Interpretable Mesomorphic Neural Networks For Tabular Data

Arlind Kadra
Department of Representation Learning
University of Freiburg
kadraa@cs.uni-freiburg.de

Sebastian Pineda Arango
Department of Representation Learning
University of Freiburg
pineda@cs.uni-freiburg.de

Josif Grabocka
Department of Machine Learning
University of Technology Nuremberg
josif.grabocka@utn.de

Abstract

Even though neural networks have been long deployed in applications involving tabular data, still existing neural architectures are not explainable by design. In this work, we propose a new class of interpretable neural networks for tabular data that are both deep and linear at the same time (i.e. mesomorphic). We optimize deep hypernetworks to generate explainable linear models on a per-instance basis. As a result, our models retain the accuracy of black-box deep networks while offering *free lunch* explainability for tabular data by design. Through extensive experiments, we demonstrate that our explainable deep networks have comparable performance to state-of-the-art classifiers on tabular data and outperform current existing methods that are explainable by design.

1 Introduction

Tabular data are arguably the most widely spread traditional data modality arising in a plethora of real-world application domains (Bischl et al., 2021; Borisov et al., 2022). There exists a recent trend to deploy neural networks for predictive tasks on tabular data (Kadra et al., 2021; Gorishniy et al., 2021; Somepalli et al., 2022; Hollmann et al., 2023). In a series of such application realms, it is important to be able to explain the predictions of deep learning models to humans (Ras et al., 2022), especially when interacting with human decision-makers, such as in healthcare (Gulum et al., 2021; Tjoa & Guan, 2021), or the financial sector (Sadhwani et al., 2020). Heavily parametrized models such as deep neural networks can fit complex interactions in tabular datasets and achieve high predictive accuracy, however, they are not explainable. In that context, achieving both high predictive accuracy and explainability remains an open research question for the Machine Learning community.

In this work, we introduce *mesomorphic* neural architectures[1], a new class of deep models that are **both deep and locally linear** at the same time, therefore, offering interpretability by design. In a nutshell, we propose a new architecture that is simultaneously (i) *deep and accurate*, as well as (ii) *linear and explainable* on a per-instance basis. Technically speaking, we learn *deep* hypernetworks that generate *linear* models that are accurate concerning the data point we are interested in explaining.

Our interpretable mesomorphic networks for tabular data (dubbed IMN) are classification or regression models that identify the relevant tabular features by design. It is important to highlight that this work

[1]The etymology of the term *mesomorphic* is inspired by Chemistry as "pertaining to an intermediate phase of matter". For instance, a liquid crystal qualifies as both solid and liquid.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

tackles explaining predictions for a single data point (Lundberg & Lee, 2017), instead of explaining a model globally for the whole dataset (Ras et al., 2022). Similarly to existing prior works (Alvarez-Melis & Jaakkola, 2018; Chen et al., 2018), we train deep models that generate explainable local models for a data sample of interest. In contrast, we train hypernetworks that generate linear models in the original feature space through a purely supervised end-to-end optimization.

We empirically show that the proposed explainable deep models are both as accurate as existing black-box classifiers for tabular datasets and achieve better performance compared to explainable end-to-end prior methods. At the same time, IMN is as interpretable as explainer techniques. Throughout this work, explainers can be categorized into two groups: *i)* interpretable surrogate models that are trained to approximate black-box models (Lundberg & Lee, 2017), and *ii)* end-to-end explainable methods by design. Concretely, we show that our method achieves comparable accuracy to competitive black-box classifiers and manages to outperform current state-of-the-art end-to-end explainable methods on the tabular datasets of the popular AutoML benchmark (Gijsbers et al., 2019). In addition, we compare our technique against state-of-the-art predictive explainers on the recent XAI explainability benchmark for tabular data (Liu et al., 2021) and empirically demonstrate that our method offers competitive interpretability. As a result, our method represents a significant step forward in making deep learning explainable by design for tabular datasets. Overall, this work offers the following contributions:

- We present a technique that makes deep learning explainable by design via training hypernetworks to generate instance-specific linear models.
- We offer ample empirical evidence that our method is as accurate as black-box classifiers, with the benefit of being as interpretable as state-of-the-art prediction explainers.

2 Proposed Method

2.1 Shallow Interpretability through Deep Hypernetworks

Let us denote a tabular dataset consisting of N instances of M-dimensional features as $X \in \mathbb{R}^{N \times M}$ and the C-dimensional categorical target variable as $Y \in \{1, \dots, C\}^N$. A model with parameters $w \in \mathcal{W}$ estimates the target variable as $f : \mathbb{R}^M \times \mathcal{W} \rightarrow \mathbb{R}^C$ and is optimized by minimizing the empirical risk $\arg\min_{w \in \mathcal{W}} \sum_{n=1}^{N} \mathcal{L}\left(y_n, f(x_n; w)\right)$, where $\mathcal{L} : \{1, \dots, C\} \times \mathbb{R}^C \rightarrow \mathbb{R}_+$ is a loss function. An explainable model f is one whose predictions $\hat{y}_n = f(x_n; w)$ for a data point x_n are interpretable by humans. For instance, linear models and decision trees are commonly accepted to be interpretable by Machine Learning practitioners (Ribeiro et al., 2016; Lundberg & Lee, 2017).

In this work, we rethink shallow interpretable models $f(x_n; w)$ by defining their parameters $w \in \mathcal{W}$ to be the output of deep non-interpretable hypernetworks $w(x_n; \theta) : \mathbb{R}^M \times \Theta \rightarrow \mathcal{W}$, where the parameters of the hypernetwork are $\theta \in \Theta$. We remind the reader that a hypernetwork (a.k.a. meta-network, or "network of networks") is a neural network that generates the parameters of another network (Ha et al., 2017). In this mechanism, we train deep non-interpretable hypernetworks to generate interpretable models f in an end-to-end manner as $\arg\min_{\theta \in \Theta} \sum_{n=1}^{N} \mathcal{L}\left(y_n, f(x_n; w(x_n; \theta))\right)$.

2.2 Interpretable Mesomorphic Networks (IMN)

Our method trains deep Multi-Layer Perceptron (MLP) hypernetworks that generate the parameters of linear models. For the case of multi-class classification, we consider linear models with parameters $w \in \mathbb{R}^{C \times (M+1)}$, denoting one set of weights and bias terms per class, as $f\left(x_n; w\right)_c = e^{z(x_n; w)_c} / \sum_{k=1}^{C} e^{z(x_n; w)_k}$, with $z\left(x_n; w\right)_c = \sum_{m=1}^{M} w_{c,m} x_{n,m} + w_{c,0}$ representing the logit predictions for the c-th class. For the case of regression the linear model is simply $f\left(x_n; w\right) = \sum_{m=1}^{M} w_m x_{n,m} + w_0$ with $w \in \mathbb{R}^{M+1}$.

Let us present our method IMN by starting with the case of multi-class classification following the hypernetwork mechanism explained in Section 2.1. The hypernetwork $w(x_n; \theta) : \mathbb{R}^M \times \Theta \rightarrow \mathbb{R}^{C \times (M+1)}$ with parameters $\theta \in \Theta$ is a function that given a data point $x_n \in \mathbb{R}^M$ generates the predictions as:

$$f\left(x_n; w(x_n;\theta)\right)_c = \frac{e^{z(x_n;w(x_n;\theta))_c}}{\sum_{k=1}^{C} e^{z(x_n;w(x_n;\theta))_k}}, \quad z\left(x_n; w(x_n;\theta)\right)_c = \sum_{m=1}^{M} w\left(x_n;\theta\right)_{c,m} x_{n,m} + w\left(x_n;\theta\right)_{c,0} \tag{1}$$

Instead of training weights w as in a standard linear classification, we use the output of an MLP network as the linear weights $w(x_n;\theta)$. We illustrate the architecture of our mesomorphic network in Figure 1. In the case of regression, our linear model with hypernetworks is $f(x_n; w(x_n;\theta)) = \sum_{m=1}^{M} w\left(x_n;\theta\right)_m x_{n,m} + w\left(x_n;\theta\right)_0$. We highlight that our experimental protocol (Section 4) includes both classification and regression datasets.

Ultimately, we train the optimal parameters of the hypernetwork to minimize the following loss in an end-to-end manner: $\arg\min_{\theta\in\Theta} \sum_{n=1}^{N} \mathcal{L}\left(y_n, f\left(x_n; w(x_n;\theta)\right)\right) + \lambda||w\left(x_n;\theta\right)||_1$.

Our hypernetworks generate interpretable models that are accurate concerning a data point of interest (e.g. "Explain why patient x_n is estimated to have cancer $f(x_n; w(x_n;\theta)) > 0.5$ by analyzing the impact of features using the generated linear weights."). We stress that our novel method IMN does not simply train one linear model per data point, contrary to prior work (Ribeiro et al., 2016). Instead, the hypernetwork learns to generate accurate linear models by a shared network across all data points. As a result, generating the linear weights demands a single forward pass through the hypernetwork, rather than a separate optimization procedure. Furthermore, our method intrinsically learns to generate similar linear hyperplanes for neighboring data instances. The outputted linear models are accurate both in correctly classifying the data point x_n, but also for the other majority of training instances in the neighborhood (see proof-of-concept experiment below). The outcome is a linear model with parameters $w(x_n;\theta)$ that both interprets the prediction, but also serves as an accurate local model for the neighborhood of points.

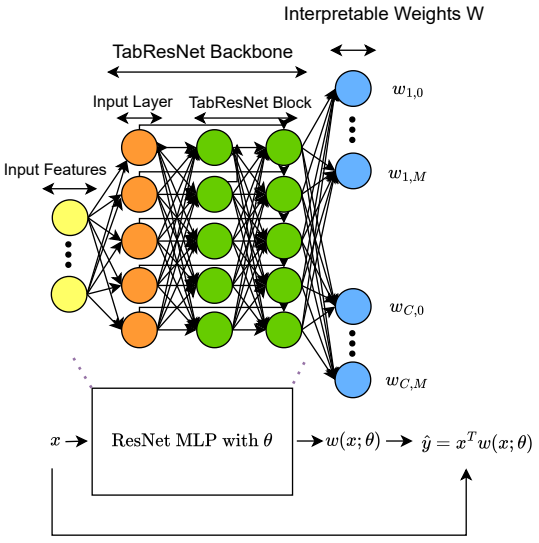

Figure 1: The IMN architecture.

2.3 Explainability Through Feature Attribution

The generated linear models $w\left(x_n;\theta\right)$ can be used to explain predictions through feature attribution (i.e. feature importance) (Liu et al., 2021). It is important to re-emphasize that our method offers interpretable predictions for the estimated target $f(x_n; w\left(x_n;\theta\right))$ of a particular data point x_n. Concretely, we can analyse the linear coefficients $\{w(x_n;\theta)_1, \ldots, w(x_n;\theta)_M\}$ to distill the importances of $\{x_{n,1}, \ldots, x_{n,M}\}$ by measuring the residual impact on the target. The impact of the m-th feature $x_{n,m}$ in estimating the target variable, is proportional to the change in the estimated target if we remove the feature (Hooker et al., 2019). Considering our linear models, the impact of the m-th feature is proportional to the change of the predicted target if we set the m-th feature to zero. In terms of notation, we multiply the feature vector element-wise with a Kronecker delta vector $\delta_i^m = \mathbb{1}_{m\neq i}$.

$$f(x_n; w\left(x_n;\theta\right)) - f(x_n \odot \delta^m; w\left(x_n;\theta\right)) \propto w(x_n;\theta)_m\, x_{n,m} \tag{2}$$

As a result, our feature attribution strategy is that the m-th feature impacts the prediction of the target variable by a signed magnitude of $w(x_n;\theta)_m\, x_{n,m}$. In our experiments, all the features are normalized to the same mean and variance, therefore, the magnitude $w(x_n;\theta)_m\, x_{n,m}$ can be directly used to explain the impact of the m-th feature. In cases where the unsigned importance is required, a practitioner can use the absolute impact $|w(x_n;\theta)_m\, x_{n,m}|$ as the attribution. Furthermore, to measure the global importance of the m-th feature for the whole dataset, we can compute $\frac{1}{N}\sum_{n=1}^{N} |w(x_n;\theta)_m\, x_{n,m}|$.

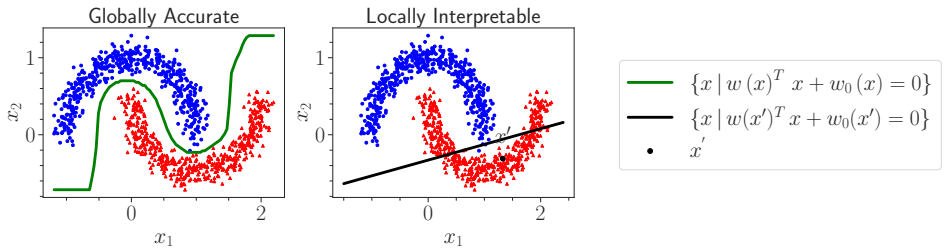

Figure 2: Investigating the accuracy and interpretability of IMN. **Left:** The global decision boundary of our method that separates the classes correctly. **Right:** The local hyperplane pertaining to an example x' which correctly classifies the local example and retains a good global classification for the neighboring points.

2.4 Proof-of-concept: Globally Accurate and Locally Interpretable Classifiers

As a proof of concept, we run our method on the half-moon toy task that consists of a 2-dimensional tabular dataset in the form of two half-moons that are not linearly separable.

Initially, we investigate the global accuracy of our method. As shown in Figure 2 (left), our method correctly classifies all the examples. Furthermore, our method learns an optimal non-linear decision boundary that separates the classes (plotted in green). To determine the decision boundary, we perform a fine-grid prediction on all possible combinations of x_1 and x_2. Subsequently, we identify the points that exhibit the minimal prediction distance to a probability prediction of 0.5. Lastly, in Figure 2 (right) we investigate the local interpretability of our method, by taking a point x' and calculating the corresponding weights $(w(x'), w(x')_0)$ generated by our hypernetwork, where we omited the dependence on θ for simplicity. The black line shows all the points that reside on the hyperplane $w(x')$ as $\{x \mid w(x')^T x + w_0(x') = 0\}$. It is important to highlight that the local hyperplane does not only correctly classify the point x', but also the neighboring points, retaining an accurate linear classifier for the neighborhood of points.

To validate our claim that the per-example (local) hyperplane correctly classifies neighboring points, we conduct the following analysis: For every datapoint x_n we take a specific number of nearest neighbor examples from every class, and we evaluate the classification accuracy of the hyperplane generated for the datapoint x_n on the set of all neighbors. We repeat the above procedure with varying neighborhood sizes and we present the results in Table 1. The results indicate that the mesomorphic neural network generates hyperplanes that are accurate in the neighborhood of the point whose prediction we are interested in explaining.

Table 1: Accuracy of local hyperplanes for neighboring points.

Number of Neighbors	Accuracy
10	0.84
25	0.82
50	0.78
100	0.77
200	0.77

3 Related Work

Interpretable Models by Design: There exist Machine Learning models that offer interpretability by default. A standard approach is to use linear models (Tibshirani, 1996; Efron et al., 2004; Berkson, 1953) that assign interpretable weights to each of the input features. On the other hand, decision trees (Loh, 2011; Craven & Shavlik, 1995) use splitting rules that build up leaves and intermediate nodes. Every leaf node is associated with a predicted label, making it possible to follow the rules that led to a specific prediction. Bayesian methods such as Naive Bayes (Murphy et al., 2006) or Bayesian Neural Networks (Friedman et al., 1997) provide a framework for reasoning on the interactions of prior beliefs with evidence, thus simplifying the interpretation of probabilistic outputs. Instance-based models allow experts to reason about predictions based on the similarity to the train samples. The prediction model aggregates the labels of the neighbors in the training set, using the average of the top-k most similar samples (Freitas, 2014; Kim et al., 2015), or decision functions extracted from prototypes Martens et al. (2007). Attention-based models like TabNet (Arik & Pfister, 2021) make

use of sequential attention to generate feature weights on a per-instance basis, while, DANet (Chen et al., 2022) generates global importance weights for both the raw input features and higher order concepts. Neural additive models (NAMs) (Agarwal et al., 2021) use a neural network per feature to model the additive function of individual features to the output. However, these models trade-off the performance for the sake of interpretability, therefore challenging their usage on applications that need high performance. A prior similar work also trains hyper-networks to generate local models by learning prototype instances through an encoder model Alvarez-Melis & Jaakkola (2018). In contrast, we directly generate interpretable linear models in the original feature space.

Interpretable Model Distillation: Given the common understanding that complex models are not interpretable, prior works propose to learn simple surrogates for mimicking the input-output behavior of the complex models (Burkart & Huber, 2021). Such surrogate models are interpretable, such as linear regression or decision trees (Ribeiro et al., 2016). The local surrogates generate interpretations only valid in the neighborhood of the selected samples. Some approaches explain the output by computing the contribution of each attribute (Lundberg & Lee, 2017) to the prediction of the particular sample. An alternative strategy is to fit globally interpretable models, by relying on decision trees (Frosst & Hinton, 2017; Yang et al., 2018), or linear models (Ribeiro et al., 2016). Moreover, global explainers sometimes provide feature importances (Goldstein et al., 2015; Cortez & Embrechts, 2011), which can be used for auxiliary purposes such as feature engineering. Most of the surrogate models tackle the explainability task disjointly, by first training a black box model, then learning a surrogate in a second step.

Interpretable Deep Learning via Visualization: Given the success of neural networks in real-world applications in computer vision, a series of prior works (Ras et al., 2022) introduce techniques aiming at explaining their predictions. A direct way to measure the feature importance is by evaluating the partial derivative of the network given the input (Simonyan et al., 2013). CAM upscales the output of the last convolutional layers after applying Global Average Pooling (GAP), obtaining a map of the class activations used for interpretability (Zhou et al., 2016). DeepLift calculates pixel-wise relevance scores by computing differences with respect to a reference image (Shrikumar et al., 2017). Integrated Gradients use a baseline image to compute the cumulative sensibility of a black-box model f to pixel-wise changes (Sundararajan et al., 2017). Other methods directly compute the pixel-wise relevance scores such that the network's output equals the sum of scores computed via Taylor Approximations (Montavon et al., 2017).

4 Experimental Protocol

4.1 Predictive Accuracy Experiments

Baselines: In terms of interpretable white-box classifiers, we compare against **Logistic Regression** and **Decision Trees**, based on their scikit-learn library implementations (Pedregosa et al., 2011). On the other hand, we compare against two strong classifiers on tabular datasets, **Random Forest** and **CatBoost**. We use the scikit-learn interface for Random Forest, while for CatBoost we use the official implementation provided by the authors (Prokhorenkova et al., 2018). Lastly, in terms of interpretable deep learning architectures, we compare against **TabNet** (Arik & Pfister, 2021), a transformer architecture that makes use of attention for instance-wise feature-selection and **NAM** (Agarwal et al., 2021), a neural additive model which learns an additive function for every feature. For TabNet we use a well-maintained public implementation [2], while, for NAM we use the official public implementation from the authors [3].

Protocol: We run our predictive accuracy experiments on the AutoML benchmark that includes 35 diverse classification problems, containing between 690 and 539 383 data points, and between 5 and 7 201 features. For more details about the datasets included in our experiments, we point the reader to Appendix C. In our experiments, numerical features are standardized, while we transform categorical features through one-hot encoding. For binary classification datasets we use target encoding, where a category is encoded based on a shrunk estimate of the average target values for the data instances belonging to that category. In the case of missing values, we impute numerical features with zero

[2] `https://github.com/dreamquark-ai/tabnet`

[3] `https://github.com/AmrMKayid/nam`

and categorical features with a new category representing the missing value. For CatBoost and TabNet we do not encode categorical features since the algorithms natively handle them. For all the methods considered we tune the hyperparameters with Optuna (Akiba et al., 2019), a well-known hyperparameter optimization (HPO) library. We use the default HPO algorithm (TPE) from the library and we tune every method for 100 HPO trials or a wall-time limit of 1 day, whichever condition gets fulfilled first. The HPO search spaces of the different baselines were taken from prior work (Gorishniy et al., 2021; Hollmann et al., 2023). For a more detailed description, we kindly refer the reader to Appendix C. Additionally, we use the area under the ROC curve (AUROC) as the evaluation metric. Lastly, the methods that offer GPU support are run on a single NVIDIA RTX2080Ti, while, the rest of the methods are run on an AMD EPYC 7502 32-core processor.

4.2 Explainability Experiments

Baselines: First, we compare against **Random**, a baseline that generates random importance weights. Furthermore, **BreakDown** decomposes predictions into parts that can be attributed to certain features (Staniak & Biecek, 2018). **TabNet** offers instance-wise feature importances by making use of attention. **LIME** is a local interpretability method (Ribeiro et al., 2016) that fits an explainable surrogate (local model) to single instance predictions of black-box models. On the other hand, **L2X** is a method that applies instance-wise feature selection via variational approximations of mutual information (Chen et al., 2018) by making use of a neural network to generate the weights of the explainer. **MAPLE** is a method that uses local linear modeling by exploring random forests as a feature selection method (Plumb et al., 2018). **SHAP** is an additive feature attribution method (Lundberg & Lee, 2017) that allows local interpretation of the data instances. Last but not least, **Kernel SHAP** offers a reformulation of the LIME constrains (Lundberg & Lee, 2017).

Metrics and Benchmark: As explainability evaluation metrics we use faithfulness (Lundberg & Lee, 2017), monotonicity (Luss et al., 2021) (including the ROAR variants (Hooker et al., 2019)), infidelity (Yeh et al., 2019) and Shapley correlation (Lundberg & Lee, 2017). For a detailed description of the metrics, we refer the reader to XAI-Bench, a recent explainability benchmark (Liu et al., 2021).

For our explainability-related experiments, we use all three datasets (Gaussian Linear, Gaussian Non-Linear, and Gaussian Piecewise) available in the XAI-Bench (Liu et al., 2021). For the state-of-the-art explainability baselines, we use the Tabular ResNet (TabResNet) backbone as the model for which the predictions are to be interpreted (same as for IMN). We experiment with different versions of the datasets that feature diverse ρ values, where ρ corresponds to the amount of correlation among features. All datasets have a train/validation set ratio of 10 to 1.

Implementation Details: We use PyTorch as the main library for our implementation. As a backbone, we use a TabResNet where the convolutional layers are replaced with fully-connected layers as suggested by recent work (Kadra et al., 2021). For the default hyperparameters of our method, we use 2 residual blocks and 128 units per layer combined with the GELU activation (Hendrycks & Gimpel, 2016). When training our network, we use snapshot ensembling (Huang et al., 2017) combined with cosine annealing with restarts (Loshchilov & Hutter, 2019). We use a learning rate and weight decay value of 0.01, where, the learning rate is warmed up to 0.01 for the first 5 epochs, a dropout value of 0.25, and an L1 penalty of 0.1 on the weights. Our network is trained for 500 epochs with a batch size of 64. We make our implementation publicly available[4].

5 Experiments and Results

Hypothesis 1: IMN outperforms interpretable white-box models in terms of predictive accuracy.

We compare our method against decision trees and logistic regression, two white-box interpretable models. We run all aforementioned methods on the AutoML benchmark and we measure the predictive performance in terms of AUROC. Lastly, we measure the statistical significance of the results using the autorank package Herbold (2020) that runs a Friedman test with a Nemenyi post-hoc test, and a 0.05 significance level. Figure 3 presents the average rank across datasets based on the AUROC performance. As observed IMN achieves the best rank across the AutoML benchmark

[4]Source code at `https://github.com/ArlindKadra/IMN`

datasets. Furthermore, the difference is statistically significant against both decision trees and logistic regression. The detailed per-dataset results are presented in Appendix C.

Hypothesis 2: The explainability of IMN does not have a statistically significant negative impact on predictive accuracy. Additionally, it achieves a comparable performance against state-of-the-art methods.

This experiment addresses a simple question: *Is our explainable neural network as accurate as a black-box neural network counterpart, that has the same architecture and same capacity?*. Since our hypernetwork is a slight modification of the TabResNet Kadra et al. (2021), we compare it against TabResNet as a classifier. For completeness, we also compare against four other strong baselines, Gradient-Boosted Decision Trees (CatBoost), Random Forest, TabNet, and NAMs. Since the official implementation of NAMs only supports binary classification and regression, we separate the results into: *i*) results over 18 binary classification datasets (Figure 4 **Top**), and *ii*) results over all datasets (Figure 4 **Bottom**).

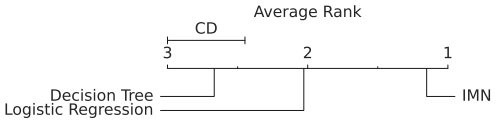

Figure 3: The critical difference diagram for the white-box interpretable methods. A lower rank indicates a better performance over datasets.

The results of Figure 4 demonstrate that IMN achieves a comparable performance to state-of-the-art tabular classification models, while significantly outperforming explainable methods by design. IMN achieves a comparable performance to TabResNet, while outperforming TabNet and NAMs, indicating that its explainability does not harm accuracy in a significant way. There is no statistical significance of the differences between IMN, TabResNet and CatBoost. However, the difference in performance between IMNs, Random Forest, TabNet and NAMs is statistically significant.

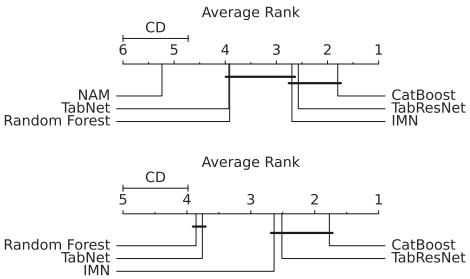

Figure 4: Black-box methods comparison with critical difference diagrams. **Top**: The average rank for the binary datasets present in the benchmark. **Bottom:** The average rank for all datasets present in the benchmark. A lower rank indicates a better performance. Connected ranks via a bold bar indicate that performances are not significantly different ($p > 0.05$).

Additionally, we investigate the runtime performance of the different baselines (NAM is excluded since it cannot be run on the full benchmark). We present the results in Table 2. As expected, deep learning methods take a longer time to train, however, both IMN and TabResNet are the most efficient during inference. We observe that TabResNet takes longer to converge compared to IMN[5], however, both methods demand approximately the same inference time. As a result, the explainability of our method comes as a *free-lunch* benefit. Lastly, IMN is **64x faster in inference** compared to TabNet, an end-to-end deep-learning interpretable method. **Hypothesis 1 and 2 are valid even when default hyperparameters are used**, for more details we kindly refer the reader to Appendix B.

Hypothesis 3: IMNs offer competitive levels of interpretability compared to state-of-the-art explainer techniques.

We compare against 8 explainer baselines in terms of 5 explainability metrics in the 3 datasets of the XAI benchmark (Liu et al., 2021), following the protocol we detailed in Section 4.2. The results of Table 3 demonstrate that IMN is competitive against all explainers across the indicated interpretability metrics. We tie in per-

Table 2: Aggregated training and inference times for all methods.

Method Name	Median Training Time (sec)	Median Inference Time (sec)
IMN (GPU)	192	0.025
TabResNet (GPU)	252	0.020
TabNet (GPU)	237	1.60
CatBoost (GPU)	63.2	0.20
Random Forest	42.55	2.20
Logistic Regression	0.23	0.07
Decision Tree	0.4	0.06

[5]The number of training epochs is a hyperparameter.

Table 3: Investigating the interpretability of IMNs against state-of-the-art interpretability methods. The results are generated from the XAI Benchmark (Liu et al., 2021) datasets (with $\rho = 0$).

Metric	Dataset	Random	Breakd.	Maple	LIME	L2X	SHAP	K. SHAP	TabNet	IMN
Faithfulness (↑)	Gaussian Linear	0.004	0.645	0.980	0.882	0.010	0.974	0.981	0.138	**0.987**
	Gaussian Non-Linear	-0.079	-0.001	0.487	0.796	0.155	0.926	**0.970**	0.161	0.621
	Gaussian Piecewise	0.091	0.634	0.967	0.929	0.016	0.981	**0.990**	0.058	0.841
Faithfulness (ROAR) (↑)	Gaussian Linear	-0.039	0.494	0.548	0.544	0.049	0.549	0.550	0.041	**0.639**
	Gaussian Non-Linear	0.050	0.006	0.040	-0.040	-0.060	-0.010	-0.036	-0.001	0.027
	Gaussian Piecewise	-0.055	0.372	0.347	0.450	0.015	0.409	0.426	0.072	0.404
Infidelity (↓)	Gaussian Linear	0.219	0.041	**0.007**	**0.007**	0.034	**0.007**	**0.007**	0.049	**0.007**
	Gaussian Non-Linear	0.075	0.086	0.021	0.071	0.089	0.030	0.022	0.047	**0.018**
	Gaussian Piecewise	0.132	0.047	0.014	0.019	0.070	0.016	0.019	0.046	**0.008**
Monotonicity (ROAR) (↑)	Gaussian Linear	0.487	0.605	0.700	0.652	0.437	0.680	0.667	0.585	**0.785**
	Gaussian Non-Linear	0.497	0.542	0.645	0.587	0.457	**0.670**	0.632	0.493	0.637
	Gaussian Piecewise	0.485	0.665	0.787	0.427	0.442	0.717	**0.797**	0.542	0.682
Shapley Correlation (↑)	Gaussian Linear	-0.016	0.246	**0.999**	0.942	-0.214	0.993	**0.999**	0.095	**0.999**
	Gaussian Non-Linear	-0.069	-0.179	0.686	0.872	-0.095	0.974	**0.999**	0.125	0.741
	Gaussian Piecewise	-0.078	0.099	0.983	0.959	0.157	0.991	**0.999**	0.070	0.875
Total Wins		1	0	2	2	0	2	**7**	0	**7**

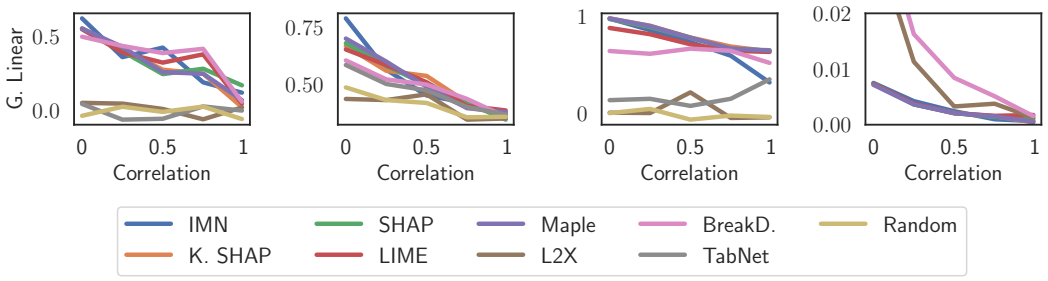

Figure 5: Performance analysis of different interpretability methods over a varying degree of feature correlation ρ. We present the performance of all methods on faithfulness (ROAR), monotonicity (ROAR), faithfulness, and infidelity (ordered from **left** to **right**) on the Gaussian Linear dataset for ρ values ranging from [0, 1].

formance with the second-best method Kernel-SHAP (Lundberg & Lee, 2017) and perform strongly against the other explainers. It is worth highlighting that in comparison to all the explainer techniques, the interpretability of our method comes as *a free-lunch*. In contrast, all the rival methods except TabNet are surrogate interpretable models to black-box models. Moreover, IMN strongly outperforms TabNet, the other baseline that offers explainability by design, achieving both better interpretability (Table 3) and better accuracy (Figure 4).

As a result, for all surrogate interpretable baselines we first need to train a black-box model. Then, for the prediction of **every** data point, we additionally train a local explainer around that point by predicting with the black-box model multiple times. In stark contrast, our method combines prediction models and explainers as an all-in-one neural network. To generate an explainable model for a data point x_n, IMN does not need to train a per-point explainer. Instead, IMN requires only a forward pass through the trained hypernetwork to generate a linear explainer. To quantify the difference in runtime between our method and other interpretable methods we compare the runtimes on a few datasets from the benchmark with a varying number of instances/features such as Credit-g (1000/21), Adult (48842/15), and Christine (5418/1637). Table 4 presents the results, where, as observed IMN has the fastest inference times, being **11-65x faster** compared to TabNet which employs attention, **1710-11400x faster** compared to SHAP that uses the same (TabResNet) backbone, and **455-215850x faster** compared to SHAP that uses CatBoost as a backbone.

Table 4: Interpretable method inference times. All the methods are run on the GPU and the time is reported in seconds.

Method Name	Credit-g	Adult	Christine
IMN	**0.01**	**0.02**	**0.02**
TabNet	0.11	1.30	0.43
SHAP (TabResNet)	17.69	565.11	228.31
SHAP (CatBoost)	4.55	66.89	4317.61

Lastly, we compare all interpretability methods on 4 out of 5 metrics in the presence of a varying ρ factor, which controls the correlation of features on the Gaussian Linear dataset. Figure 5 presents the comparison, where IMN behaves similarly to other interpretable methods and has a comparable performance with the top methods in the majority of metrics. The results agree with the findings of prior work (Liu et al., 2021), where the performance in the interpretability metrics drops in the presence of feature correlations. Although our work focuses on tabular data, in Appendix A we present an application of IMN in the vision domain.

Table 5: The feature rank importances for the Census dataset. A lower rank is associated with a higher feature importance.

Feature	SHAP	Decision Tree	TabNet	CatBoost	IMN
Age	2	5	2	3	3
Capital Gain	9	4	3	1	1
Capital Loss	10	9	14	4	5
Demographic	1	2	9	10	6
Education	5	3	5	9	9
Education num.	4	12	6	6	2
Hours per week	6	7	7	7	4
Race	8	10	5	12	7
Occupation	3	6	8	8	10
Relationship	7	1	1	2	8

Hypothesis 4: IMN offers a global (dataset-wide) interpretability of feature importance.

The purpose of this experiment is to showcase that IMN can be used to analyze the global interpretability of feature attributions, where the dataset-wide importance of the m-th feature is aggregated as $\frac{1}{N}\sum_{n=1}^{N}|w(x_n;\theta)_m\, x_{n,m}|$. Since we are not aware of a public benchmark offering ground-truth global interpretability of features, we experiment with the Adult Census Income (Kohavi et al., 1996), a very popular dataset, where the goal is to predict whether income exceeds \$50K/yr based on census data. We consider Decision Trees, CatBoost, TabNet, and IMN as explainable methods. Additionally, we use SHAP to explain the predictions of the TabResNet backbone.

We present the importance that the different methods assign to features in Table 5. To verify the feature rankings generated by the models, we analyze the top 5 features of every individual method by investigating the drop in model performance if we remove the feature. The more important a feature is, the more accuracy should drop when removing that feature. The results of Figure 6 show that IMNs have a higher relative drop in the model's accuracy when the most important predicted feature is removed. This shows that the feature ranking generated by IMN is proportional to the predictive importance of the feature and monotonously decreasing. In contrast, in the case of CatBoost, TabNet, SHAP, and Decision Trees, the decrease in accuracy is not proportional to the order of the feature importance (e.g. the case of Top-1 for Decision Trees, TabNet, SHAP or Top-2 for CatBoost).

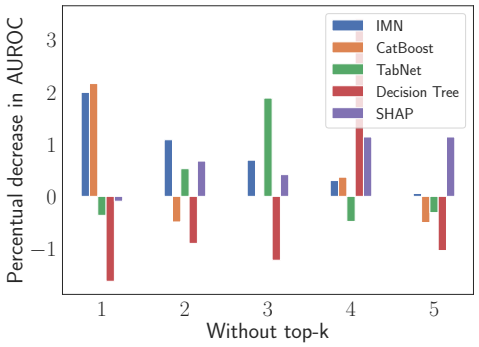

Figure 6: Investigating the decrease in AUROC when removing the k-th most important feature.

We additionally consider the task of predicting mushroom edibility (Lincoff, 1997). The *odor* feature allows one to predict whether a mushroom is edible or not and basing the predictions only on odor would allow a model to achieve more than 98.5% accuracy (Arik & Pfister, 2021). We run IMNs on the mushroom edibility task and we achieve a perfect test AUROC of 1. Furthermore, in Figure 7 we investigate the impact of every feature as described in Section 2.3, where, as observed, our method correctly identifies *odor* as the feature with the highest impact in the output. Based on the results, we conclude that **IMNs offer global interpretability**.

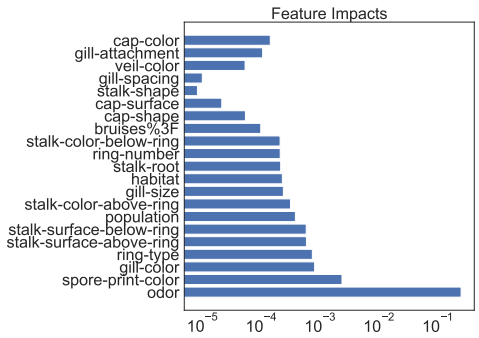

Figure 7: Feature impacts for the mushroom-edibility task.

6 Conclusion

In this work, we propose explainable deep networks that are comparable in performance to their black-box counterparts but also as interpretable as state-of-the-art explanation techniques. With extensive experiments, we show that the explainable deep learning networks outperform traditional white-box models in terms of performance. Moreover, the experiments confirm that the explainable deep-learning architecture does not include a significant degradation in performance or an overhead on time compared to the plain black-box counterpart, achieving competitive results against state-of-the-art classifiers in tabular data. Our method matches competitive state-of-the-art explainability methods on a recent explainability benchmark in tabular data, offering explanations of predictions as a free lunch.

7 Limitations and Future Work

One potential limitation of our method is that although interpretable, the per-instance models are linear. A potential future work can focus on generating other types of non-linear interpretable models, such as decision trees. More concretely, the hypernetwork can generate the parameters of the decision splits and the decision value at each node, as well as the leaf weights. Another potential strategy is to generate local Support Vector Machines, by expressing the prediction for a data point as a function of the similarity of the informative neighbors.

Acknowledgements

JG, **AK** and **SBA** would like to acknowledge the funding by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under grant number 417962828 and grant INST 39/963-1 FUGG (bwForCluster NEMO). In addition, **JG** and **AK** acknowledge the support of the BrainLinks-BrainTools center of excellence. Moreover, the authors acknowledge the support and HPC resources provided by the Erlangen National High Performance Computing Center (NHR@FAU) of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) under the NHR project v101be. NHR funding is provided by federal and Bavarian state authorities. NHR@FAU hardware is partially funded by the German Research Foundation (DFG) – 440719683.

References

Agarwal, R., Melnick, L., Frosst, N., Zhang, X., Lengerich, B., Caruana, R., and Hinton, G. E. Neural additive models: Interpretable machine learning with neural nets. *Advances in neural information processing systems*, 34:4699–4711, 2021.

Akiba, T., Sano, S., Yanase, T., Ohta, T., and Koyama, M. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 2623–2631, 2019.

Alvarez-Melis, D. and Jaakkola, T. S. Towards robust interpretability with self-explaining neural networks. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, pp. 7786–7795, Red Hook, NY, USA, 2018. Curran Associates Inc.

Ancona, M., Ceolini, E., Öztireli, C., and Gross, M. Towards better understanding of gradient-based attribution methods for deep neural networks. *arXiv preprint arXiv:1711.06104*, 2017.

Arik, S. Ö. and Pfister, T. Tabnet: Attentive interpretable tabular learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 6679–6687, 2021.

Berkson, J. A statistically precise and relatively simple method of estimating the bio-assay with quantal response, based on the logistic function. *Journal of the American Statistical Association*, 48(263):565–599, 1953.

Bischl, B., Casalicchio, G., Feurer, M., Gijsbers, P., Hutter, F., Lang, M., Mantovani, R. G., van Rijn, J. N., and Vanschoren, J. OpenML benchmarking suites. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. URL `https://openreview.net/forum?id=OCrD8ycKjG`.

Borisov, V., Leemann, T., Seßler, K., Haug, J., Pawelczyk, M., and Kasneci, G. Deep neural networks and tabular data: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–21, 2022. doi: 10.1109/TNNLS.2022.3229161.

Burkart, N. and Huber, M. F. A survey on the explainability of supervised machine learning. *Journal of Artificial Intelligence Research*, 70:245–317, 2021.

Chen, J., Song, L., Wainwright, M., and Jordan, M. Learning to explain: An information-theoretic perspective on model interpretation. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 883–892. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/chen18j.html.

Chen, J., Liao, K., Wan, Y., Chen, D. Z., and Wu, J. Danets: Deep abstract networks for tabular data classification and regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 3930–3938, 2022.

Cortez, P. and Embrechts, M. J. Opening black box data mining models using sensitivity analysis. In *Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining, CIDM 2011, part of the IEEE Symposium Series on Computational Intelligence 2011, April 11-15, 2011, Paris, France*, pp. 341–348. IEEE, 2011. doi: 10.1109/CIDM.2011.5949423. URL https://doi.org/10.1109/CIDM.2011.5949423.

Craven, M. and Shavlik, J. Extracting tree-structured representations of trained networks. *Advances in neural information processing systems*, 8, 1995.

Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. Least angle regression. *The Annals of Statistics*, 32(2):407 – 499, 2004. doi: 10.1214/009053604000000067. URL https://doi.org/10.1214/009053604000000067.

Freitas, A. A. Comprehensible classification models: a position paper. *ACM SIGKDD explorations newsletter*, 15(1):1–10, 2014.

Friedman, N., Geiger, D., and Goldszmidt, M. Bayesian network classifiers. *Machine learning*, 29: 131–163, 1997.

Frosst, N. and Hinton, G. E. Distilling a neural network into a soft decision tree. In Besold, T. R. and Kutz, O. (eds.), *Proceedings of the First International Workshop on Comprehensibility and Explanation in AI and ML 2017 co-located with 16th International Conference of the Italian Association for Artificial Intelligence (AI*IA 2017), Bari, Italy, November 16th and 17th, 2017*, volume 2071 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2017. URL https://ceur-ws.org/Vol-2071/CExAIIA_2017_paper_3.pdf.

Gijsbers, P., LeDell, E., Poirier, S., Thomas, J., Bischl, B., and Vanschoren, J. An open source automl benchmark. *arXiv preprint arXiv:1907.00909 [cs.LG]*, 2019. URL https://arxiv.org/abs/1907.00909. Accepted at AutoML Workshop at ICML 2019.

Goldstein, A., Kapelner, A., Bleich, J., and Pitkin, E. Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. *journal of Computational and Graphical Statistics*, 24(1):44–65, 2015.

Gorishniy, Y., Rubachev, I., Khrulkov, V., and Babenko, A. Revisiting deep learning models for tabular data. *Advances in Neural Information Processing Systems*, 34:18932–18943, 2021.

Gulum, M. A., Trombley, C. M., and Kantardzic, M. A review of explainable deep learning cancer detection models in medical imaging. *Applied Sciences*, 11(10):4573, 2021.

Ha, D., Dai, A. M., and Le, Q. V. Hypernetworks. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=rkpACe1lx.

Hendrycks, D. and Gimpel, K. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

Herbold, S. Autorank: A python package for automated ranking of classifiers. *Journal of Open Source Software*, 5(48):2173, 2020. doi: 10.21105/joss.02173. URL https://doi.org/10.21105/joss.02173.

Hollmann, N., Müller, S., Eggensperger, K., and Hutter, F. TabPFN: A transformer that solves small tabular classification problems in a second. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=cp5PvcI6w8_.

Hooker, S., Erhan, D., Kindermans, P.-J., and Kim, B. *A Benchmark for Interpretability Methods in Deep Neural Networks*. Curran Associates Inc., Red Hook, NY, USA, 2019.

Huang, G., Li, Y., Pleiss, G., Liu, Z., Hopcroft, J. E., and Weinberger, K. Q. Snapshot ensembles: Train 1, get m for free. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=BJYwwY9ll.

Kadra, A., Lindauer, M., Hutter, F., and Grabocka, J. Well-tuned simple nets excel on tabular datasets. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.

Kim, B., Shah, J. A., and Doshi-Velez, F. Mind the gap: A generative approach to interpretable feature selection and extraction. *Advances in neural information processing systems*, 28, 2015.

Kohavi, R. et al. Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid. In *Kdd*, volume 96, pp. 202–207, 1996.

Lincoff, G. H. *Field guide to North American mushrooms*. Knopf National Audubon Society, 1997.

Liu, Y., Khandagale, S., White, C., and Neiswanger, W. Synthetic benchmarks for scientific research in explainable machine learning. In *Advances in Neural Information Processing Systems Datasets Track*, 2021.

Loh, W.-Y. Classification and regression trees. *Wiley interdisciplinary reviews: data mining and knowledge discovery*, 1(1):14–23, 2011.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Lundberg, S. M. and Lee, S. A unified approach to interpreting model predictions. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 4765–4774, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html.

Luss, R., Chen, P.-Y., Dhurandhar, A., Sattigeri, P., Zhang, Y., Shanmugam, K., and Tu, C.-C. Leveraging latent features for local explanations. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '21, pp. 1139–1149, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383325. doi: 10.1145/3447548.3467265. URL https://doi.org/10.1145/3447548.3467265.

Martens, D., Baesens, B., Gestel, T. V., and Vanthienen, J. Comprehensible credit scoring models using rule extraction from support vector machines. *Eur. J. Oper. Res.*, 183(3):1466–1476, 2007. doi: 10.1016/j.ejor.2006.04.051. URL https://doi.org/10.1016/j.ejor.2006.04.051.

Montavon, G., Lapuschkin, S., Binder, A., Samek, W., and Müller, K. Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern Recognit.*, 65:211–222, 2017. doi: 10.1016/j.patcog.2016.11.008. URL https://doi.org/10.1016/j.patcog.2016.11.008.

Murphy, K. P. et al. Naive bayes classifiers. *University of British Columbia*, 18(60):1–8, 2006.

Pedregosa, Fabian acnd Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

Plumb, G., Molitor, D., and Talwalkar, A. S. Model agnostic supervised local explanations. *Advances in neural information processing systems*, 31, 2018.

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., and Gulin, A. Catboost: unbiased boosting with categorical features. *Advances in neural information processing systems*, 31, 2018.

Ras, G., Xie, N., van Gerven, M., and Doran, D. Explainable deep learning: A field guide for the uninitiated. *J. Artif. Intell. Res.*, 73:329–396, 2022. doi: 10.1613/jair.1.13200. URL https://doi.org/10.1613/jair.1.13200.

Ribeiro, M. T., Singh, S., and Guestrin, C. "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, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450342322. doi: 10.1145/2939672.2939778. URL https://doi.org/10.1145/2939672.2939778.

Sadhwani, A., Giesecke, K., and Sirignano, J. Deep Learning for Mortgage Risk*. *Journal of Financial Econometrics*, 19(2):313–368, 07 2020. ISSN 1479-8409. doi: 10.1093/jjfinec/nbaa025. URL https://doi.org/10.1093/jjfinec/nbaa025.

Shrikumar, A., Greenside, P., and Kundaje, A. Learning important features through propagating activation differences. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3145–3153. PMLR, 2017. URL http://proceedings.mlr.press/v70/shrikumar17a.html.

Simonyan, K., Vedaldi, A., and Zisserman, A. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

Somepalli, G., Schwarzschild, A., Goldblum, M., Bruss, C. B., and Goldstein, T. SAINT: Improved neural networks for tabular data via row attention and contrastive pre-training, 2022. URL https://openreview.net/forum?id=nL2lDlsrZU.

Staniak, M. and Biecek, P. Explanations of model predictions with live and breakdown packages. *arXiv preprint arXiv:1804.01955*, 2018.

Sundararajan, M., Taly, A., and Yan, Q. Axiomatic attribution for deep networks. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3319–3328. PMLR, 2017. URL http://proceedings.mlr.press/v70/sundararajan17a.html.

Tibshirani, R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

Tjoa, E. and Guan, C. A survey on explainable artificial intelligence (xai): Toward medical xai. *IEEE Transactions on Neural Networks and Learning Systems*, 32(11):4793–4813, 2021. doi: 10.1109/TNNLS.2020.3027314.

Wydmański, W., Bulenok, O., and Śmieja, M. Hypertab: Hypernetwork approach for deep learning on small tabular datasets. In *2023 IEEE 10th International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 1–9. IEEE, 2023.

Yang, C., Rangarajan, A., and Ranka, S. Global model interpretation via recursive partitioning. In *20th IEEE International Conference on High Performance Computing and Communications; 16th IEEE International Conference on Smart City; 4th IEEE International Conference on Data Science and Systems, HPCC/SmartCity/DSS 2018, Exeter, United Kingdom, June 28-30, 2018*, pp. 1563–1570. IEEE, 2018. doi: 10.1109/HPCC/SmartCity/DSS.2018.00256. URL https://doi.org/10.1109/HPCC/SmartCity/DSS.2018.00256.

Yeh, C.-K., Hsieh, C.-Y., Suggala, A. S., Inouye, D. I., and Ravikumar, P. *On the (in)Fidelity and Sensitivity of Explanations*. Curran Associates Inc., Red Hook, NY, USA, 2019.

Zhou, B., Khosla, A., Lapedriza, À., Oliva, A., and Torralba, A. Learning deep features for discriminative localization. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pp. 2921–2929. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.319. URL https://doi.org/10.1109/CVPR.2016.319.

A IMN can be extended to image classification backbones

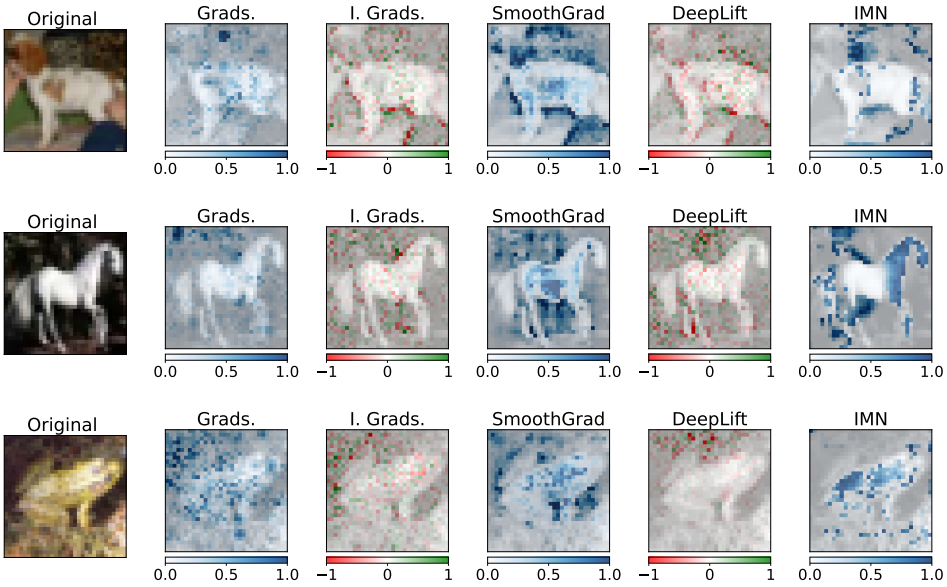

Figure 8: Comparison of IMN against explainability techniques for image classification.

We use IMN to explain the predictions of ResNet50, a broadly used computer vision backbone. We take the pre-trained backbone $\phi(\cdot) : \mathrm{R}^{H \times W \times K} \rightarrow \mathrm{R}^{D}$ from PyTorch and change the output layer to a fully-connected layer $w : \mathrm{R}^{D} \rightarrow \mathrm{R}^{H \times W \times K \times C}$ that generates the weights for multiplying the input image $x \in \mathrm{R}^{H \times W \times K}$ with K channels, and finally obtain the logits z_c for the class c. In this experiment, we use $\lambda = 10^{-3}$ as the L1 penalty strength.

We fine-tuned the ImageNet pre-trained ResNet50 models, both for the explainable (IMN-ResNet) and the black-box (ResNet) variants for 400 epochs on the CIFAR-10 dataset with a learning rate of 10^{-4}. To test whether the explainable variant is as accurate as the black-box model, we evaluate the validation accuracy after 5 independent training runs. IMN-ResNet achieves an accuracy of 87.49 ± 1.73 and the ResNet 88.76 ± 1.50, with the difference being statistically insignificant.

We compare our method to the following image explainability baselines: Saliency Maps (Gradients) (Simonyan et al., 2013), DeepLift (Shrikumar et al., 2017), Integrated Gradients (Ancona et al., 2017) with SmoothGrad. All of the baselines are available via the *captum* library[6]. We compare the rival explainers to IMN-ResNet by visually interpreting the pixel-wise weights of selected images in Figure 8. The results confirm that IMN-ResNet generates higher weights for pixel regions that include descriptive parts of the object.

B Plots

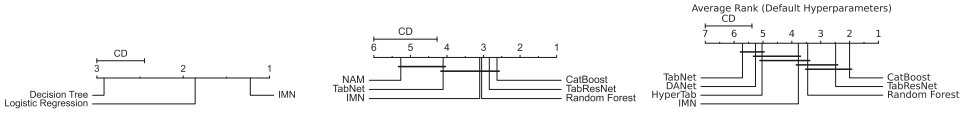

Figure 9: The critical difference diagrams that represent the average rank over all datasets based on the AUROC test performance for: **left)** The white-box methods and IMN, **middle)** The black-box methods and IMN for the binary classification datasets, **right)** The black-box methods and IMN for the entire benchmark of datasets.

In Figure 9, we repeat the experiments from Hypotheses 1 and 2, however, without performing hyperparameter optimization. Moreover, we consider two additional baselines, DANet (Chen et al.,

[6] `https://github.com/pytorch/captum`

2022) and HyperTab (Wydmański et al., 2023). As observed, our findings are consistent and both hypotheses are validated even when default hyperparameters are used for all the methods considered.

To further investigate the results on individual datasets, in Figure 10 we plot the distribution of the gains in performance of all methods over a single decision tree model (with default hyperparameters). The gain G of a method m run on a dataset D for a single run is calculated as shown in Equation 3.

$$G\,(m, DTree, D) = \frac{\text{AUROC}(m, D)}{\text{AUROC}(DTree, D)} \tag{3}$$

The results indicate that all methods except NAM achieve a comparable gain in performance across the AutoML benchmark datasets, while, the latter achieves a worse performance overall. We present detailed results in Appendix C.

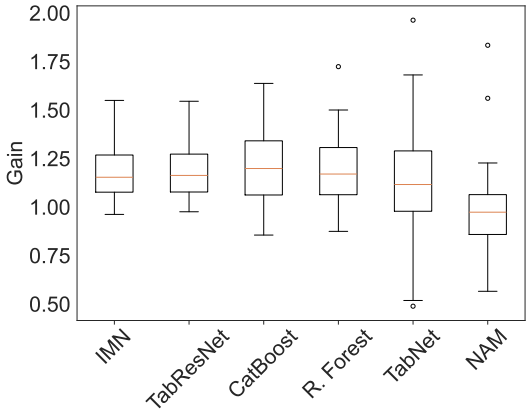

Figure 10: The gain distribution of the state-of-the-art models. The gain is calculated by dividing the test AUROC against the test AUROC of a decision tree.

Additionally, in Figure 11, we present the performance of the different explainers for the different explainability metrics. We present results for the Gaussian Non-Linear Additive and Gaussian Piecewise Constant datasets over a varying presence of correlation ρ between the features. The results show that our method achieves competitive results against Kernel Shap (K. SHAP) and LIME, the strongest baselines.

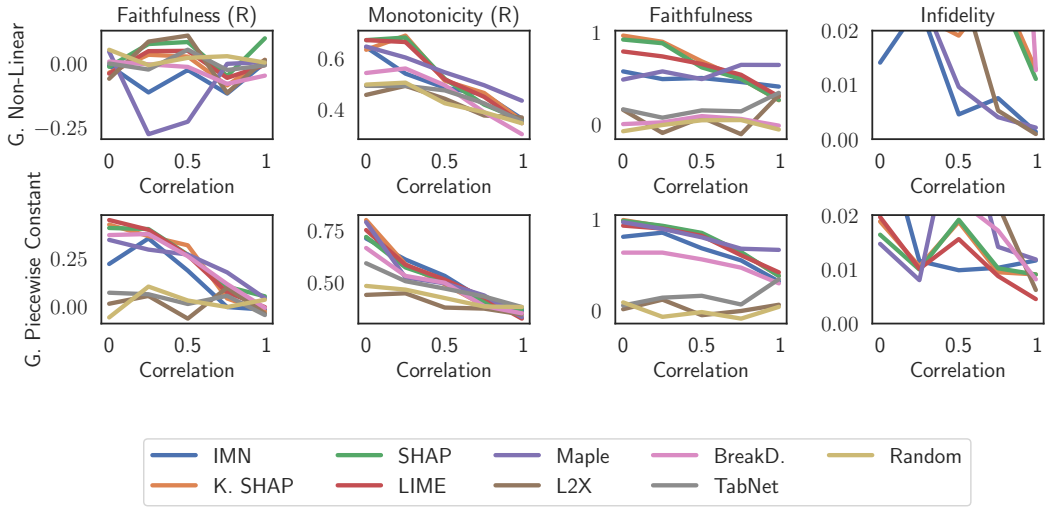

Figure 11: Performance analysis of all explainable methods on faithfulness (ROAR), monotonicity (ROAR), faithfulness, and infidelity. The results are shown for the Gaussian Non-Linear Additive and Gaussian Piecewise datasets where, correlation (ρ) ranges from [0, 1].

Lastly, to investigate how sensitive IMN is to the controlling hyperparameter configuration, we compare IMN and CatBoost (a method known for being robust to its hyperparameters in the community). Specifically, for every task, we plot the distribution of the performance of all hyperparameter configurations for every method. We present the results in Figure 12, where, as observed, IMN has a comparable sensitivity to CatBoost with regard to the controlling hyperparameter configuration. Moreover, in the majority of cases, the IMN validation performance does not vary significantly.

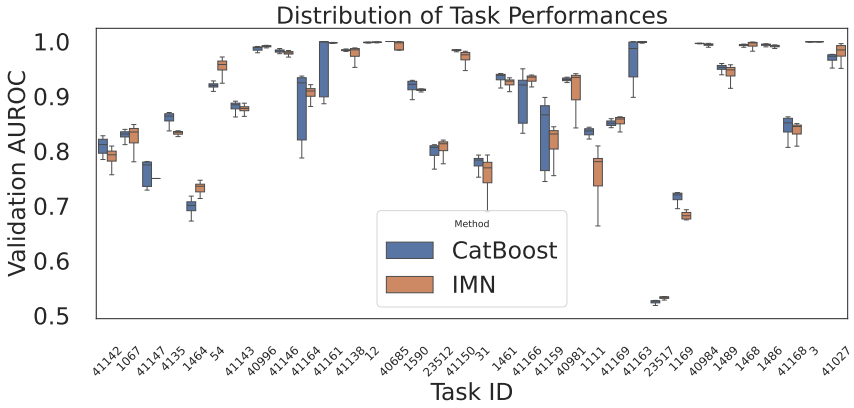

Figure 12: The distribution of the validation performance of the different hyperparameter configurations per task for CatBoost and IMN.

C Tables

To describe the 35 datasets present in our accuracy-related experiments, we summarize the main descriptive statistics in Table 6. The statistics show that our datasets are diverse, covering both binary and multi-class classification problems with imbalanced and balanced datasets that contain a diverse number of features and examples.

Additionally, we provide the per-dataset performances for the accuracy-related experiments of every method with the default configurations. Table 7 summarizes the performances on the train split, where, as observed Random Forest and Decision Trees overfit the training data excessively compared to the other methods. Moreover, Table 8 provides the performance of every method on the test split, where, IMN, TabResNet, and CatBoost achieve similar performances. We provide the same per-dataset performances of every method with the best-found hyperparameter configuration during HPO for the train split in Table 9 and test split in Table 10.

Lastly, we provide the HPO search spaces of the different methods considered in our experiments in Table 11, 12, 13, 14, 15, 16.

Table 6: Statistics regarding the AutoML benchmark datasets.

Dataset ID	Dataset Name	Number of Instances	Number of Features	Number of Classes	Majority Class Percentage	Minority Class Percentage
3	kr-vs-kp	3196	37	2	52.222	47.778
12	mfeat-factors	2000	217	10	10.000	10.000
31	credit-g	1000	21	2	70.000	30.000
54	vehicle	846	19	4	25.768	23.522
1067	kc1	2109	22	2	84.542	15.458
1111	KDDCup09 appetency	50000	231	2	98.220	1.780
1169	airlines	539383	8	2	55.456	44.544
1461	bank-marketing	45211	17	2	88.302	11.698
1464	blood-transfusion-service-center	748	5	2	76.203	23.797
1468	cnae-9	1080	857	9	11.111	11.111
1486	nomao	34465	119	2	71.438	28.562
1489	phoneme	5404	6	2	70.651	29.349
1590	adult	48842	15	2	76.072	23.928
4135	Amazon employee access	32769	10	2	94.211	5.789
23512	higgs	98050	29	2	52.858	47.142
23517	numerai28.6	96320	22	2	50.517	49.483
40685	shuttle	58000	10	7	78.597	0.017
40981	Australian	690	15	2	55.507	44.493
40984	segment	2310	20	7	14.286	14.286
40996	Fashion-MNIST	70000	785	10	10.000	10.000
41027	jungle chess	44819	7	3	51.456	9.672
41138	APSFailure	76000	171	2	98.191	1.809
41142	christine	5418	1637	2	50.000	50.000
41143	jasmine	2984	145	2	50.000	50.000
41146	sylvine	5124	21	2	50.000	50.000
41147	albert	425240	79	2	50.000	50.000
41150	MiniBooNE	130064	51	2	71.938	28.062
41159	guillermo	20000	4297	2	59.985	40.015
41161	riccardo	20000	4297	2	75.000	25.000
41163	dilbert	10000	2001	5	20.490	19.130
41164	fabert	8237	801	7	23.394	6.094
41165	robert	10000	7201	10	10.430	9.580
41166	volkert	58310	181	10	21.962	2.334
41168	jannis	83733	55	4	46.006	2.015
41169	helena	65196	28	100	6.143	0.170

Table 7: The per-dataset train AUROC performance for all methods in the accuracy experiments with default hyperparameter configurations. The train performance is the mean value from 10 runs with different seeds. A dashed line '-' represents a failure of running on that particular dataset.

Dataset ID	Decision Tree	Logistic Regression	Random Forest	TabNet	TabResNet	CatBoost	IMN
3	**1.000**	0.990	**1.000**	0.980	**1.000**	**1.000**	**1.000**
12	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
31	**1.000**	0.795	**1.000**	0.514	**1.000**	0.963	**1.000**
54	**1.000**	0.955	**1.000**	0.492	**1.000**	**1.000**	**1.000**
1067	**0.998**	0.818	0.997	0.825	0.928	0.971	0.920
1111	**1.000**	0.822	**1.000**	-	0.966	0.899	0.895
1169	**0.994**	0.680	**0.994**	0.705	0.697	0.733	0.697
1461	**1.000**	0.908	**1.000**	0.947	0.945	0.948	0.942
1464	0.983	0.757	0.978	0.490	0.830	**0.934**	0.834
1468	**1.000**	**1.000**	**1.000**	0.493	**1.000**	**1.000**	**1.000**
1486	**1.000**	0.988	**1.000**	0.995	0.999	0.997	0.998
1489	**1.000**	0.813	**1.000**	0.947	0.974	0.982	0.977
1590	**1.000**	0.903	**1.000**	0.920	0.920	0.935	0.920
1596	**1.000**	0.951	**1.000**	0.949	0.992	0.997	0.995
4135	**1.000**	0.839	0.998	-	0.890	0.981	0.877
23512	**1.000**	0.683	**1.000**	0.820	0.863	0.831	0.853
23517	**1.000**	0.533	**1.000**	0.529	0.587	0.703	0.546
40685	**1.000**	0.999	**1.000**	0.988	0.999	-	0.994
40981	**1.000**	0.932	**1.000**	0.472	**1.000**	0.996	**1.000**
40984	**1.000**	0.983	**1.000**	0.990	0.999	**1.000**	0.999
40996	**1.000**	0.989	**1.000**	0.997	**1.000**	0.999	**1.000**
41027	**1.000**	0.799	**1.000**	0.980	0.982	0.989	0.982
41138	**1.000**	0.992	**1.000**	0.999	0.999	**1.000**	0.996
41142	**1.000**	0.942	**1.000**	0.951	**1.000**	0.999	**1.000**
41143	**1.000**	0.868	**1.000**	0.874	**1.000**	0.992	**1.000**
41146	**1.000**	0.967	**1.000**	0.989	**1.000**	**1.000**	**1.000**
41147	**1.000**	0.746	**1.000**	-	0.769	0.827	0.763
41150	**1.000**	0.938	**1.000**	0.896	0.985	0.988	0.985
41159	**1.000**	0.826	**1.000**	0.840	**1.000**	0.977	**1.000**
41161	**1.000**	**1.000**	**1.000**	0.999	**1.000**	**1.000**	**1.000**
41163	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
41164	**1.000**	0.994	**1.000**	0.968	**1.000**	0.983	**1.000**
41165	**1.000**	**1.000**	**1.000**	0.876	**1.000**	**1.000**	**1.000**
41166	**1.000**	0.889	**1.000**	0.943	0.978	0.992	0.995
41168	**1.000**	0.804	**1.000**	0.911	0.915	0.971	0.921
41169	**1.000**	0.854	**1.000**	0.867	0.938	0.998	0.986

Table 8: The per-dataset test AUROC performance for all methods in the accuracy experiments with default hyperparameter configurations. The test performance is the mean value from 10 runs with different seeds. A dashed line '-' represents a failure of running on that particular dataset.

Dataset ID	Decision Tree	Logistic Regression	NAM	Random Forest	TabNet	TabResNet	CatBoost	IMN
3	0.987	0.990	0.977	0.998	0.983	**0.999**	**0.999**	**0.999**
12	0.938	**0.999**	-	0.998	0.995	**0.999**	**0.999**	**0.999**
31	0.643	0.775	0.717	**0.795**	0.511	0.756	0.790	0.751
54	0.804	0.938	-	0.927	0.501	**0.968**	0.934	0.957
1067	0.620	0.802	0.659	0.801	0.789	**0.808**	0.800	0.805
1111	0.535	0.816	0.544	0.793	-	0.778	**0.843**	0.816
1169	0.592	0.679	0.588	0.692	**0.699**	0.695	**0.718**	0.695
1461	0.703	0.908	0.827	0.930	0.926	0.931	**0.937**	0.930
1464	0.599	0.749	0.738	0.666	0.516	0.740	0.709	**0.742**
1468	0.926	**0.996**	-	0.995	0.495	0.995	**0.996**	0.994
1486	0.935	0.987	0.934	0.993	0.991	**0.994**	**0.994**	0.993
1489	0.842	0.805	0.806	**0.962**	0.928	0.949	0.948	**0.950**
1590	0.752	0.903	0.874	0.917	0.908	0.915	**0.930**	0.915
1596	0.942	0.951	-	**0.997**	0.949	0.991	0.996	0.994
4135	0.639	0.853	0.838	0.846	-	0.855	**0.883**	0.858
23512	0.626	0.683	0.583	0.794	0.803	**0.825**	0.804	0.823
23517	0.501	0.530	0.505	0.515	0.522	0.529	0.526	**0.530**
40685	0.967	0.994	-	**1.000**	0.986	0.995	-	0.993
40981	0.817	0.930	0.918	**0.945**	0.463	0.919	0.935	0.908
40984	0.946	0.980	-	**0.995**	0.985	0.994	**0.995**	0.994
40996	0.886	0.984	-	0.991	0.989	**0.994**	0.993	0.992
41027	0.792	0.797	-	0.931	0.976	**0.979**	0.974	0.978
41138	0.861	0.974	0.558	0.989	0.970	0.972	**0.992**	0.980
41142	0.626	0.742	0.724	0.796	0.713	0.782	**0.822**	0.775
41143	0.749	0.850	0.831	0.880	0.823	0.860	**0.870**	0.865
41146	0.910	0.966	-	0.983	0.974	0.982	**0.988**	0.981
41147	0.606	0.748	0.675	0.762	-	0.765	**0.779**	0.762
41150	0.867	0.938	0.912	0.981	0.896	**0.984**	**0.984**	**0.984**
41159	0.730	0.712	0.618	**0.892**	0.754	0.871	**0.897**	0.841
41161	0.857	0.995	0.972	0.999	0.997	0.998	**1.000**	0.998
41163	0.873	0.994	-	0.999	0.998	**1.000**	**1.000**	**1.000**
41164	0.786	0.898	-	0.925	0.888	0.913	**0.935**	0.902
41165	0.579	0.748	-	0.835	0.788	0.838	**0.895**	0.817
41166	0.699	0.882	-	0.927	0.918	**0.952**	0.949	0.943
41168	0.633	0.798	-	0.831	0.813	**0.868**	0.862	0.856
41169	0.554	0.841	-	0.800	0.842	**0.883**	0.866	0.865

Table 9: The per-dataset train AUROC performance for all methods in the accuracy experiments parametrized with the best hyperparameter configuration found during HPO. A dashed line '-' represents a failure to run on that particular dataset.

Dataset ID	Decision Tree	Logistic Regression	Random Forest	TabNet	TabResNet	CatBoost	IMN
3	0.999	0.995	0.997	-	1.000	**1.000**	1.000
12	0.986	**1.000**	0.999	1.000	**1.000**	**1.000**	**1.000**
31	0.820	0.785	0.945	**0.982**	0.806	0.888	0.813
54	0.930	0.960	0.971	0.988	0.990	**1.000**	0.968
1067	0.831	0.807	0.819	**0.891**	0.813	0.875	0.810
1111	0.836	0.829	**0.885**	-	0.825	0.842	0.844
1169	0.682	0.680	0.690	0.714	0.696	**0.754**	0.683
1461	0.900	0.907	0.922	0.946	0.947	**0.954**	0.944
1464	0.796	0.765	0.867	0.835	0.811	**0.947**	0.763
1468	0.972	1.000	0.996	1.000	**1.000**	1.000	**1.000**
1486	0.981	0.988	0.987	0.997	0.998	**0.999**	0.997
1489	0.908	0.815	0.938	0.993	0.995	**1.000**	0.998
1590	0.904	0.903	0.911	-	0.920	**0.943**	0.926
1596	0.931	0.951	0.946	0.952	-	**0.998**	-
4135	0.833	0.826	0.865	-	0.843	**0.995**	0.850
23512	0.737	0.684	0.766	0.831	0.869	**0.900**	0.859
23517	0.529	0.532	0.562	0.523	0.531	**0.587**	0.535
40685	1.000	0.999	1.000	1.000	1.000	**1.000**	0.996
40981	0.931	0.932	**0.978**	-	0.949	0.943	0.947
40984	0.989	0.988	0.996	0.999	0.999	**1.000**	0.999
40996	0.956	0.988	0.976	0.998	**1.000**	0.994	-
41027	0.873	0.801	0.905	**0.999**	0.996	0.989	0.996
41138	0.979	0.989	0.989	0.996	0.992	**1.000**	0.991
41142	0.786	0.874	0.860	-	0.976	0.952	**1.000**
41143	0.849	0.872	0.934	0.877	0.977	**0.997**	0.941
41146	0.966	0.968	0.986	**1.000**	1.000	0.999	0.994
41147	0.727	0.745	0.746	-	0.768	**0.865**	0.750
41150	0.946	0.956	0.961	0.968	0.989	**1.000**	0.989
41159	0.775	0.823	0.878	0.552	**1.000**	**1.000**	1.000
41161	0.860	0.999	0.961	0.999	1.000	**1.000**	1.000
41163	0.901	1.000	0.976	1.000	**1.000**	**1.000**	**1.000**
41164	0.707	0.975	0.901	0.998	0.982	0.999	**0.999**
41165	0.750	0.936	0.840	0.960	**0.995**	0.988	-
41166	0.822	0.892	0.867	0.986	0.984	**1.000**	0.999
41167	0.941	0.997	-	**1.000**	-	-	-
41168	0.787	0.804	0.823	0.908	0.900	**0.944**	0.908
41169	0.791	0.853	0.846	0.879	0.926	**0.976**	0.961

Table 10: The per-dataset test AUROC performance for all methods in the accuracy experiments parametrized with the best hyperparameter configuration found during HPO. A dashed line '-' represents a failure to run on that particular dataset.

Dataset ID	Decision Tree	Logistic Regression	NAM	Random Forest	TabNet	TabResNet	CatBoost	IMN
3	0.999	0.997	0.977	0.998	-	1.000	1.000	**1.000**
12	0.960	0.998	-	0.998	0.998	0.998	0.998	**1.000**
31	0.765	0.852	0.717	0.826	0.741	0.844	0.826	**0.863**
54	0.844	0.959	-	0.924	0.955	**0.966**	0.935	0.959
1067	0.777	0.804	0.659	0.813	0.784	0.795	**0.834**	0.805
1111	0.805	0.814	0.544	0.829	-	0.801	**0.839**	0.806
1169	0.679	0.678	0.588	0.686	0.702	0.693	**0.724**	0.680
1461	0.903	0.907	0.827	0.919	0.917	0.932	**0.939**	0.933
1464	0.647	0.729	**0.738**	0.679	0.668	0.697	0.678	0.710
1468	0.954	0.999	-	0.988	0.987	0.999	0.994	**0.999**
1486	0.977	0.986	0.934	0.985	0.989	0.994	**0.996**	0.992
1489	0.890	0.802	0.806	0.920	0.945	**0.958**	0.955	0.956
1590	0.901	0.901	0.874	0.908	-	0.913	**0.930**	0.913
1596	0.931	0.951	-	0.946	0.952	-	**0.997**	-
4135	0.845	0.854	0.838	0.870	-	0.861	**0.909**	0.865
23512	0.725	0.681	0.583	0.754	0.803	**0.817**	0.808	0.817
23517	0.522	0.530	0.505	0.529	0.525	0.527	**0.532**	0.531
40685	0.908	0.999	-	1.000	1.000	0.999	**1.000**	1.000
40981	0.905	0.916	0.918	**0.933**	-	0.916	0.920	0.909
40984	0.981	0.989	-	0.991	0.995	0.994	**0.996**	0.993
40996	0.954	0.985	-	0.976	0.989	**0.994**	0.991	-
41027	0.870	0.800	-	0.900	**0.999**	0.993	0.980	0.991
41138	0.974	0.986	0.558	0.985	**0.987**	0.986	0.986	0.986
41142	0.744	0.811	0.724	0.794	-	0.814	**0.819**	0.808
41143	0.843	0.839	0.831	0.871	0.846	0.860	**0.871**	0.856
41146	0.966	0.962	-	0.975	**0.991**	0.984	0.987	0.984
41147	0.728	0.748	0.675	0.748	-	0.767	**0.786**	0.754
41150	0.942	0.954	0.912	0.958	0.967	0.985	**0.986**	0.984
41159	0.753	0.725	0.618	0.847	0.543	0.871	**0.912**	0.868
41161	0.849	0.997	0.972	0.950	0.996	0.999	**0.999**	0.998
41163	0.885	0.996	-	0.969	0.997	1.000	**1.000**	1.000
41164	0.697	0.914	-	0.877	0.869	0.919	**0.937**	0.919
41165	0.706	0.822	-	0.804	0.777	0.870	**0.875**	-
41166	0.811	0.886	-	0.862	0.929	0.954	**0.956**	0.947
41168	0.768	0.797	-	0.809	0.808	0.866	**0.871**	0.856
41169	0.762	0.845	-	0.823	0.839	**0.884**	0.869	0.873

Table 11: The hyperparameter search space for IMN. TabResNet has the same search space without weight normalization.

Hyperparameter	**Type**	**Range**	**Log scale**
nr_epochs	Integer	$[10, 500]$	-
$learning\_rate$	Continuous	[1e-5, 1e-1]	✓
$batch\_size$	Categorical	$\{32, 64, 128, 256, 512\}$	-
$weight\_decay$	Continuous	$[1e-5, 1e-1]$	✓
$weight\_norm$	Continuous	$[1e-5, 1e-1]$	✓
$dropout\_rate$	Continuous	$[0, 0.5]$	-

Table 12: The hyperparameter search space for logistic regression.

Hyperparameter	**Type**	**Range**	**Log scale**
C	Continuous	$[1e-5, 5]$	-
$penalty$	Categorical	$\{l2, none\}$	-
$max\_iterations$	Integer	$[50, 500]$	-
$fit\_intercept$	Categorical	$\{True, False\}$	-

Table 13: The hyperparameter search space for a decision tree.

Hyperparameter	**Type**	**Range**	**Log scale**
$criterion$	Categorical	$\{Gini, Entropy\}$	-
max_depth	Integer	$[1, 21]$	-
$min\_samples\_split$	Integer	$[2, 11]$	-
max_leaf_nodes	Integer	$[3, 26]$	-
$splitter$	Categorical	$\{Best, Random\}$	-

Table 14: The hyperparameter search space for CatBoost.

Hyperparameter	Type	Range	Log scale
$learning\_rate$	Continuous	$[1e-5, 1]$	✓
$random\_strength$	Integer	$[1, 20]$	-
$l2\_leaf\_reg$	Continuous	$[1, 10]$	✓
$bagging\_temperature$	Continuous	$[1e-6, 1]$	✓
$leaf\_estimation\_iterations$	Integer	$[1, 20]$	-
$iterations$	Integer	$[100, 4000]$	-

Table 15: The hyperparameter search space for Random Forest.

Hyperparameter	Type	Range	Log scale
$criterion$	Categorical	$\{Gini, Entropy\}$	-
max_depth	Integer	$[1, 21]$	-
$min\_samples\_split$	Integer	$[2, 11]$	-
max_leaf_nodes	Integer	$[3, 26]$	-
$n\_estimators$	Integer	$[100, 4000]$	-

Table 16: Hyperparameter search space for the TabNet model.

Hyperparameter	Type	Range	Log scale
n_a	Categorical	$\{8, 16, 24, 32, 64, 128\}$	-
$learning\_rate$	Categorical	$\{0.005, 0.01, 0.02, 0.025\}$	-
$gamma$	Categorical	$\{1.0, 1.2, 1.5, 2.0\}$	-
n_steps	Categorical	$\{3, 4, 5, 6, 7, 8, 9, 10\}$	-
$lambda\_sparse$	Categorical	$\{0, 0.000001, 0.0001, 0.001, 0.01, 0.1\}$	-
$batch\_size$	Categorical	$\{256, 512, 1024, 2048, 4096, 8192, 16384, 32768\}$	-
$virtual\_batch\_size$	Categorical	$\{256, 512, 1024, 2048, 4096\}$	-
$decay\_rate$	Categorical	$\{0.4, 0.8, 0.9, 0.95\}$	-
$decay\_iterations$	Categorical	$\{500, 2000, 8000, 10000, 20000\}$	-
$momentum$	Categorical	$\{0.6, 0.7, 0.8, 0.9, 0.95, 0.98\}$	-

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The results in Section 2.4 and Section 5 validate our claims.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations of the proposed method are mentioned in Section 7.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: There is no theory in this paper.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: In Sections 2 and 4 we provide all the information regarding our method/baselines and the preprocessing of the data. We additionally open-source the code. Lastly, the results are reproducible as the experiments were seeded.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The code is open-sourced (Section 4) and all of the necessary information regarding the datasets is provided in Table 6, combined with their online identifier where they can be easily accessed from OpenML.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: The information is provided in Section 4. The code is additionally open-sourced.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Critical difference diagrams that provide statistical significance information are provided in Section 5.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar then state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: All the necessary information is provided in Section 4 and 5.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The impact of our work has been mentioned in the Introduction Section and in Section 5.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Everything that was used from previous work be it a method or dataset has been properly cited in the manuscript.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The code is provided and documented.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Not applicable.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Not applicable.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.