{x}\in X} \mathop{\mathbb{E}}_{p(S)} \Big[\Big(\mathcal{V}ia^{\textit{SHAP}}(\mathbf{x}^S) - \mathcal{V}ia^{\textit{SHAP}}(\mathbf{0}) - \mathbf{1}_S^\top \phi^{\mathcal{V}ia}(\mathbf{x};\theta) \Big)^2 \Big],$$

This loss is non-negative, and is thus minimized for a value of 0, implying all terms in the expectancy are equal to 0. In particular, for any set $S \subseteq N \setminus \{i\}$, we have:

$$\begin{aligned}
0 &= \begin{cases} \mathcal{V}ia^{\textit{SHAP}}(\mathbf{x}^{S\cup\{i\}}) - \mathcal{V}ia^{\textit{SHAP}}(\mathbf{0}) - \mathbf{1}_{S\cup\{i\}}^\top \phi^{\mathcal{V}ia}(\mathbf{x};\theta) \\ \mathcal{V}ia^{\textit{SHAP}}(\mathbf{x}^S) - \mathcal{V}ia^{\textit{SHAP}}(\mathbf{0}) - \mathbf{1}_S^\top \phi^{\mathcal{V}ia}(\mathbf{x};\theta) \end{cases} \\
&\Rightarrow \mathcal{V}ia^{\textit{SHAP}}(\mathbf{x}^{S\cup\{i\}}) - \mathbf{1}_{S\cup\{i\}}^\top \phi^{\mathcal{V}ia}(\mathbf{x};\theta) = \mathcal{V}ia^{\textit{SHAP}}(\mathbf{x}^S) - \mathbf{1}_S^\top \phi^{\mathcal{V}ia}(\mathbf{x};\theta) \\
&\Rightarrow \mathcal{V}ia^{\textit{SHAP}}(\mathbf{x}^S) - \mathbf{1}_{S\cup\{i\}}^\top \phi^{\mathcal{V}ia}(\mathbf{x};\theta) = \mathcal{V}ia^{\textit{SHAP}}(\mathbf{x}^S) - \mathbf{1}_S^\top \phi^{\mathcal{V}ia}(\mathbf{x};\theta) \\
&\Rightarrow \sum_{j\in S\cup\{i\}} \phi_j^{\mathcal{V}ia}(\mathbf{x};\theta^*) = \sum_{j\in S} \phi_j^{\mathcal{V}ia}(\mathbf{x};\theta^*) \\
&\Rightarrow \phi_i^{\mathcal{V}ia}(\mathbf{x};\theta^*) = 0
\end{aligned}$$

In practice, it is unlikely for a loss to exactly reach its global optimum. Instead, it approximates it. We assume here that the loss has reached a value ϵ^2 for an $\epsilon \geq 0$. We propose an upper bound on $\phi_i^{\mathcal{V}ia}(x;\theta)$ conditioned on ϵ.

Since the loss is composed only of non-negative terms, this means that:

$$\begin{aligned}
&\forall S \subseteq N,\ \Big(\mathcal{V}ia^{\textit{SHAP}}(\mathbf{x}^S) - \mathcal{V}ia^{\textit{SHAP}}(\mathbf{0}) - \mathbf{1}_S^\top \phi^{\mathcal{V}ia}(\mathbf{x};\theta) \Big)^2 \leq \epsilon^2 \\
&\Rightarrow \Big| \mathcal{V}ia^{\textit{SHAP}}(\mathbf{x}^S) - \mathcal{V}ia^{\textit{SHAP}}(\mathbf{0}) - \mathbf{1}_S^\top \phi^{\mathcal{V}ia}(\mathbf{x};\theta) \Big| \leq \epsilon
\end{aligned}$$

$$\epsilon \geq \begin{cases} \left|\mathcal{V}ia^{\textit{SHAP}}(\mathbf{x}^{S\cup\{i\}}) - \mathcal{V}ia^{\textit{SHAP}}(\mathbf{0}) - \mathbf{1}_{S\cup\{i\}}^{\top}\phi^{\mathcal{V}ia}(\mathbf{x};\theta)\right| \\ \left|\mathcal{V}ia^{\textit{SHAP}}(\mathbf{x}^{S}) - \mathcal{V}ia^{\textit{SHAP}}(\mathbf{0}) - \mathbf{1}_{S}^{\top}\phi^{\mathcal{V}ia}(\mathbf{x};\theta)\right| \end{cases}$$

$$\Rightarrow \left|\mathcal{V}ia^{\textit{SHAP}}(\mathbf{x}^{S\cup\{i\}}) - \mathcal{V}ia^{\textit{SHAP}}(\mathbf{0}) - \mathbf{1}_{S\cup\{i\}}^{\top}\phi^{\mathcal{V}ia}(\mathbf{x};\theta) - \mathcal{V}ia^{\textit{SHAP}}(\mathbf{x}^{S}) + \mathcal{V}ia^{\textit{SHAP}}(\mathbf{0}) + \mathbf{1}_{S}^{\top}\phi^{\mathcal{V}ia}(\mathbf{x};\theta)\right| \leq 2\epsilon$$

$$\Rightarrow \left|\mathcal{V}ia^{\textit{SHAP}}(\mathbf{x}^{S}) - \mathbf{1}_{S\cup\{i\}}^{\top}\phi^{\mathcal{V}ia}(\mathbf{x};\theta) - \mathcal{V}ia^{\textit{SHAP}}(\mathbf{x}^{S}) + \mathbf{1}_{S}^{\top}\phi^{\mathcal{V}ia}(\mathbf{x};\theta)\right| \leq 2\epsilon \text{ by equation 8}$$

$$\Rightarrow \left|\sum_{j\in S\cup\{i\}}\phi_j^{\mathcal{V}ia}(\mathbf{x};\theta) - \sum_{j\in S}\phi_j^{\mathcal{V}ia}(\mathbf{x};\theta)\right| \leq 2\epsilon$$

$$\Rightarrow \left|\phi_i^{\mathcal{V}ia}(\mathbf{x};\theta)\right| \leq 2\epsilon$$

$$\Rightarrow \left|\phi_i^{\mathcal{V}ia}(\mathbf{x};\theta)\right| \leq 2\mathcal{L}_{\phi}(\theta)$$

Thus, as the loss function converges to 0, so does the importance attributed to features with no influence on the outcome.

C Proof of Lemma 3

Since both $\mathcal{V}$ and $\mathcal{V}'$ optimize their respective targets, they satisfy efficiency, i.e.:

$$\forall S \subseteq N,\ \ \mathcal{V}(\mathbf{x}^S) = \mathbf{1}_S^{\top}\phi^{\mathcal{V}ia}(\mathbf{x};\theta^*);\ \mathcal{V}'(\mathbf{x}^S) = \mathbf{1}_S^{\top}\phi'^{\mathcal{V}ia}(\mathbf{x};\theta^{*'}) \tag{9}$$

Then:

$$\forall S \subseteq N\setminus\{i\},$$
$$\mathcal{V}(\mathbf{x}^{S\cup\{i\}}) - \mathcal{V}(\mathbf{x}^S) \geq \mathcal{V}'(\mathbf{x}^{S\cup\{i\}}) - \mathcal{V}'(\mathbf{x}^S)$$
$$\Rightarrow \sum_{j\in S\cup\{i\}}\phi_j^{\mathcal{V}ia}(\mathbf{x};\theta^*) - \sum_{j\in S}\phi_j^{\mathcal{V}ia}(\mathbf{x};\theta^*) \geq \sum_{j\in S\cup\{i\}}\phi_j^{\mathcal{V}ia}(\mathbf{x};\theta^{*'}) - \sum_{j\in S}\phi_j^{\mathcal{V}ia}(\mathbf{x};\theta^{*'})$$
$$\Rightarrow \phi_i^{\mathcal{V}ia}(\mathbf{x};\theta^*) \geq \phi_i^{\mathcal{V}ia}(\mathbf{x};\theta^{*'})$$

In the same way as for the Lemma 2, the proof assumes perfect minimization of the loss. Thus, we propose a relaxed variant, where the loss term $\mathcal{L}_{\phi}(\theta)$ was minimized down to ϵ^2 with $\epsilon \geq 0$. Thus, following similar reasoning as in the proof of Lemma 2, we have that $\forall S$:

$$\left|\mathcal{V}ia^{\textit{SHAP}}(\mathbf{x}^{S}) - \mathcal{V}ia^{\textit{SHAP}}(\mathbf{0}) - \mathbf{1}_{S}^{\top}\phi^{\mathcal{V}ia}(\mathbf{x};\theta)\right| \leq \epsilon$$

We also have:

$$\left|\mathcal{V}ia^{\textit{SHAP}}(\mathbf{x}^{S}) - \mathbf{1}_{S}^{\top}\phi^{\mathcal{V}ia}(\mathbf{x};\theta)\right| = \left|\mathcal{V}ia^{\textit{SHAP}}(\mathbf{x}^{S}) - \mathbf{1}_{S}^{\top}\phi^{\mathcal{V}ia}(\mathbf{x};\theta) - \mathcal{V}ia^{\textit{SHAP}}(\mathbf{0}) + \mathcal{V}ia^{\textit{SHAP}}(\mathbf{0})\right|$$

By the triangle inequality on the right-hand side:

$$\left|\mathcal{V}ia^{\textit{SHAP}}(\mathbf{x}^{S}) - \mathbf{1}_{S}^{\top}\phi^{\mathcal{V}ia}(\mathbf{x};\theta)\right| \leq \left|\mathcal{V}ia^{\textit{SHAP}}(\mathbf{x}^{S}) - \mathbf{1}_{S}^{\top}\phi^{\mathcal{V}ia}(\mathbf{x};\theta) - \mathcal{V}ia^{\textit{SHAP}}(\mathbf{0})\right| + \left|\mathcal{V}ia^{\textit{SHAP}}(\mathbf{0})\right|$$

But observe that all features in $\mathbf{0}$ are non-contributive since, $\forall S \subseteq N$, $\mathbf{0}^S = \mathbf{0}$ by definition of the masking operation. Thus, by the bound found in Lemma 2: $\forall i \in N, \left|\phi_i(\mathbf{0},\theta)\right| \leq 2\epsilon$. Thus $\left|\mathcal{V}ia^{\textit{SHAP}}(\mathbf{0})\right| \leq 2n\epsilon$.

Thus:

$$\left|\mathcal{V}ia^{SHAP}(\mathbf{x}^S) - \mathbf{1}_S^\top \phi^{\mathcal{V}ia}(\mathbf{x};\theta) - \mathcal{V}ia^{SHAP}(\mathbf{0})\right| + \left|\mathcal{V}ia^{SHAP}(\mathbf{0})\right| \leq \epsilon + 2n\epsilon$$

and we thus derive the following upper bound on the ϕ_i-wise error as:

$$\left|\mathcal{V}ia^{SHAP}(\mathbf{x}^S) - \mathbf{1}_S^\top \phi^{\mathcal{V}ia}(\mathbf{x};\theta)\right| \leq \epsilon(2n+1)$$

.

D Predictive Performance

We evaluated the performance of the four variants of $\mathcal{V}ia^{SHAP}$ implementations mentioned in the experimental setup, i.e., $KAN^{\mathcal{V}ia}$, $KAN_{\varrho}^{\mathcal{V}ia}$, $MLP^{\mathcal{V}ia}$, and $MLP_{\theta}^{\mathcal{V}ia}$, are compared to the following algorithms for structured data: Random Forests, XGBoost, and TabNet, where Random Forests and XGBoost result in black-box models, while TabNet is explainable by visualizing feature selection masks that highlight important features. The predictive performance evaluation is conducted using 25 datasets. The results show that $KAN^{\mathcal{V}ia}$ comes in first place as the best-performing classifier, followed by XGBoost and $KAN_{\varrho}^{\mathcal{V}ia}$, based on AUC values.

The Friedman test confirmed that the differences in predictive performance are statistically significant at the 0.05 level. A subsequent post-hoc Nemenyi test revealed that while the differences between KAN-based implementations and tree ensemble models (XGBoost and Random Forests) are statistically insignificant, the performance differences between $KAN^{\mathcal{V}ia}$ and MLP variants are significant. Moreover, the differences between KANVia and TabNet are also statistically significant. The ranking of the seven models on the 25 datasets and the results of the post-hoc Nemenyi test are illustrated in Figure 3. The detailed results on the 25 datasets are shown in Table 1.

While the MLP variants of $\mathcal{V}ia^{SHAP}$ significantly underperformed compared to the KAN variants, their performance can still be enhanced by using, for instance, deeper and more expressive models, particularly for datasets with high dimensionality and large training sets. However, we defer the task of improving MLP-based $\mathcal{V}ia^{SHAP}$ implementations to future work, as the core concept of $\mathcal{V}ia^{SHAP}$ can be integrated with any deep learning model. More importantly, $\mathcal{V}ia^{SHAP}$ is not limited to structured data and can be incorporated easily into the training loop of models in computer vision and natural language processing.

Table 1: The AUC of $KAN^{\mathcal{V}ia}$, $KAN_{\varrho}^{\mathcal{V}ia}$, $MLP^{\mathcal{V}ia}$, and $MLP_{\theta}^{\mathcal{V}ia}$, TabNet, Random Forests, and XGBoost. The best-performing model is colored in light green, and the second best-performing is colored in light blue.

Dataset	$KAN^{\mathcal{V}ia}$	$KAN_{\varrho}^{\mathcal{V}ia}$	$MLP^{\mathcal{V}ia}$	$MLP_{\theta}^{\mathcal{V}ia}$	TabNet	Random Forests	XGBoost
Abalone	0.87 ± 0.003	0.871 ± 0.003	0.877 ± 0.003	0.878 ± 0.004	0.873 ± 0.01	0.875 ± 0.001	0.868
Ada Prior	0.89 ± 0.005	0.899 ± 0.002	0.887 ± 0.005	0.878 ± 0.005	0.82 ± 0.062	0.885 ± 0.001	0.887
Adult	0.914 ± 0.003	0.916 ± 0.001	0.91 ± 0.007	0.912 ± 0.006	0.911 ± 0.001	0.913 ± 0.001	0.928
Bank32nh	0.878 ± 0.001	0.876 ± 0.005	0.873 ± 0.002	0.872 ± 0.005	0.862 ± 0.005	0.875 ± 0.003	0.875
Electricity	0.93 ± 0.004	0.92 ± 0.004	0.864 ± 0.003	0.87 ± 0.008	0.88 ± 0.009	0.97 ± 0.0004	0.973
Elevators	0.935 ± 0.002	0.939 ± 0.001	0.922 ± 0.031	0.938 ± 0.003	0.942 ± 0.002	0.909 ± 0.001	0.935
Fars	0.96 ± 0.0003	0.955 ± 0.003	0.953 ± 0.001	0.952 ± 0.002	0.953 ± 0.0008	0.951 ± 0.0003	0.963
Helena	0.884 ± 0.0001	0.881 ± 0.0005	0.88 ± 0.002	0.881 ± 0.002	0.883 ± 0.002	0.855 ± 0.001	0.874
Heloc	0.788 ± 0.002	0.784 ± 0.002	0.775 ± 0.006	0.773 ± 0.004	0.774 ± 0.007	0.779 ± 0.001	0.767
Higgs	0.801 ± 0.001	0.805 ± 0.003	0.786 ± 0.006	0.786 ± 0.005	0.801 ± 0.003	0.79 ± 0.001	0.799
LHC Identify Jet	0.944 ± 0.0001	0.942 ± 0.0003	0.931 ± 0.001	0.932 ± 0.002	0.942 ± 0.0008	0.935 ± 0.0002	0.941
House 16H	0.949 ± 0.0007	0.944 ± 0.001	0.929 ± 0.005	0.932 ± 0.01	0.94 ± 0.003	0.955 ± 0.0004	0.952
Indian Pines	0.985 ± 0.0004	0.885 ± 0.086	0.918 ± 0.014	0.692 ± 0.191	0.983 ± 0.003	0.979 ± 0.0005	0.987
Jannis	0.864 ± 0.001	0.861 ± 0.0005	0.571 ± 0.159	0.569 ± 0.151	0.864 ± 0.003	0.861 ± 0.0002	0.871
JM1	0.732 ± 0.003	0.74 ± 0.006	0.719 ± 0.01	0.717 ± 0.005	0.726 ± 0.004	0.746 ± 0.004	0.708
Magic Telescope	0.929 ± 0.001	0.927 ± 0.001	0.917 ± 0.004	0.917 ± 0.004	0.929 ± 0.001	0.933 ± 0.0005	0.935
MC1	0.94 ± 0.003	0.93 ± 0.013	0.909 ± 0.014	0.896 ± 0.01	0.916 ± 0.021	0.845 ± 0.002	0.934
Microaggregation2	0.783 ± 0.002	0.765 ± 0.003	0.746 ± 0.005	0.733 ± 0.029	0.759 ± 0.014	0.768 ± 0.0008	0.781
Mozilla4	0.968 ± 0.0008	0.962 ± 0.001	0.948 ± 0.002	0.947 ± 0.001	0.96 ± 0.004	0.988 ± 0.0007	0.989
Satellite	0.996 ± 0.001	0.992 ± 0.002	0.997 ± 0.001	0.996 ± 0.001	0.991 ± 0.002	0.998 ± 0.0004	0.992
PC2	0.827 ± 0.009	0.818 ± 0.032	0.685 ± 0.088	0.662 ± 0.016	0.631 ± 0.16	0.631 ± 0.05	0.646
Phonemes	0.946 ± 0.003	0.936 ± 0.004	0.904 ± 0.004	0.894 ± 0.032	0.918 ± 0.014	0.965 ± 0.002	0.951
Pollen	0.515 ± 0.006	0.496 ± 0.007	0.504 ± 0.007	0.5 ± 0.014	0.493 ± 0.012	0.485 ± 0.006	0.475
Telco Customer Churn	0.854 ± 0.003	0.852 ± 0.003	0.843 ± 0.003	0.839 ± 0.004	0.832 ± 0.004	0.847 ± 0.002	0.846
1st order theorem proving	0.822 ± 0.002	0.695 ± 0.024	0.737 ± 0.013	0.64 ± 0.093	0.727 ± 0.01	0.855 ± 0.001	0.858

E Explanations Accuracy Evaluation

The explainability of the four implementations of ViaSHAP, based on MLP and KAN, were evaluated by comparing their Shapley values ($\phi^{\text{Via}}(x;\theta)$) to the ground truth Shapley values (ϕ). As mentioned in the experimental set, the ground truth Shapley values were generated by KernelSHAP after convergence on each example in the test set. In the explainability evaluation, we used the models trained with default hyperparameters in the predictive performance evaluation, which generally showed high similarity to the ground truth, as demonstrated by the cosine similarity measurements. The Friedman test found no significant differences in the cosine similarity between the compared algorithms over the 25 datasets. The detailed results are available in Table 2.

Table 2: The cosine similarity of the ground truth Shapley values to the Shapley values obtained from $KAN^{\mathcal{V}ia}$, $KAN^{\mathcal{V}ia}_{\varrho}$, $MLP^{\mathcal{V}ia}$, and $MLP^{\mathcal{V}ia}_{\theta}$. The best-performing model is colored in light green.

Dataset	$KAN^{\mathcal{V}ia}$	$KAN^{\mathcal{V}ia}_{\varrho}$	$MLP^{\mathcal{V}ia}$	$MLP^{\mathcal{V}ia}_{\theta}$
Abalone	**0.969 ± 0.0166**	0.966 ± 0.013	0.647 ± 0.21	0.807 ± 0.214
Ada Prior	0.935 ± 0.046	**0.982 ± 0.006**	0.663 ± 0.142	0.908 ± 0.045
Adult	0.931 ± 0.049	**0.992 ± 0.011**	0.574 ± 0.16	0.947 ± 0.032
Bank32nh	0.779 ± 0.163	0.713 ± 0.187	0.794 ± 0.166	**0.876 ± 0.084**
Electricity	0.970 ± 0.02	**0.971 ± 0.017**	0.912 ± 0.131	0.913 ± 0.09
Elevators	0.966 ± 0.024	0.966 ± 0.026	0.976 ± 0.025	**0.976 ± 0.02**
Fars	0.886 ± 0.253	0.886 ± 0.28	**0.95 ± 0.104**	0.943 ± 0.058
Helena	**0.856 ± 0.092**	0.715 ± 0.157	0.840 ± 0.099	0.789 ± 0.104
Heloc	**0.844 ± 0.111**	0.671 ± 0.182	0.759 ± 0.176	0.832 ± 0.125
Higgs	0.917 ± 0.068	**0.925 ± 0.062**	0.92 ± 0.093	0.912 ± 0.097
LHC Identify Jet	0.971 ± 0.021	0.952 ± 0.065	0.97 ± 0.042	**0.972 ± 0.041**
House 16H	0.919 ± 0.048	0.922 ± 0.043	0.927 ± 0.06	**0.944 ± 0.048**
Indian Pines	**0.796 ± 0.121**	0.241 ± 0.07	0.304 ± 0.077	0.325 ± 0.084
Jannis	**0.852 ± 0.141**	0.546 ± 0.189	0.675 ± 0.13	0.439 ± 0.164
JM1	**0.88 ± 0.044**	0.667 ± 0.217	0.795 ± 0.203	0.839 ± 0.159
Magic Telescope	0.922 ± 0.067	0.935 ± 0.058	**0.973 ± 0.035**	0.962 ± 0.058
MC1	0.466 ± 0.268	0.794 ± 0.084	0.777 ± 0.127	**0.887 ± 0.055**
Microaggregation2	**0.938 ± 0.049**	0.610 ± 0.149	0.840 ± 0.099	0.81 ± 0.096
Mozilla4	0.953 ± 0.023	0.948 ± 0.016	0.975 ± 0.018	**0.979 ± 0.022**
Satellite	0.841 ± 0.116	**0.870 ± 0.077**	0.766 ± 0.159	0.861 ± 0.093
PC2	0.534 ± 0.183	**0.905 ± 0.053**	0.786 ± 0.137	0.827 ± 0.098
Phonemes	0.811 ± 0.162	0.868 ± 0.082	0.873 ± 0.126	**0.916 ± 0.083**
Pollen	**0.952 ± 0.059**	0.945 ± 0.023	0.464 ± 0.476	0.592 ± 0.439
Telco Customer Churn	0.81 ± 0.108	**0.904 ± 0.051**	0.43 ± 0.189	0.592 ± 0.231
1st order theorem proving	**0.725 ± 0.179**	0.464 ± 0.517	0.387 ± 0.182	0.539 ± 0.144

We also measured similarity in ranking the important features between the computed Shapley values ($\phi^{\text{Via}}(x;\theta)$) and the ground truth Shapley values (ϕ) using the Spearman rank correlation coefficient. $KAN^{\mathcal{V}ia}$ is ranked first with respect to the correlation values across the 25 datasets, followed by both $KAN^{\mathcal{V}ia}\varrho$ and $MLP^{\mathcal{V}ia}\theta$ in the second place, and $MLP^{\mathcal{V}ia}$ in the last place. The Spearman rank test revealed that the observed differences are significant. Subsequently, the post-hoc Nemenyi test confirmed that $MLP^{\mathcal{V}ia}$ significantly underperformed the compared algorithms, while the differences between the remaining algorithms are insignificant. Overall, if both the cosine similarity and the Spearman rank are considered, $KAN^{\mathcal{V}ia}$ proved to be a more stable approximator, as detailed in Tables 2 and 3.

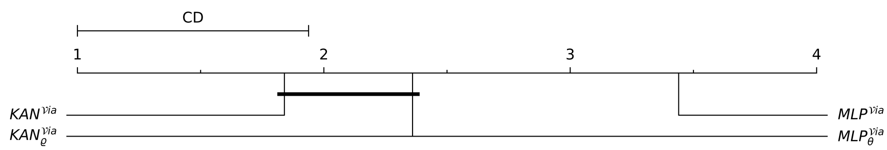

Figure 6: **The average rank** of $KAN^{\mathcal{V}ia}$, $KAN_{\varrho}^{\mathcal{V}ia}$, $MLP^{\mathcal{V}ia}$, and $MLP_{\theta}^{\mathcal{V}ia}$ on the 25 datasets with respect to the Spearman correlation between the ground truth Shapley values and the values obtained from the compared models. A lower rank is better and the critical difference (CD) represents the largest difference that is not statistically significant.

Table 3: The Spearman rank correlation between the ground truth Shapley values and the Shapley values obtained from $KAN^{\mathcal{V}ia}$, $KAN_{\varrho}^{\mathcal{V}ia}$, and $MLP^{\mathcal{V}ia}$. The best-performing model is colored in light green .

Dataset	$KAN^{\mathcal{V}ia}$	$KAN_{\varrho}^{\mathcal{V}ia}$	$MLP^{\mathcal{V}ia}$	$MLP_{\theta}^{\mathcal{V}ia}$
Abalone	0.663 ± 0.234	0.879 ± 0.14	0.529 ± 0.246	0.649 ± 0.236
Ada Prior	0.876 ± 0.088	0.962 ± 0.025	0.576 ± 0.163	0.869 ± 0.081
Adult	0.959 ± 0.035	0.932 ± 0.034	0.398 ± 0.214	0.864 ± 0.084
Bank32nh	0.432 ± 0.151	0.433 ± 0.139	0.349 ± 0.15	0.486 ± 0.129
Electricity	0.798 ± 0.183	0.838 ± 0.142	0.751 ± 0.206	0.848 ± 0.137
Elevators	0.920 ± 0.064	0.888 ± 0.072	0.883 ± 0.07	0.902 ± 0.06
Fars	0.347 ± 0.328	0.106 ± 0.133	0.512 ± 0.164	0.491 ± 0.115
Helena	0.669 ± 0.152	0.475 ± 0.188	0.656 ± 0.159	0.660 ± 0.168
Heloc	0.741 ± 0.147	0.673 ± 0.159	0.589 ± 0.173	0.701 ± 0.143
Higgs	0.674 ± 0.12	0.718 ± 0.112	0.535 ± 0.143	0.568 ± 0.139
LHC Identify Jet	0.857 ± 0.119	0.726 ± 0.184	0.737 ± 0.164	0.724 ± 0.146
House 16H	0.888 ± 0.092	0.858 ± 0.102	0.823 ± 0.112	0.864 ± 0.095
Indian Pines	0.699 ± 0.116	0.057 ± 0.054	0.099 ± 0.07	0.181 ± 0.056
Jannis	0.477 ± 0.131	0.314 ± 0.174	0.343 ± 0.132	0.227 ± 0.137
JM1	0.756 ± 0.202	0.682 ± 0.223	0.59 ± 0.188	0.715 ± 0.189
Magic Telescope	0.9 ± 0.098	0.91 ± 0.087	0.882 ± 0.098	0.828 ± 0.141
MC1	0.621 ± 0.157	0.885 ± 0.088	0.619 ± 0.169	0.716 ± 0.108
Microaggregation2	0.876 ± 0.096	0.411 ± 0.183	0.656 ± 0.159	0.705 ± 0.2
Mozilla4	0.942 ± 0.092	0.971 ± 0.063	0.909 ± 0.161	0.913 ± 0.137
Satellite	0.746 ± 0.212	0.786 ± 0.151	0.677 ± 0.208	0.8 ± 0.132
PC2	0.733 ± 0.161	0.924 ± 0.09	0.675 ± 0.154	0.737 ± 0.135
Phonemes	0.941 ± 0.103	0.954 ± 0.083	0.807 ± 0.213	0.862 ± 0.159
Pollen	0.285 ± 0.442	0.171 ± 0.484	0.297 ± 0.498	0.407 ± 0.545
Telco Customer Churn	0.848 ± 0.098	0.938 ± 0.043	0.262 ± 0.297	0.471 ± 0.211
1st order theorem proving	0.623 ± 0.188	0.082 ± 0.145	0.183 ± 0.146	0.367 ± 0.14

F IMAGE EXPERIMENTS

We implemented $\mathcal{V}ia^{SHAP}$ for image classification using three architectures: ResNet50 (He et al., 2016) (*ResNet50*$^{\mathcal{V}ia}$), ResNet18 (*ResNet18*$^{\mathcal{V}ia}$), and U-Net (Ronneberger et al., 2015) (*U-Net*$^{\mathcal{V}ia}$). The predictive performance of these models was evaluated using Top-1 Accuracy, with the results summarized in Table 4. All models were trained on the CIFAR-10 (Krizhevsky et al., 2014) dataset without transfer learning or pre-trained weights (i.e., trained from scratch) using four masks (samples) per data instance. The training incorporated early stopping, terminating after ten epochs without improvement on a validation split (10% of the training data). The results of evaluating the performance of the trained models on the test set demonstrate that $\mathcal{V}ia^{SHAP}$ can achieve high predictive performance on standard image classification tasks.

Table 4: A comparison of the predictive performance of *ResNet50*$^{\mathcal{V}ia}$, *ResNet18*$^{\mathcal{V}ia}$, and *U-Net*$^{\mathcal{V}ia}$ measured in AUC.

Dataset	AUC	0.95 Confidence Interval
U-Net$^{\mathcal{V}ia}$	**0.983**	**(0.981, 0.986)**
ResNet18$^{\mathcal{V}ia}$	0.968	(0.964, 0.971)
ResNet50$^{\mathcal{V}ia}$	0.96	(0.956, 0.964)

In order to assess the accuracy of the Shapley values computed by $\mathcal{V}ia^{SHAP}$ implementations, we followed a methodology similar to Jethani et al. (2022). Specifically, we selected the top 50% most important features identified by the explainer and evaluated the predictive performance of the explained model under two conditions: using only the selected top features (Inclusion Accuracy) and excluding the top features (Exclusion Accuracy).

We compared the accuracy of Shapley value approximations of the three models (*ResNet50*$^{\mathcal{V}ia}$, *ResNet18*$^{\mathcal{V}ia}$, and *U-Net*$^{\mathcal{V}ia}$). We also evaluated the accuracy of FastSHAP's approximations where the three $\mathcal{V}ia^{SHAP}$ implementations for image classification are provided as black boxes to FastSHAP. The results indicate that the $\mathcal{V}ia^{SHAP}$ implementations consistently provide more accurate Shapley value approximations than those generated by FastSHAP, as shown in Table 5. We also show the effects of using different percentages of the top features considered for inclusion and exclusion on the top-1 accuracy in Figure 7.

Table 5: The accuracy of the Shapley values is evaluated using the top 50% of the most important features (according to their Shapley values). The Inclusion AUC (higher values are better) and the Exclusion AUC (lower values are better) are computed using the top 1 accuracy.

Dataset	Exclusion AUC	0.95 Confidence Interval	Inclusion AUC	0.95 Confidence Interval
U-Net$^{\mathcal{V}ia}$	0.773	(0.747, 0.799)	0.988	(0.981, 0.995)
FastSHAP(*U-Net*$^{\mathcal{V}ia}$)	0.864	(0.843, 0.885)	0.978	(0.969, 0.987)
ResNet18$^{\mathcal{V}ia}$	0.611	(0.581, 0.642)	0.99	(0.983, 0.996)
FastSHAP(*ResNet18*$^{\mathcal{V}ia}$)	0.755	(0.728, 0.782)	0.954	(0.941, 0.967)
ResNet50$^{\mathcal{V}ia}$	**0.554**	**(0.523, 0.585)**	**0.997**	**(0.994, 1.0)**
FastSHAP(*ResNet50*$^{\mathcal{V}ia}$)	0.778	(0.753, 0.804)	0.978	(0.969, 0.987)

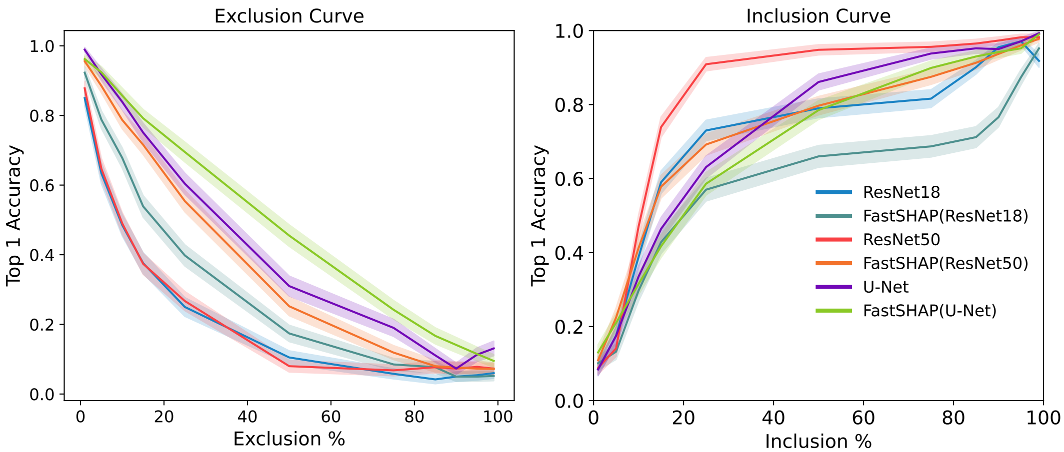

Figure 7: The inclusion and exclusion curves of ViaSHAP implementations as well as their FastSHAP explainers. We show how the top-1 accuracy of the predictive model changes as we exclude or include an increasing share of the important features, where the important features are determined by each explainer in the comparison.

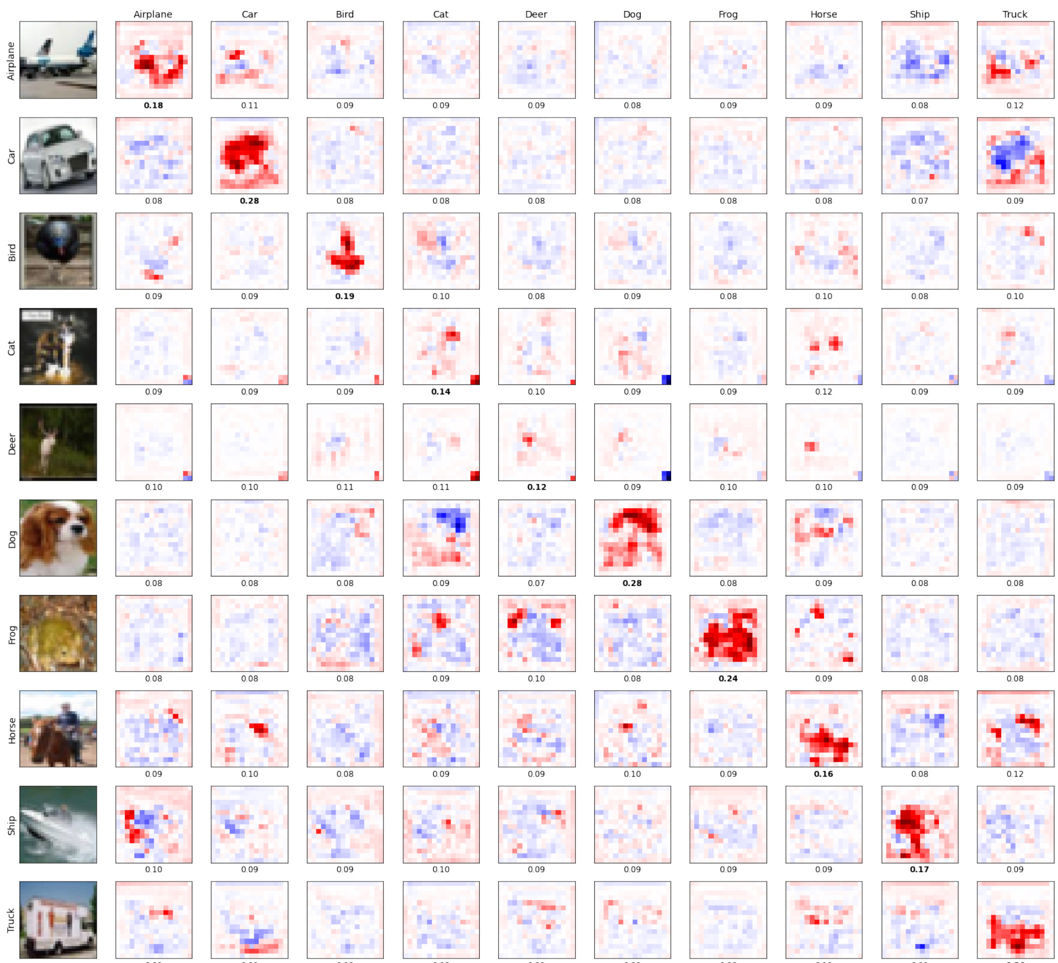

Figure 8: The explanations of *ResNet18*$^{\mathcal{V}ia}$ for 10 randomly selected predictions on the CIFAR-10 dataset. Each column corresponds to a CIFAR-10 class, and the predicted probability by *ResNet18*$^{\mathcal{V}ia}$ displayed beneath each image.

G A Comparison Between ViaSHAP and a KAN Model with the Same Architecture

We conducted an experiment to assess the impact of incorporating Shapley loss in the optimization process on predictive performance of a KAN model. Consequently, we compared $KAN^{\mathcal{V}ia}$ to a KAN model with an identical architecture that does not compute Shapley values. As summarized in Table 6, the results indicate that $KAN^{\mathcal{V}ia}$ generally outperforms the KAN model with the same architecture. In order to determine the statistical significance of these results, the Wilcoxon signed-rank test (Wilcoxon, 1945) was employed to test the null hypothesis that no difference exists in predictive performance, as measured by AUC, between $KAN^{\mathcal{V}ia}$ and the identical KAN model without Shapley values. The test results allowed for the rejection of the null hypothesis, indicating that $KAN^{\mathcal{V}ia}$ significantly outperforms the KAN architecture that is not optimized to compute Shapley values with respect to the predictive performance as measured by the AUC.

Table 6: A comparison between the predictive performance of $KAN^{\mathcal{V}ia}$ and a KAN model with an identical architecture to $KAN^{\mathcal{V}ia}$ but does not compute the Shapley values. The results are reported in AUC.

Dataset	KAN	$KAN^{\mathcal{V}ia}$
Abalone	0.882 ± 0.001	0.87 ± 0.003
Ada Prior	0.895 ± 0.005	0.89 ± 0.005
Adult	0.917 ± 0.001	0.914 ± 0.003
Bank32nh	0.886 ± 0.001	0.878 ± 0.001
Electricity	0.924 ± 0.005	0.93 ± 0.004
Elevators	0.935 ± 0.003	0.935 ± 0.002
Fars	0.957 ± 0.001	0.96 ± 0.0003
Helena	0.883 ± 0.001	0.884 ± 0.0001
Heloc	0.793 ± 0.002	0.788 ± 0.002
Higgs	0.801 ± 0.002	0.801 ± 0.001
LKC Identify Jet	0.944 ± 0.0003	0.944 ± 0.0001
House 16H	0.948 ± 0.001	0.949 ± 0.0007
Indian Pines	0.935 ± 0.001	0.985 ± 0.0004
Jannis	0.860 ± 0.002	0.864 ± 0.001
JM1	0.725 ± 0.008	0.732 ± 0.003
Magic Telescope	0.931 ± 0.001	0.929 ± 0.001
MC1	0.933 ± 0.019	0.94 ± 0.003
Microaggregation2	0.783 ± 0.002	0.783 ± 0.002
Mozilla4	0.967 ± 0.001	0.968 ± 0.0008
Satellite	0.987 ± 0.003	0.996 ± 0.001
PC2	0.458 ± 0.049	0.827 ± 0.009
Phonemes	0.945 ± 0.002	0.946 ± 0.003
Pollen	0.491 ± 0.005	0.515 ± 0.006
Telco Customer Churn	0.848 ± 0.005	0.854 ± 0.003
1st order theorem proving	0.805 ± 0.005	0.822 ± 0.002

H Ablation Study

In this section, we explore the influence of key hyperparameters on the performance and behavior of $\mathcal{V}ia^{SHAP}$. Specifically, we investigate the effects of the scaling hyperparameter β and the number of sampled coalitions per data instance. We begin by analyzing how variations in β impact both predictive performance and the accuracy of the Shapley values generated by $\mathcal{V}ia^{SHAP}$. We then examine the role of the number of sampled coalitions in model performance, followed by an evaluation of how changes in β affect the progress of the computed loss values during training. The findings provide valuable insights into the robustness and efficiency of $\mathcal{V}ia^{SHAP}$ under different hyperparameter settings.

H.1 The Impact of Scaling Hyperparameter β on the Performance of ViaSHAP

We evaluated the performance of the models trained with different β values (in equation 7), where exponentially increasing values are tested. The models were trained using the default hyperparameter settings described in the experimental setup, except for the values of β. The AUC of the trained models is measured on the test set, as well as the similarity of the predicted Shapley values to the ground truth. The results indicate that the predictive performance of $\mathcal{V}ia^{SHAP}$, as measured by the area under the ROC curve, remains largely unaffected by the value of β, even when β is increased exponentially. On the other hand, the similarity between the computed Shapley values and the ground truth improves as β increases. However, the model struggles to learn effectively after β exceeds 200, as shown in Figures 9 and 10.

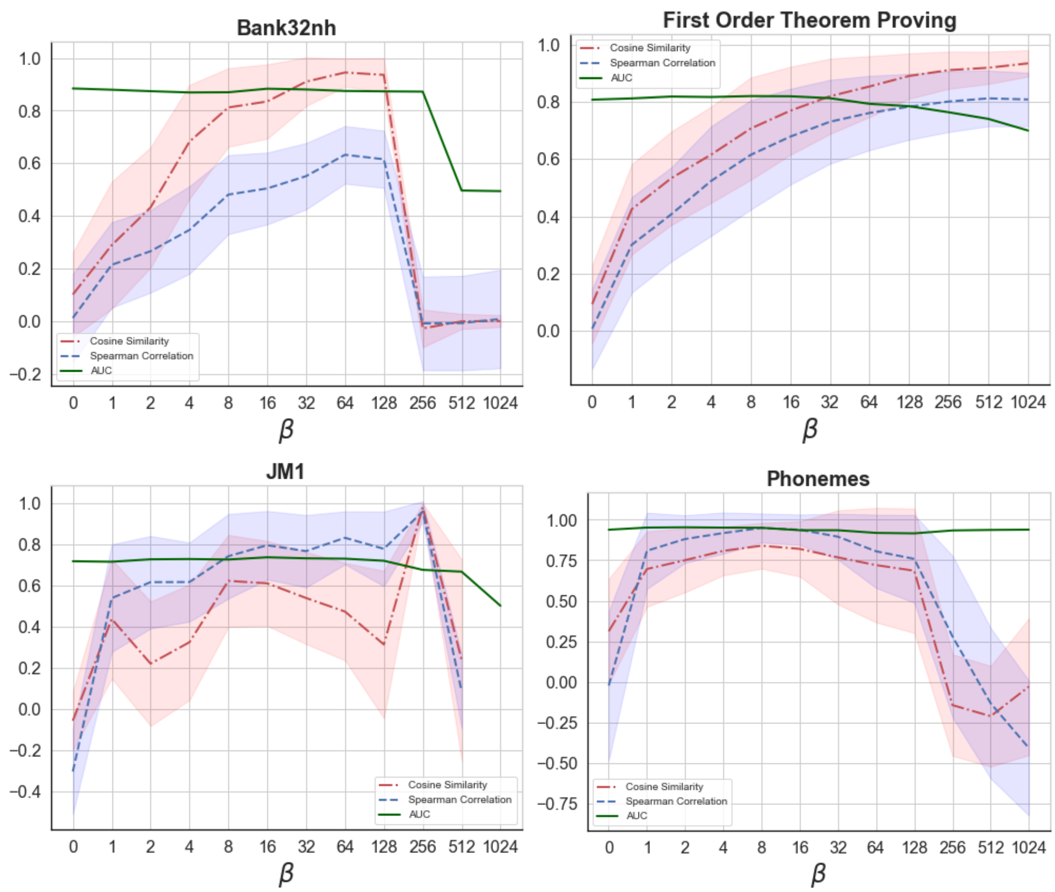

Figure 9: The effect of different values of β on the predictive performance (AUC), alignment with the true Shapley values (cosine similarity), and the similarity in the order of features to the ground truth (Spearman rank).

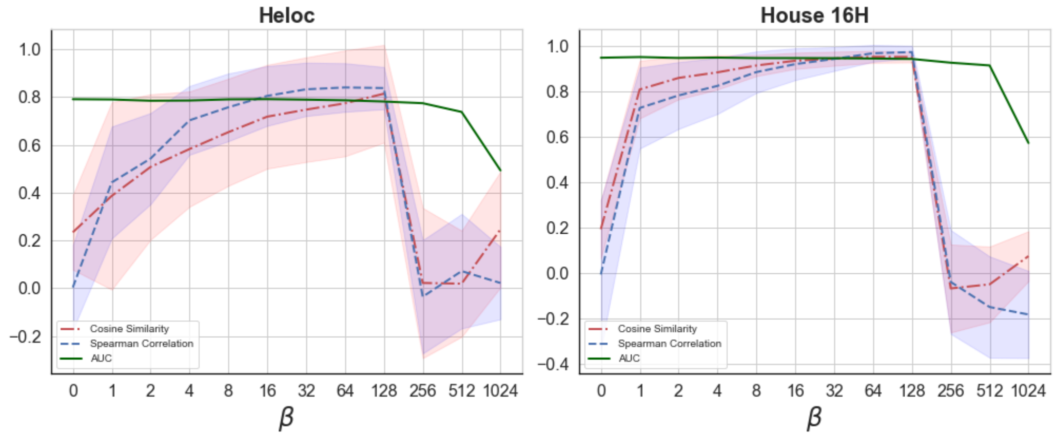

Figure 10: The effect of different values of β on the predictive performance (AUC), alignment with the true Shapley values (cosine similarity), and the similarity in the order of features to the ground truth (Spearman rank).

H.2 The Number of Samples

We assessed the impact of the number of sampled coalitions per data example on the performance of $\mathcal{V}ia^{SHAP}$, retraining the model using the default hyperparameters with the exception of the sample size. We investigated an exponentially increasing range of sample sizes (2^s), from 1 to 128. The findings suggest that the number of samples has a smaller effect on the performance of the trained models compared to β, which allows for effective training of $\mathcal{V}ia^{SHAP}$ models with as few as one sample per data instance. The results are illustrated in Figures 11 and 12.

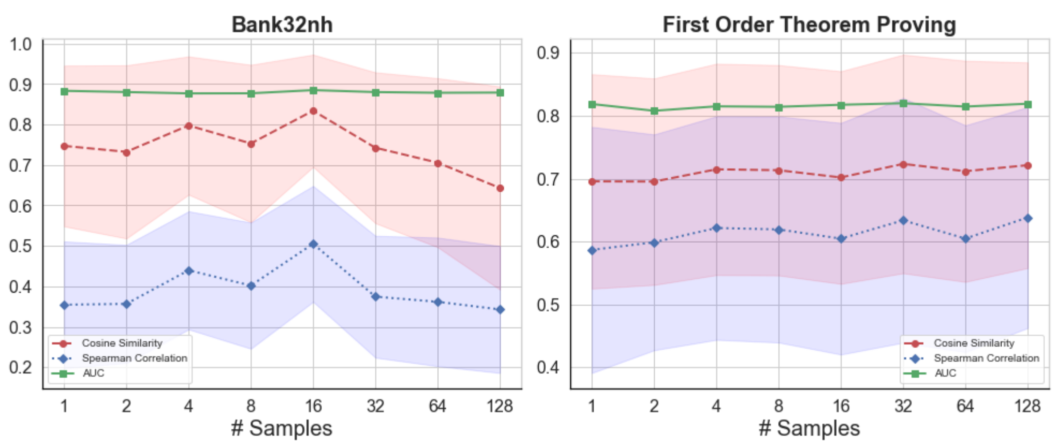

Figure 11: The effect of different number of samples on the predictive performance (AUC), alignment with the true Shapley values (cosine similarity), and the similarity in the order of features to the ground truth (Spearman rank).

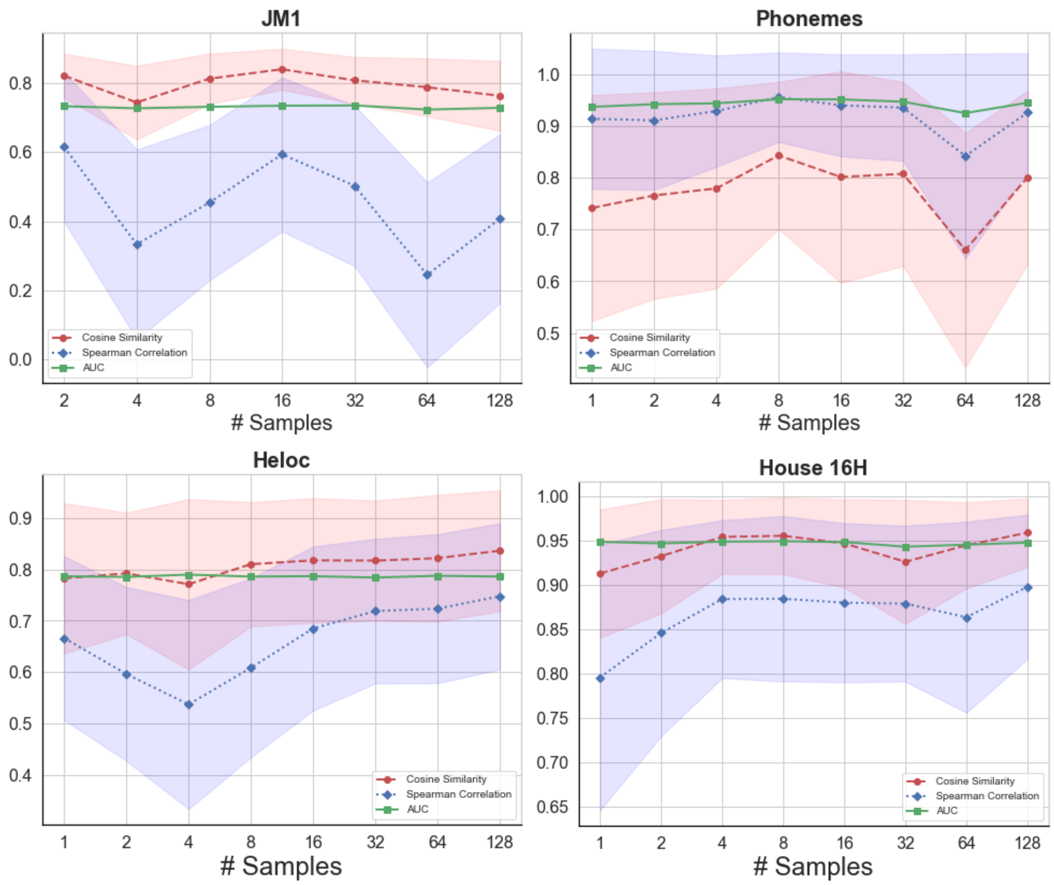

Figure 12: The effect of different number of samples on the predictive performance (AUC), alignment with the true Shapley values (cosine similarity), and the similarity in the order of features to the ground truth (Spearman rank).

H.3 The Effect of Applying a Link Function to the Predicted Outcome

To examine the impact of employing a link function on the predictive performance of $\mathcal{V}ia^{SHAP}$ and the accuracy of its Shapley value approximations, we trained $KAN^{\mathcal{V}ia}$ without applying a link function at the output layer and compared the predictive performance to that of $KAN^{\mathcal{V}ia}$ with the default settings mentioned in the experimental setup. The results of the predictive comparison are summarized in Table 7. To evaluate the null hypothesis that there is no difference in predictive performance, measured by the AUC, between $KAN^{\mathcal{V}ia}$ with and without a link function, the Wilcoxon signed-rank test was employed, given that only two methods were compared. The results indicate that the null hypothesis can be rejected at the 0.05 significance level. Therefore, the results indicate that the presence of a link function does not significantly influence predictive performance in general.

The similarity between the ground truth and the approximated Shapley values by $KAN^{\mathcal{V}ia}$, both with and without link functions, are reported in Table 8. The similarity of $KAN^{\mathcal{V}ia}$'s approximations to the ground truth is measured using the cosine similarity and the Spearman's Rank as described in the experimental setup, which allow for measuring the similarity even if two explanations are not on the same scale, since $\mathcal{V}ia^{SHAP}$ allows for applying a link function to accommodate a valid range of outcomes which can lead $\mathcal{V}ia^{SHAP}$'s approximations to be on a different scale than the ground truth obtained using the unbiased KernelSHAP. However, since we measure the effect of using the link function on the accuracy of Shapley values, we can also apply a metric that measures the similarity on the same scale for models without a link function. Therefore, we also apply R^2 as a similarity metric to the ground truth Shapley values for models without link functions. The results presented in Table 8 demonstrate that $\mathcal{V}ia^{SHAP}$ without a link function significantly outperforms its counterpart with a link function. In order to test the null hypothesis that no difference exists in the accuracy of Shapley value approximations by $KAN^{\mathcal{V}ia}$ with and without a link function, the Wilcoxon signed-rank test was applied. The test results confirm that the null hypothesis can be rejected in both cases, whether Spearman's rank or cosine similarity is used as the similarity metric. Furthermore, the results show that R^2 as a similarity metric is consistent with both Spearman's rank and cosine similarity.

Table 7: The effect of the link function on the predictive performance of $KAN^{\mathcal{V}ia}$ as measured by AUC. The best-performing model is colored in light green.

Dataset	$KAN^{\mathcal{V}ia}$ (without a link function)	$KAN^{\mathcal{V}ia}$ (default settings)
Abalone	**0.883 ± 0.0002**	0.87 ± 0.003
Ada Prior	**0.898 ± 0.003**	0.89 ± 0.005
Adult	**0.919 ± 0.0005**	0.914 ± 0.003
Bank32nh	**0.883 ± 0.003**	0.878 ± 0.001
Electricity	**0.934 ± 0.004**	0.93 ± 0.004
Elevators	**0.936 ± 0.002**	0.935 ± 0.002
Fars	0.958 ± 0.001	**0.96 ± 0.0003**
Helena	0.868 ± 0.006	**0.884 ± 0.0001**
Heloc	**0.792 ± 0.001**	0.788 ± 0.002
Higgs	0.801 ± 0.001	0.801 ± 0.001
hls4ml lhc jets hlf	0.939 ± 0.0005	**0.944 ± 0.0001**
House 16H	0.949 ± 0.001	0.949 ± 0.0007
Indian Pines	0.982 ± 0.001	**0.985 ± 0.0004**
Jannis	0.861 ± 0.001	**0.864 ± 0.001**
JM1	0.686 ± 0.024	**0.732 ± 0.003**
Magic Telescope	0.921 ± 0.002	**0.929 ± 0.001**
MC1	**0.952 ± 0.011**	0.94 ± 0.003
Microaggregation2	0.764 ± 0.008	**0.783 ± 0.002**
Mozilla4	0.965 ± 0.001	**0.968 ± 0.0008**
Satellite	0.944 ± 0.01	**0.996 ± 0.001**
PC2	0.659 ± 0.06	**0.827 ± 0.009**
Phonemes	0.923 ± 0.003	**0.946 ± 0.003**
Pollen	0.501 ± 0.002	**0.515 ± 0.006**
Telco Customer Churn	**0.857 ± 0.003**	0.854 ± 0.003
1st order theorem proving	0.810 ± 0.006	**0.822 ± 0.002**

H.4 The Progress of Training and Validation Losses

In this subsection, we report the progression of training and validation losses with different values of the hyperparameter β using six datasets. A common trend observed across models trained on the six datasets is that, with different values of β, the Shapley loss (scaled by β) consistently decreases quickly below the level of the classification loss, except for the First Order Theorem Proving dataset (Figure 14), which is a multinomial classification dataset. For the First Order Theorem Proving dataset, the Shapley loss remains at a scale determined by the β factor throughout the training time. However, the model for the First Order Theorem Proving dataset can still learn a function that estimates Shapley values with good accuracy, as shown in Tables 2 and 3. Moreover, it benefits from larger β values to achieve accurate Shapley value approximations, as illustrated in Figure 9.

Table 8: The effect of the link function on the similarity of the approximated Shapley values by $KAN^{\mathcal{V}ia}$. The best-performing model is colored in light green .

Dataset	ViaSHAP with default settings: Cosine Similarity	ViaSHAP with default settings: Spearman's Rank	ViaSHAP without a link function: Cosine Similarity	ViaSHAP without a link function: Spearman's Rank	ViaSHAP without a link function: R^2
Abalone	0.969 ± 0.017	0.6635 ± 0.234	0.999 ± 0.0008	0.971 ± 0.052	0.999 ± 0.002
Ada Prior	0.935 ± 0.046	0.8763 ± 0.088	0.963 ± 0.037	0.909 ± 0.068	0.9 ± 0.095
Adult	0.931 ± 0.049	0.9594 ± 0.035	0.981 ± 0.03	0.931 ± 0.074	0.948 ± 0.079
Bank32nh	0.779 ± 0.163	0.432 ± 0.151	0.948 ± 0.045	0.648 ± 0.114	0.87 ± 0.142
Electricity	0.970 ± 0.02	0.7983 ± 0.183	0.998 ± 0.004	0.967 ± 0.043	0.992 ± 0.012
Elevators	0.966 ± 0.024	0.9203 ± 0.064	0.997 ± 0.004	0.969 ± 0.026	0.993 ± 0.009
Fars	0.886 ± 0.253	0.347 ± 0.328	0.962 ± 0.036	0.882 ± 0.073	0.895 ± 0.073
Helena	0.856 ± 0.092	0.669 ± 0.152	0.874 ± 0.095	0.702 ± 0.148	0.016 ± 1.307
Heloc	0.844 ± 0.111	0.7409 ± 0.147	0.962 ± 0.036	0.882 ± 0.073	0.895 ± 0.105
Higgs	0.917 ± 0.068	0.674 ± 0.12	0.991 ± 0.006	0.87 ± 0.057	0.977 ± 0.014
LHC Identify Jet	0.971 ± 0.021	0.8575 ± 0.119	0.999 ± 0.002	0.974 ± 0.032	0.998 ± 0.005
House 16H	0.919 ± 0.048	0.8876 ± 0.092	0.988 ± 0.015	0.952 ± 0.044	0.961 ± 0.057
Indian Pines	0.796 ± 0.121	0.6991 ± 0.116	0.683 ± 0.171	0.553 ± 0.18	0.333 ± 0.192
Jannis	0.852 ± 0.141	0.4775 ± 0.131	0.898 ± 0.072	0.624 ± 0.113	0.722 ± 0.183
JM1	0.88 ± 0.044	0.7561 ± 0.202	0.965 ± 0.042	0.916 ± 0.085	0.901 ± 0.094
Magic Telescope	0.922 ± 0.067	0.9 ± 0.098	0.994 ± 0.006	0.959 ± 0.042	0.98 ± 0.02
MC1	0.466 ± 0.268	0.6212 ± 0.157	0.951 ± 0.093	0.881 ± 0.139	0.873 ± 0.332
Microaggregation2	0.938 ± 0.049	0.8756 ± 0.096	0.982 ± 0.021	0.957 ± 0.049	0.929 ± 0.114
Mozilla4	0.953 ± 0.023	0.9423 ± 0.092	0.9998 ± 0.0003	0.967 ± 0.074	0.9996 ± 0.0007
Satellite	0.841 ± 0.116	0.746 ± 0.212	0.976 ± 0.033	0.894 ± 0.102	0.814 ± 0.296
PC2	0.534 ± 0.183	0.7326 ± 0.161	0.956 ± 0.087	0.875 ± 0.127	0.895 ± 0.223
Phonemes	0.811 ± 0.162	0.9407 ± 0.103	0.993 ± 0.013	0.951 ± 0.094	0.975 ± 0.076
Pollen	0.952 ± 0.059	0.372 ± 0.429	0.994 ± 0.013	0.959 ± 0.076	0.929 ± 0.212
Telco Customer Churn	0.81 ± 0.108	0.8476 ± 0.098	0.978 ± 0.025	0.934 ± 0.052	0.939 ± 0.054
1st order theorem proving	0.725 ± 0.179	0.6228 ± 0.188	0.778 ± 0.123	0.66 ± 0.146	0.429 ± 0.479

Additionally, the results indicate that $\mathcal{V}ia^{SHAP}$ generally tends to take longer to converge as β values increase.

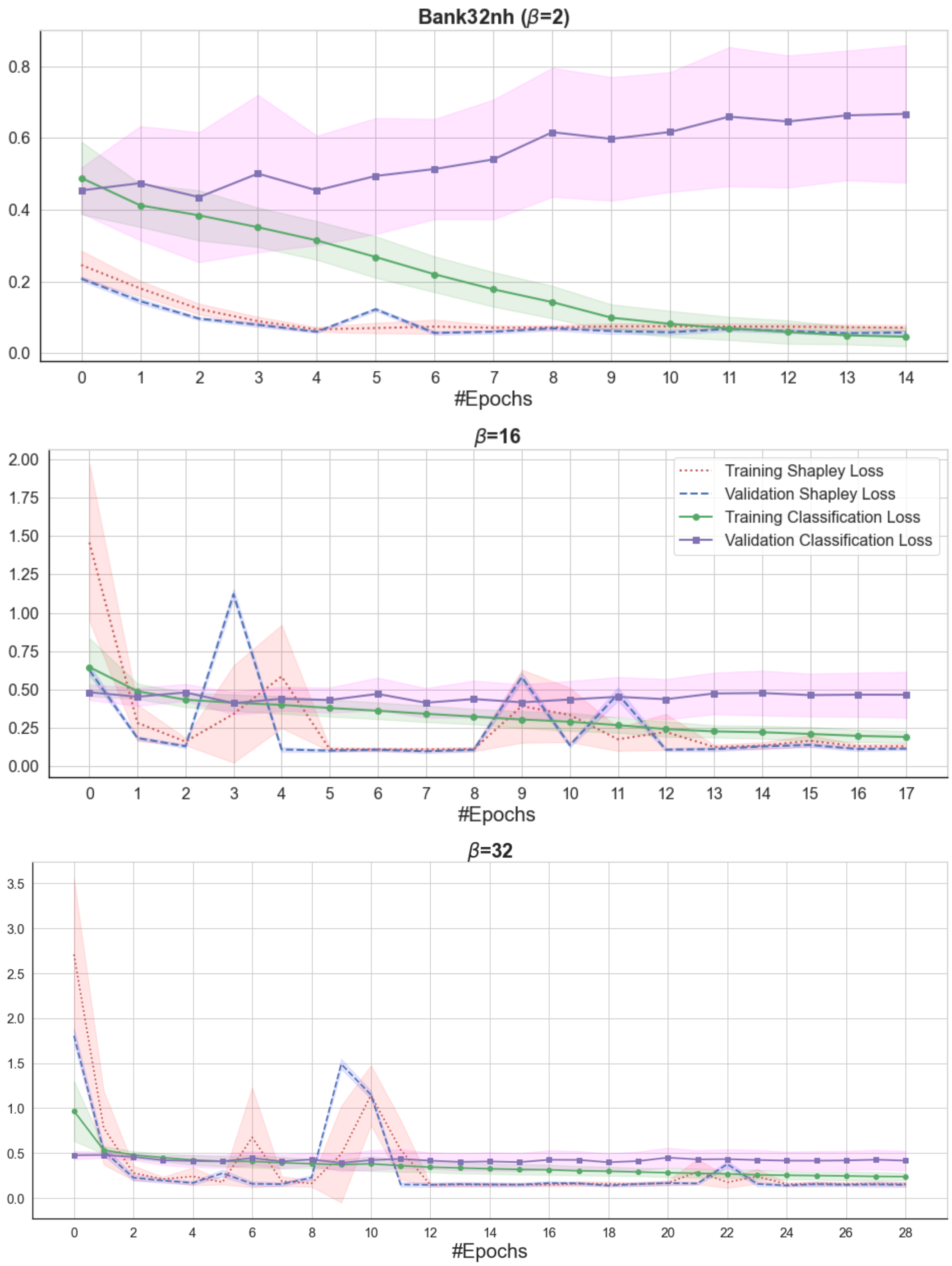

Figure 13: The effect of β value on the progress of the training and the validation loss values.

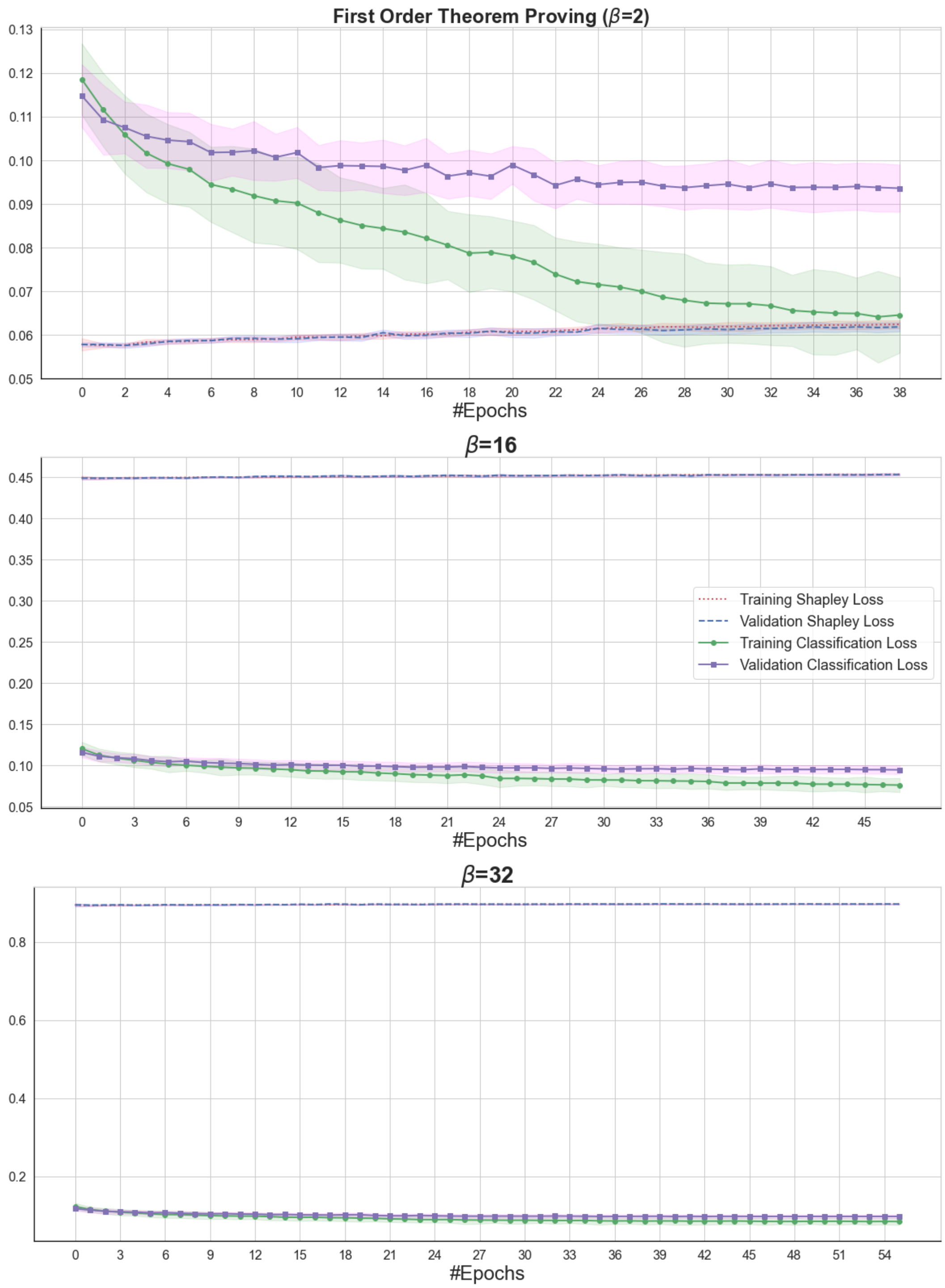

Figure 14: The effect of β value on the progress of the training and the validation loss values.

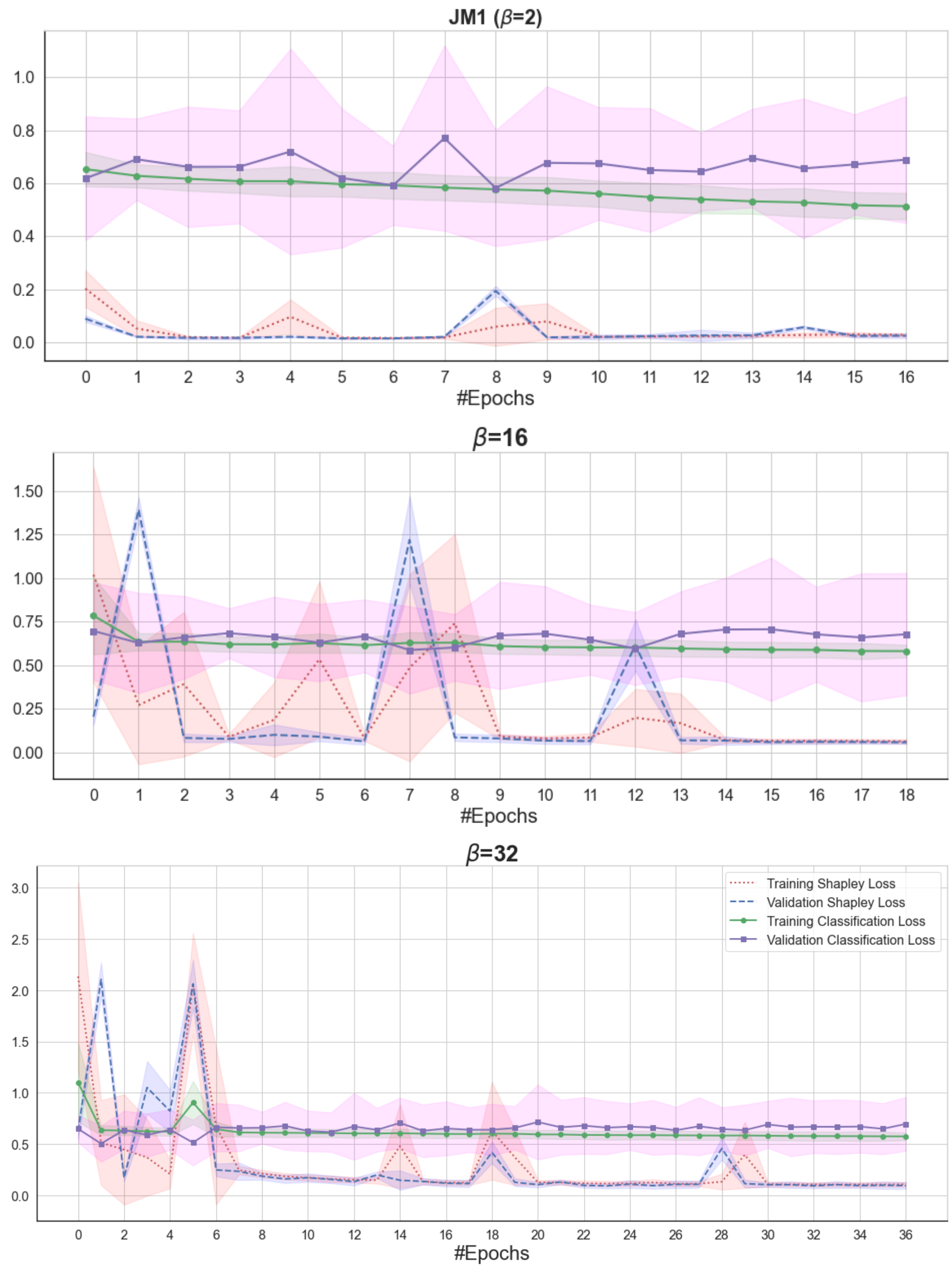

Figure 15: The effect of β value on the progress of the training and the validation loss values.

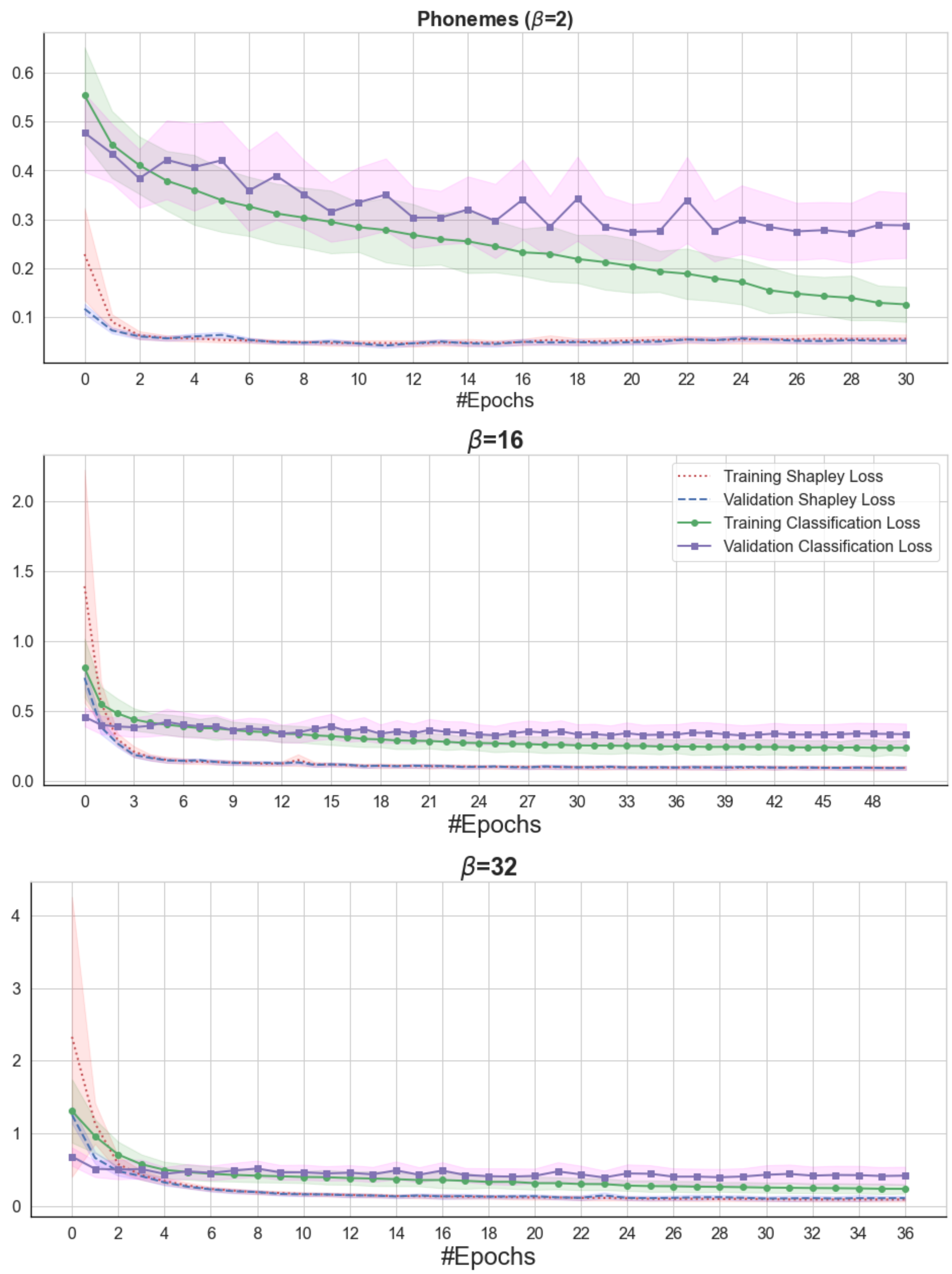

Figure 16: The effect of β value on the progress of the training and the validation loss values.

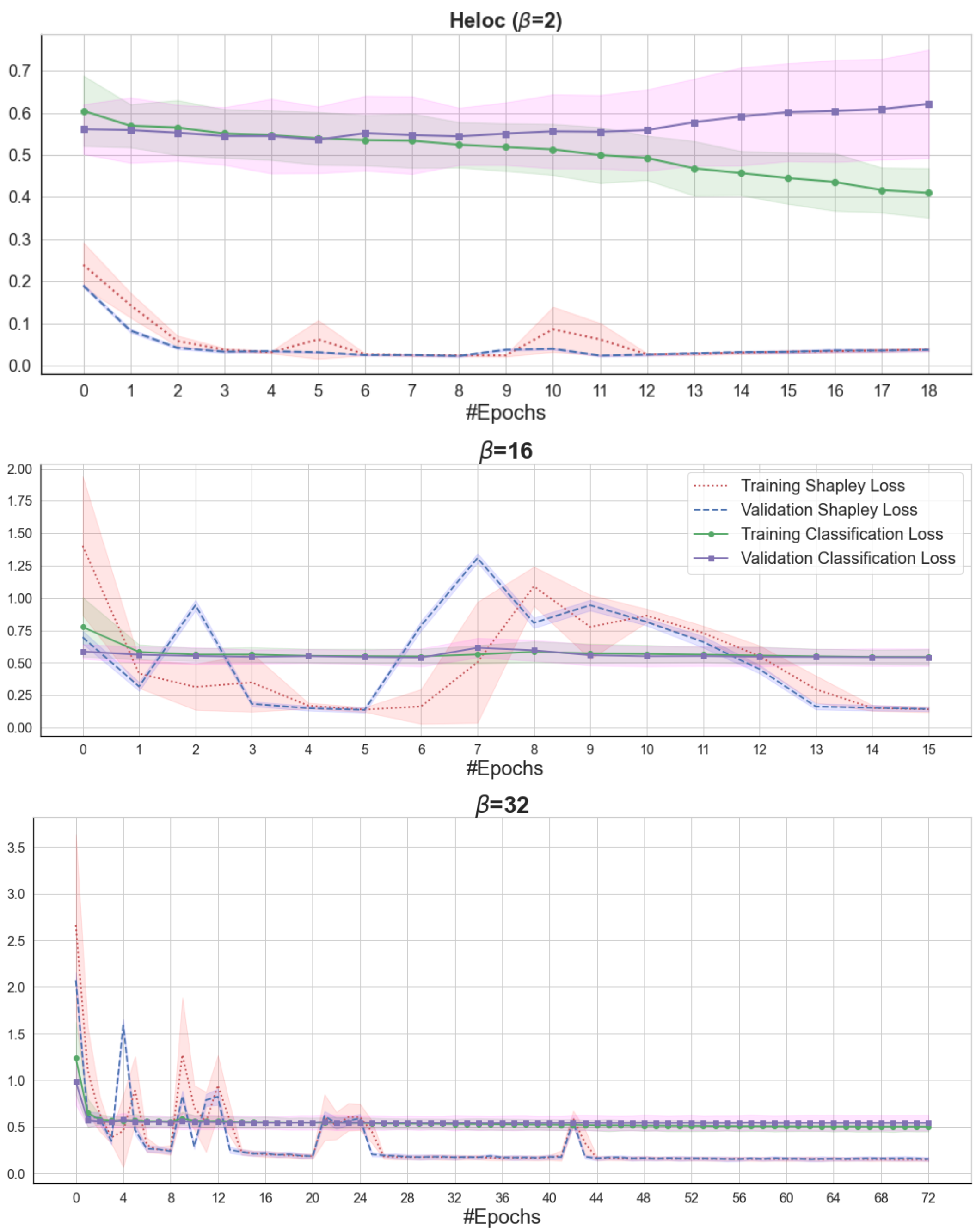

Figure 17: The effect of β value on the progress of the training and the validation loss values.

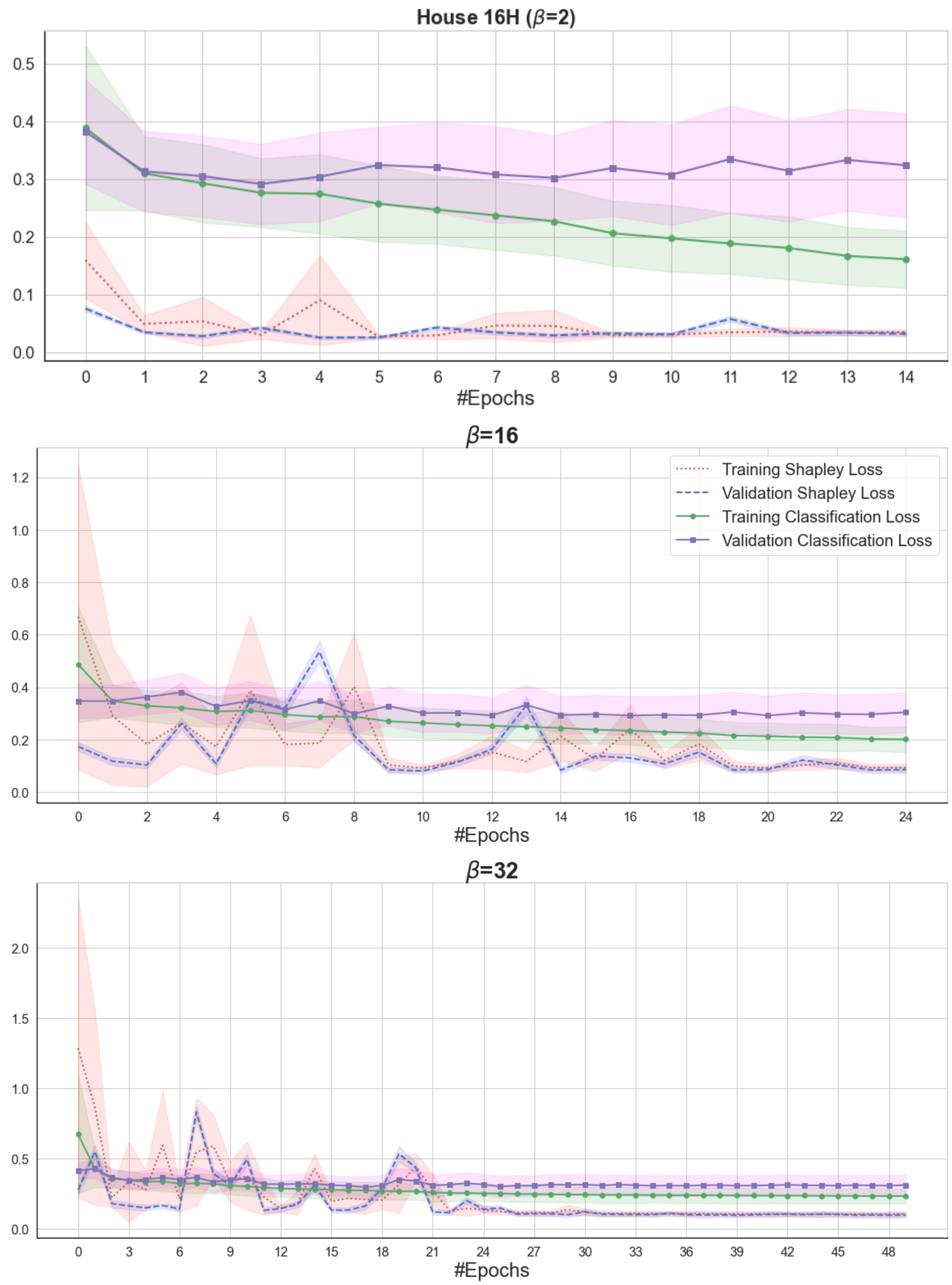

Figure 18: The effect of β value on the progress of the training and the validation loss values.

I A COMPARISON BETWEEN VIASHAP AND FASTSHAP

We compared the accuracy of $\mathcal{V}ia^{SHAP}$'s Shapley value approximations to FastSHAP, using $\mathcal{V}ia^{SHAP}$ as a black-box model within the FastSHAP framework. $\mathcal{V}ia^{SHAP}$ is implemented using $KAN^{\mathcal{V}ia}$ without a link function, while FastSHAP is using the default settings. The evaluation employed metrics such as R^2, cosine similarity, and Spearman's rank correlation to measure the similarity between the computed Shapley values and the ground truth. The results demonstrate that $\mathcal{V}ia^{SHAP}$ achieves significantly higher similarity to the ground truth compared to FastSHAP. This conclusion is supported by the Wilcoxon signed-rank test, which enabled rejection of the null hypothesis that there is no difference in similarity to the ground truth Shapley values between $\mathcal{V}ia^{SHAP}$ and Fast-SHAP. The test confirmed significant differences using all evaluated similarity metrics, including R^2, cosine similarity, and Spearman's rank correlation. The detailed results are available in Table 10.

J A COMPARISON BETWEEN THE INFERENCE TIME OF VIASHAP AND KERNELSHAP

In Table 9, we report the time required to explain 1000 instances using KernelSHAP and ViaSHAP ($KAN^{\mathcal{V}ia}$) on six datasets using an NVIDIA Tesla V100f GPU and 16 cores of an Intel Xeon Gold 6338 processor.

Table 9: The time (in seconds) required to explain 1000 predictions from 6 different datasets using KernelSHAP and ViaSHAP.

Dataset	KernelSHAP	$KAN^{\mathcal{V}ia}$
Adult	56.92	0.0026
Elevators	54.22	0.0021
House 16	53.12	0.0052
Indian Pines	43124.66	0.0023
Microaggregation 2	79.97	0.0022
First order proving theorem	436.25	0.0022

Table 10: A comparison between ViaSHAP and FastSHAP with respect to the similarity of the approximated Shapley values to the ground truth values. The best-performing model is colored in light green.

Dataset	Cosine Similarity		Spearman's Rank		R^2	
	ViaSHAP	FastSHAP	ViaSHAP	FastSHAP	ViaSHAP	FastSHAP
Abalone	0.999 ± 0.0008	0.999 ± 0.002	0.971 ± 0.052	0.966 ± 0.055	0.999 ± 0.002	0.996 ± 0.008
Ada Prior	0.963 ± 0.037	0.703 ± 0.25	0.909 ± 0.068	0.64 ± 0.24	0.887 ± 0.105	0.042 ± 1.359
Adult	0.981 ± 0.03	0.956 ± 0.072	0.931 ± 0.074	0.893 ± 0.115	0.952 ± 0.072	0.853 ± 0.298
Bank32nh	0.948 ± 0.045	0.897 ± 0.079	0.648 ± 0.114	0.527 ± 0.133	0.852 ± 0.161	0.728 ± 0.29
Electricity	0.998 ± 0.004	0.978 ± 0.06	0.967 ± 0.043	0.921 ± 0.101	0.993 ± 0.011	0.914 ± 0.306
Elevators	0.997 ± 0.004	0.994 ± 0.006	0.969 ± 0.026	0.941 ± 0.047	0.993 ± 0.009	0.983 ± 0.023
Fars	0.997 ± 0.008	0.997 ± 0.021	0.849 ± 0.098	0.834 ± 0.124	0.994 ± 0.022	0.991 ± 0.028
Helena	0.874 ± 0.095	0.822 ± 0.139	0.702 ± 0.148	0.6 ± 0.193	0.677 ± 0.204	0.532 ± 0.29
Heloc	0.962 ± 0.036	0.935 ± 0.064	0.882 ± 0.073	0.826 ± 0.111	0.894 ± 0.098	0.824 ± 0.177
Higgs	0.991 ± 0.006	0.994 ± 0.004	0.87 ± 0.057	0.899 ± 0.049	0.977 ± 0.014	0.986 ± 0.01
hls4ml lhc jets hlf	0.999 ± 0.002	0.999 ± 0.003	0.974 ± 0.032	0.971 ± 0.035	0.998 ± 0.005	0.997 ± 0.016
House 16H	0.988 ± 0.015	0.964 ± 0.035	0.952 ± 0.044	0.891 ± 0.09	0.964 ± 0.039	0.89 ± 0.107
Indian Pines	0.683 ± 0.171	0.423 ± 0.154	0.553 ± 0.18	0.204 ± 0.122	0.333 ± 0.192	-0.615 ± 0.912
Jannis	0.898 ± 0.072	0.92 ± 0.064	0.624 ± 0.113	0.673 ± 0.106	0.722 ± 0.183	0.776 ± 0.179
JM1	0.965 ± 0.042	0.98 ± 0.042	0.916 ± 0.085	0.934 ± 0.083	0.887 ± 0.206	0.925 ± 0.37
Magic Telescope	0.994 ± 0.006	0.984 ± 0.023	0.959 ± 0.042	0.918 ± 0.084	0.98 ± 0.021	0.946 ± 0.094
MC1	0.951 ± 0.093	0.789 ± 0.254	0.881 ± 0.139	0.638 ± 0.297	0.881 ± 0.346	-0.024 ± 9.964
Microaggregation2	0.982 ± 0.021	0.99 ± 0.017	0.957 ± 0.049	0.97 ± 0.041	0.944 ± 0.061	0.966 ± 0.054
Mozilla4	0.9998 ± 0.0003	0.994 ± 0.017	0.967 ± 0.074	0.921 ± 0.141	0.9996 ± 0.0007	0.984 ± 0.049
Satellite	0.976 ± 0.033	0.858 ± 0.114	0.894 ± 0.102	0.55 ± 0.25	0.873 ± 0.151	0.126 ± 0.793
PC2	0.956 ± 0.087	0.786 ± 0.234	0.875 ± 0.127	0.619 ± 0.249	0.891 ± 0.272	0.274 ± 1.616
Phonemes	0.993 ± 0.013	0.981 ± 0.036	0.951 ± 0.094	0.946 ± 0.096	0.971 ± 0.071	0.925 ± 0.165
Pollen	0.994 ± 0.013	0.984 ± 0.024	0.959 ± 0.076	0.905 ± 0.129	0.933 ± 0.276	0.855 ± 0.23
Telco Customer Churn	0.978 ± 0.025	0.963 ± 0.045	0.934 ± 0.052	0.892 ± 0.087	0.924 ± 0.085	0.899 ± 0.109
1st order theorem proving	0.778 ± 0.123	0.776 ± 0.174	0.66 ± 0.146	0.658 ± 0.206	0.429 ± 0.479	0.367 ± 2.832

K Computational Cost

The experiments were conducted using an NVIDIA Tesla V100f GPU and 16 cores of an Intel Xeon Gold 6338 processor. The training time required for both $KAN^{\mathcal{V}ia}$ and $MLP^{\mathcal{V}ia}$ are recorded on 1,000 data examples with varying numbers of coalitions (Table 11). The inference time is also recorded on 1,000 data example for both $KAN^{\mathcal{V}ia}$ and $MLP^{\mathcal{V}ia}$ as shown in Table 12. All the results are reported as the mean and standard deviation across five different runs. Generally, $MLP^{\mathcal{V}ia}$ is faster than $KAN^{\mathcal{V}ia}$ in both training and inference. Additionally, while the number of samples per data example increased exponentially, the computational cost during training did not rise at the same rate, as depicted in Figure 19.

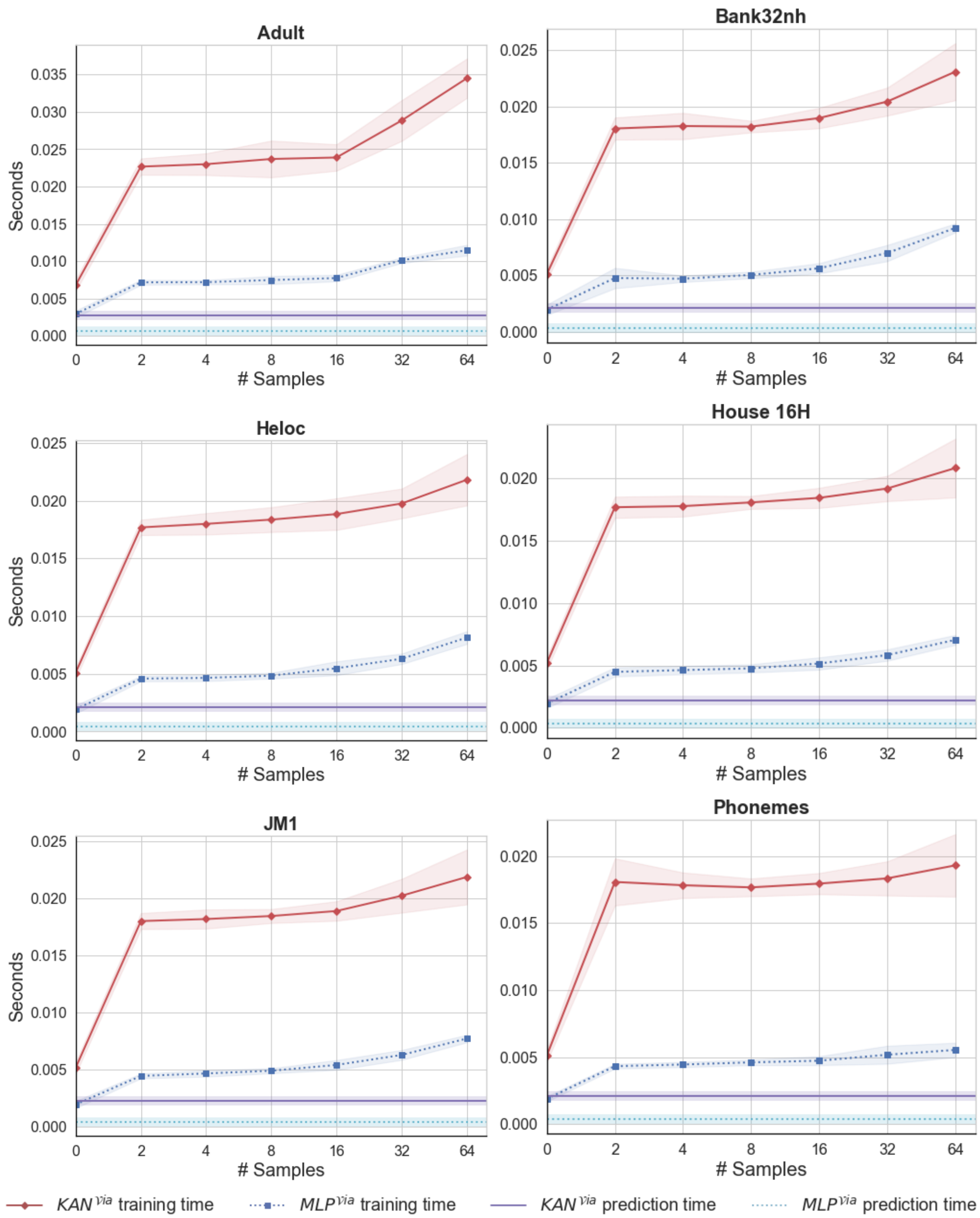

Figure 19: The training time and prediction time on 1000 data instance of $KAN^{\mathcal{V}ia}$ and $MLP^{\mathcal{V}ia}$.

Table 11: The training time in seconds for 1000 data instances using $KAN^{\mathcal{V}ia}$ and $MLP^{\mathcal{V}ia}$.

Dataset	$KAN^{\mathcal{V}ia}$				$MLP^{\mathcal{V}ia}$			
	No Sampling	2 Samples	16 Samples	32 Samples	No Sampling	2 Samples	16 Samples	32 Samples
Abalone	0.0058 ± 0.0005	0.0208 ± 0.0013	0.0209 ± 0.0009	0.0228 ± 0.0016	0.002 ± 0.0002	0.0053 ± 0.0007	0.0056 ± 0.0005	0.0062 ± 0.0006
Ada Prior	0.0068 ± 0.0005	0.024 ± 0.0014	0.0235 ± 0.0017	0.0284 ± 0.0021	0.0028 ± 0.0003	0.0071 ± 0.0003	0.0079 ± 0.0005	0.0104 ± 0.0007
Adult	0.0068 ± 0.0008	0.0227 ± 0.0011	0.0239 ± 0.0018	0.0288 ± 0.0027	0.003 ± 0.0005	0.0072 ± 0.0003	0.0078 ± 0.0004	0.0101 ± 0.0004
Bank32nh	0.0051 ± 0.0005	0.018 ± 0.001	0.019 ± 0.0009	0.0204 ± 0.0012	0.002 ± 0.0005	0.0048 ± 0.0009	0.0056 ± 0.0005	0.007 ± 0.0007
Electricity	0.0059 ± 0.0005	0.0209 ± 0.0012	0.0211 ± 0.0013	0.023 ± 0.0017	0.0021 ± 0.0004	0.0051 ± 0.0004	0.0056 ± 0.0005	0.0061 ± 0.0005
Elevators	0.0051 ± 0.0005	0.0177 ± 0.0007	0.0187 ± 0.0013	0.0193 ± 0.0011	0.0019 ± 0.0004	0.0044 ± 0.0003	0.0052 ± 0.0005	0.0059 ± 0.0004
Fars	0.009 ± 0.0006	0.0262 ± 0.0009	0.0274 ± 0.0017	0.0358 ± 0.0017	0.0064 ± 0.0004	0.0102 ± 0.0004	0.0114 ± 0.0005	0.0153 ± 0.0005
Helena	0.0054 ± 0.0007	0.0184 ± 0.0012	0.0195 ± 0.0014	0.0208 ± 0.002	0.0021 ± 0.0005	0.0048 ± 0.0004	0.0061 ± 0.0008	0.0073 ± 0.0008
Heloc	0.0051 ± 0.0005	0.0177 ± 0.0007	0.0188 ± 0.0014	0.0198 ± 0.0013	0.0019 ± 0.0004	0.0046 ± 0.0002	0.0055 ± 0.0006	0.0063 ± 0.0004
Higgs	0.0067 ± 0.0003	0.019 ± 0.0005	0.0198 ± 0.0006	0.0211 ± 0.0013	0.0021 ± 0.0002	0.0062 ± 0.0004	0.0067 ± 0.0004	0.0075 ± 0.0005
LHC Identify Jet	0.0055 ± 0.0003	0.0184 ± 0.0007	0.0187 ± 0.0008	0.0198 ± 0.0012	0.0022 ± 0.0003	0.005 ± 0.0002	0.0058 ± 0.0004	0.0064 ± 0.0003
House 16H	0.0052 ± 0.0006	0.0176 ± 0.0008	0.0238 ± 0.0075	0.0195 ± 0.0018	0.002 ± 0.0004	0.0045 ± 0.0004	0.0052 ± 0.0005	0.0058 ± 0.0005
Indian Pines	0.0054 ± 0.0006	0.0194 ± 0.0011	0.0268 ± 0.0022	0.0355 ± 0.0026	0.0021 ± 0.0004	0.0058 ± 0.0004	0.0129 ± 0.0006	0.0208 ± 0.0009
Jannis	0.0073 ± 0.0006	0.0195 ± 0.0006	0.0214 ± 0.001	0.0235 ± 0.0016	0.0022 ± 0.0003	0.0053 ± 0.0002	0.0073 ± 0.0005	0.0096 ± 0.0002
JM1	0.0051 ± 0.0005	0.018 ± 0.0007	0.0189 ± 0.0009	0.0202 ± 0.0015	0.0019 ± 0.0003	0.0044 ± 0.0002	0.0054 ± 0.0004	0.0063 ± 0.0004
MagicTelescope	0.0051 ± 0.0005	0.0178 ± 0.0009	0.0183 ± 0.001	0.0188 ± 0.0012	0.0019 ± 0.0002	0.0046 ± 0.0007	0.005 ± 0.0005	0.0054 ± 0.0005
MC1	0.0051 ± 0.0005	0.0182 ± 0.0008	0.0194 ± 0.0008	0.0209 ± 0.0014	0.0021 ± 0.0007	0.0045 ± 0.0003	0.0063 ± 0.0004	0.0076 ± 0.0007
Microaggregation 2	0.0053 ± 0.0007	0.0187 ± 0.0011	0.0189 ± 0.001	0.0198 ± 0.0013	0.0019 ± 0.0003	0.0045 ± 0.0002	0.0055 ± 0.0004	0.0062 ± 0.0004
Mozilla4	0.0051 ± 0.0006	0.0178 ± 0.0007	0.0184 ± 0.0015	0.0188 ± 0.0016	0.0019 ± 0.0003	0.0044 ± 0.0003	0.0048 ± 0.0003	0.005 ± 0.0004
Satellite	0.0051 ± 0.0005	0.018 ± 0.001	0.0197 ± 0.002	0.0207 ± 0.0015	0.002 ± 0.0004	0.0046 ± 0.0003	0.006 ± 0.0007	0.0073 ± 0.0005
PC2	0.005 ± 0.0004	0.0178 ± 0.0007	0.0192 ± 0.0009	0.0209 ± 0.0019	0.0019 ± 0.0003	0.0045 ± 0.0002	0.0059 ± 0.0005	0.0073 ± 0.0006
Phonemes	0.0051 ± 0.0005	0.0181 ± 0.0018	0.0179 ± 0.0008	0.0183 ± 0.0013	0.0019 ± 0.0002	0.0043 ± 0.0002	0.0047 ± 0.0003	0.0052 ± 0.0007
Pollen	0.005 ± 0.0004	0.018 ± 0.0015	0.0181 ± 0.0013	0.0185 ± 0.0015	0.0019 ± 0.0004	0.0044 ± 0.0003	0.0048 ± 0.0004	0.005 ± 0.0004
Telco Customer Churn	0.0076 ± 0.0008	0.0247 ± 0.0014	0.0256 ± 0.0016	0.0338 ± 0.002	0.0038 ± 0.0005	0.0093 ± 0.0004	0.0103 ± 0.0006	0.0138 ± 0.0004
1st Ord. Theorem Prov.	0.0052 ± 0.0006	0.0186 ± 0.0013	0.0294 ± 0.0197	0.0224 ± 0.002	0.0019 ± 0.0003	0.0048 ± 0.0003	0.0065 ± 0.0003	0.0087 ± 0.0005

Table 12: The prediction running time in seconds for 1000 data instances using $KAN^{\mathcal{V}ia}$ and $MLP^{\mathcal{V}ia}$.

Dataset	$KAN^{\mathcal{V}ia}$	$MLP^{\mathcal{V}ia}$
Abalone	0.0024 ± 0.0003	0.0004 ± 0.00003
Ada Prior	0.003 ± 0.0008	0.0006 ± 0.000005
Adult	0.0026 ± 0.0004	0.0006 ± 0.000005
Bank32nh	0.0021 ± 0.0002	0.0004 ± 0.0001
Electricity	0.0024 ± 0.0003	0.0005 ± 0.0002
Elevators	0.0021 ± 0.0002	0.0005 ± 0.0003
Fars	0.0031 ± 0.0005	0.0009 ± 0.0001
Helena	0.0023 ± 0.0004	0.0004 ± 0.0001
Heloc	0.0022 ± 0.0002	0.0003 ± 0.000005
Higgs	0.0022 ± 0.0002	0.0003 ± ´0.00001
LHC Identify Jet	0.0023 ± 0.0004	0.0004 ± 0.00001
House 16H	0.0052 ± 0.0005	0.0004 ± 0.0001
Indian Pines	0.0023 ± 0.0003	0.0004 ± 0.0001
Jannis	0.0023 ± 0.0003	0.0004 ± 0.00001
JM1	0.0026 ± 0.0012	0.0003 ± 0.00001
MagicTelescope	0.0022 ± 0.0002	0.0003 ± 0.00001
MC1	0.0023 ± 0.0003	0.0004 ± 0.0001
Microaggregation 2	0.0022 ± 0.0002	0.0004 ± 0.00001
Mozilla 4	0.0022 ± 0.0002	0.0004 ± 0.0001
Satellite	0.0022 ± 0.0003	0.0004 ± 0.0001
PC2	0.0021 ± 0.0003	0.0003 ± 0.00001
Phonemes	0.0021 ± 0.0001	0.0003 ± 0.000005
Pollen	0.0022 ± 0.0003	0.0004 ± 0.0001
Telco Customer Churn	0.003 ± 0.0005	0.0009 ± 0.0001
1st Order Theorem Proving	0.0022 ± 0.0003	0.0004 ± 0.000004

L DATASET DETAILS

Table 13 presents an overview of the datasets used in the experiments. The table includes the number of classes, number of features, dataset size, training, validation, and test split sizes. Additionally, the table provides the corresponding dataset ID from OpenML.

Table 13: The dataset information.

Dataset	# Features	# Classes	Dataset Size	Train. Set	Val. Set	Test Set	OpenML ID
Abalone	8	2	4177	2506	836	835	720
Ada Prior	14	2	4562	2737	913	912	1037
Adult	14	2	48842	43957	2443	2442	1590
Bank32nh	32	2	8,192	5,734	1,229	1,229	833
Electricity	8	2	45,312	36,249	4,532	4,531	151
Elevators	18	2	16,599	11,619	2,490	2,490	846
Fars	29	8	100,968	80,774	10,097	10,097	40672
Helena	27	100	65,196	41,724	10,432	13,040	41169
Heloc	22	2	10,000	7,500	1,250	1,250	45023
Higgs	28	2	98,050	88,245	4,903	4,902	23512
LHC Identify Jet	16	5	830,000	749,075	39,425	41,500	42468
House 16H	16	2	22,784	18,227	2,279	2,278	821
Indian Pines	220	8	9,144	5,852	1,463	1,829	41972
Jannis	54	4	83,733	53,588	13,398	16,747	41168
JM1	21	2	10,885	8,708	1,089	1,088	1053
MagicTelescope	10	2	19,020	15,216	1,902	1,902	1120
MC1	38	2	9,466	7,478	994	994	1056
Microaggregation 2	20	5	20,000	12,800	3,200	4,000	41671
Mozilla 4	5	2	15,545	12,436	1,555	1,554	1046
Satellite	36	2	5,100	2,805	1,148	1,147	40900
PC2	36	2	5,589	3,353	1,118	1,118	1069
Phonemes	5	2	5,404	3,782	811	811	1489
Pollen	5	2	3,848	2,308	770	770	871
Telco Customer Churn	19	2	7,043	4,930	1,057	1,056	42178
1st Order Theorem Proving	51	6	6,118	3,915	979	1,224	1475