Exploring Multi-Task Learning for Explainability

Foivos Charalampakos(✉) and Iordanis Koutsopoulos(✉)

Department of Informatics, Athens University of Economics and Business, Athens, Greece

phoebuschar@aueb.gr, jordan@aueb.gr

Abstract. Machine Learning (ML) model understanding and interpretation is an essential component of several applications in different domains. Several explanation techniques have been developed in order to provide insights about decisions of complex ML models. One of the most common explainability methods, Feature Attribution, assigns an importance score to each input feature that denotes its contribution (relative significance) to the complex (black-box) ML model's decision. Such scores can be obtained through another model that acts as a surrogate, e.g., a linear one, which is trained after the black-box model so as to approximate its predictions. In this paper, we propose a training procedure based on Multi-Task Learning (MTL), where we *concurrently* train a black-box neural network and a surrogate linear model whose coefficients can then be used as feature significance scores. The two models exchange information through their predictions via the optimization objective which is a convex combination of a predictive loss function for the black-box model and of an explainability metric which aims to keep the predictions of the two models close together. Our method manages to make the surrogate model achieve a more accurate approximation of the black-box one, compared to the baseline of separately training the black-box and surrogate models, and therefore improves the quality of produced explanations, both global and local ones. We also achieve a good trade-off between predictive performance and explainability with minimal to negligible accuracy decrease. This enables black-box models acquired from the MTL training procedure to be used instead of normally trained models whilst being more interpretable.

Keywords: Multi-Task Leaning · Explainable Artificial Intelligence · Feature Attribution methods

1 Introduction

Contemporary, complex Deep Neural Networks (DNNs) are increasingly used in order to assist the decision-making process. Despite their impressive predictive abilities, these networks provide a very limited understanding of the reasoning behind their decisions [15]. In domains with high-stakes applications such as law, finance and healthcare, model understanding and therefore interpretation is essential so that the model's predictions can be trusted [15]. Interpretability

S. Nowaczyk et al. (Eds.): ECAI 2023 Workshops, CCIS 1947, pp. 349–365, 2024.
https://doi.org/10.1007/978-3-031-50396-2_20

of ML algorithms has thus become a pressing issue, and the field of eXplainable - or Interpretable - Artificial Intelligence (XAI) has emerged and constitutes an important component of Trustworthy AI.

XAI methods can be arranged to several categories according to different criteria. The most apparent distinction is the one of 'transparent' versus 'opaque' models. The former category concerns models like Linear/Logistic Regression and Decision Trees whose structure is simple, and their decision-making process is understandable by humans. Unfortunately, the simplicity of these models often comes with an unsatisfactory performance in real-world applications. This caveat is known as the accuracy-interpretability trade-off. XAI aims to fill this gap by providing explainability for 'opaque' models such as Neural Networks and Random Forests which require the development of separate specialized algorithms in order to render their predictions interpretable [15]. Usually, these algorithms make use of the predictions produced by the model after its training, and are referred to as *post-hoc* explainability methods.

Post-hoc methods can be further categorized into global and local methods. The former aim at explaining the general machinery of the ML model, by describing its average behavior over the entire dataset [5], while local methods focus on explaining predictions for individual data instances [5]. Another categorization is based on whether the algorithm is model-agnostic (i.e., it does not require access to the model architecture) or model-specific.

One well-known class of explainability algorithms are the Feature Attribution (FA) methods [6] which rely on a score that captures how much the input features contribute to the model's output. FA methods can be used in both global and local settings, as well as in model-specific [2] and model-agnostic [6] contexts. On the other hand, the class of counterfactual explanations [7] concerns local model-agnostic methods that describe the smallest changes to the feature values that change the output of the prediction for a given instance, while decision rule-based explanations are simple IF-THEN natural language hypothetical statements, consisting of a condition which contains one or more input features, and a corresponding prediction based on the values of the features involved in the condition [5].

Real-world problems are multi-objective ones, which means that ML training should address multiple tasks simultaneously, possibly belonging to different data modalities. For example, an autonomous vehicle should be able to segment the lane markings, detect humans, locate road signs, and identify their meaning [21]. In the medical sector, prediction accuracy and prediction explainability are simultaneously required, e.g., when a patient should be informed about potential side-effect risks for a particular tretment plan. Such problems motivate the development of Deep Learning models that, given an input, can infer several desired task outputs [21]. This kind of models can be trained using the *Multi-Task Learning (MTL)* paradigm that permits multiple tasks to be concurrently learned by a single model, enabling the different tasks to share potential common underlying information, and removing the need for training different models for each task. In the case of XAI, a way to use MTL is to *think of prediction and*

explainability as two distinct tasks, and to simultaneously solve for these tasks in order to allow information exchange between the two tasks and to produce more specific and accurate explanations for the predictions.

In this work, we utilize the MTL paradigm, which has recently been used in the field of XAI [8,24,37], in order to develop a framework that concurrently solves a ML prediction task and an explainability task. We focus on surrogate models and employ them to produce FA explanations. We aim at finding a black-box neural network model f along with a surrogate approximation model g, by forcing the former to take into account, during training, how well it is approximated by the latter. To that end, we optimize a loss function that includes a term for predictive training loss and an explainability-based metric. For the latter, we use a known explainability metric such as fidelity, which measures the difference between the predictions of g and f. This component aims to improve f's approximation through g and to enhance the quality of post-hoc explanations of the black-box model. Furthermore, the combined objective acts as the information-sharing 'channel' between the two models in the course of back-propagation [18] during the *joint* training. In another point of view, g could be considered as an explainability-regularizing model that constrains the values of f's predictions to being similar to those of the interpretable model g. In order to demonstrate the concept of our approach, we choose g to be a parameterized linear model which can be trained along with the black-box, but other choices are possible as well. Using such linear models, feature importance explanations for the predictions of f can be acquired through the coefficients of g [5].

We experiment with a variety of regression and binary classification tasks, where we compare models trained with and without MTL. We show that, our approach that uses MTL to concurrently train f and g, results in a more accurate approximation of the black-box by the surrogate linear model, compared to the standard practice where the two models are trained sequentially and separately. Therefore, the global explanation's fidelity is very much improved and in addition, only a minimal drop in the predictive performance is observed as a trade-off. Furthermore, we show that the same black-box model can be more accurately approximated by local linear explainers (like Local Interpretable Model-Agnostic Explanations (LIME) [6]), thus resulting on a lower-fidelity local explanation.

2 Related Work

2.1 Feature Attribution (FA) Methods for Explainability

FA algorithms are most commonly used as local explainers and assign importance scores to how much a given input feature contributes to the model's prediction result for a single instance of interest. Much work has been done on model-specific techniques that are gradient-based and work for DNNs by computing the significance of input features based on the gradient values of the model's parameters [2,25]. Another line of research works create a local neighborhood around the instance of interest $\mathbf{x}$ based on perturbations of $\mathbf{x}$'s feature values and measure the change in the model's output in order to calculate the significance

of each feature [6,29,35], based on a surrogate model. One of the most popular FA explanation systems, LIME [6], results to a local surrogate model-based explanation by optimizing the following objective, given the instance of interest $\mathbf{x}$ and a trained black-box model f:

$$e(\mathbf{x}, f) = \arg\min_{g \in \mathcal{G}} \left[\sum_{\mathbf{x}' \in N_{\mathbf{x}}} \mathbf{w}_{\mathbf{x}} \left(f(\mathbf{x}') - g(\mathbf{x}') \right)^2 + \Omega(g) \right] \tag{1}$$

where $N_{\mathbf{x}}$ is the neighborhood around $\mathbf{x}$, consisting of synthetic perturbations of $\mathbf{x}$. The class of surrogate models is denoted by $\mathcal{G}$ (e.g., linear models or decision trees), and $\Omega(g)$ is a measure of complexity that encourages desirable properties of g such as sparsity, i.e., using a small number of features [6]. LIME also weighs each neighbor of $\mathbf{x}$ to denote its importance, using a proximity measure (e.g., ℓ_2-distance from $\mathbf{x}$) and solves a *weighted linear regression objective*, using a weight vector $\mathbf{w}_{\mathbf{x}}$. The form of the resulting explanation $e(\mathbf{x}, f)$ depends on $\mathcal{G}$. For instance, if $\mathcal{G}$ includes all possible linear functions, then $e(\mathbf{x}, f)$ will consist of the coefficients of the learned linear function g, while in the case of decision trees, $e(\mathbf{x}, f)$ will consist of decision rules based on the trained tree.

In addition to local explainability, FA methods have also been used for global explainability through global surrogate models which aim to approximate (mimic) the predictions of the underlying black-box model [5,10]. Global surrogate models are similar to local surrogate models, except that they are trained by using the whole dataset and not just a generated neighborhood of a specific instance $\mathbf{x}$. The most common way to learn a global surrogate model is to train it on the predictions $\{\mathbf{x}_i, f(\mathbf{x}_i)\}_{i=1}^N$ of the black-box model, where $\mathbf{x}_i$, $f(\mathbf{x}_i)$ respectively are the i-th input training feature vector and the corresponding black-box model's output. This is also the baseline that we use in our experiments for global explainability.

2.2 Multi-Task Learning

MTL has been extensively studied for training a model on multiple tasks at the same time. This formulation can result in both improved training efficiency and better model performance for each task [14]. The most widely used multi-task learning architecture comprises a shared-parameter model structure, where the first (representation learning) layers are shared across tasks [21] and N task-specific parallel heads are added on top, one for each task. This approach is called a hard parameter-sharing one, where essentially the parameters are divided into shared and task-specific [21]. In an alternative approach, the soft parameter-sharing one, there are no shared layers, and each task is assigned its own set of parameters, a subset of weights of the DNN corresponds to a certain task. In addition, a mechanism is employed to allow information flow among tasks (i.e., soft sharing) [21,22]. For example, individual (task) modules could exchange information by sharing a segment of their learned latent features (also see Fig. 1). Clearly, the soft parameter-sharing approach requires more training time and computational resources due to the larger number of task-specific parameters.

However, it can prove more useful in settings where the tasks at hand are not so closely related.

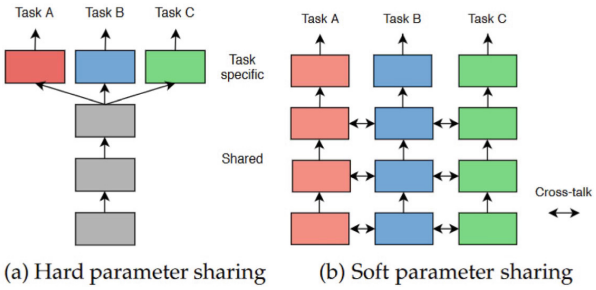

Fig. 1. Two widely used MTL architectures. Each box represents a layer. In (a), the hard parameter-sharing approach is depicted. Grey boxes denote shared layers while colored ones denote task-specific heads. In (b), the soft parameter-sharing approach is shown with no shared layers. Three dedicated subsets of the model's parameters correspond to the three different tasks. Figure is taken from [21] (Color figure online).

In this work, a soft sharing-based approach is utilized, where the surrogate model g does not share parameters with the black-box f in order to preserve the former model's transparency (by keeping its linear structure), and the two models exchange information only through their respective predictions which we aim to make as similar as possible. In other words, we treat the black-box model f and the surrogate model g as two separate sets of parameters, one for each task, which however communicate through the optimization of the joint training loss function which includes both f and g.

MTL has recently been used as a facilitator of XAI in specific settings. Some works propose its use in the design of explainable recommendation systems, either by producing accompanying textual explanations about the recommendation [8] or by solving joint tensor factorization objectives of *"user preference modeling for recommendation"* and *"opinionated content modeling for explanation"* that involve tensors regarding the user, the items and the users' preferences on individual items' features [37]. Another line of work, related to ours [24], considers MTL for weakly-supervised concept-based explainability. In a fraud detection setting, the authors employ *distant supervision* using domain knowledge and a rule-based database in order to acquire imprecise (noisy) concept explainability labels. They map rule descriptions present in the database that hold for specific data instances to concepts which stem from a concept taxonomy (related to the task). For instance: {Rule: Order contains risky product styles. → Concepts: Suspicious Items}.

They also explore various training strategies for jointly training ML models for two classification tasks, one about a prediction task and one based on the concept labels which is essentially a multi-label classification task.

Following a rationale similar to that of [8,24], in this work, we jointly solve a prediction and an explainability task. However, our approach differs in the following. First, instead of solving an additional supervised learning task such as text generation [8] or classification of concept categories [24], we make use of a quantitative explainability-related metric as one of the two objectives, corresponding to the task of explainablity, and we incorporate it into the loss function. Additionally, we focus on surrogate models that produce feature importance values without the need for any additional labeled data (e.g., text reviews or interpretable concepts). Our method aims at obtaining an accurate black-box model while at the same time learning a better approximation of it through the surrogate model. On the contrary, in the baseline, currently used method, a surrogate is obtained separately, after the training of the black-box is completed. Thus, the adoption of MTL allows us to achieve this improved approximation as the parameters of the black-box model are updated through the shared optimization objective with respect to the performance of the explainability task which quantitatively measures how accurate the approximation is between the black-box and the surrogate models.

2.3 Explainability Through Regularization

Some works consider the direction of explainability-based model optimization, which we also address in this work. However, they use various types of regularizers in the optimization scheme of the black-box model. The method of Functional Transparency for Structured Data (FTSD) [33] uses a non-differentiable game-theoretic approach to regularize black-box models so that they become more locally interpretable. It focuses on graph and time-series data, and thus requires domain knowledge to define the neighborhood $N_{\mathbf{x}}$. Self-Explaining Neural Networks (SENN) [31] generalize linear models, enriching them with complex features and maintaining interpretability via gradient regularization and an autoencoder network. The Right for the Right Reasons (RRR) method [11] and some similar works [4,30,36] use domain knowledge to decide on the features that are used by the underlying model through a loss regularizer. This regularization affects the model's explanations. Regularization for tree-based approximation was proposed in [12,13]. Finally, Explanation-based Optimization (ExpO) [1] uses a model-agnostic regularizer based on XAI metrics aiming at improving the quality of local post-hoc explanations of the black-box model.

Our work is related to these methods on the aspect of explainability-based optimization. However, different from these works, we utilize MTL, which allows us to obtain a more interpretable black-box model as well as an explainer without affecting the black-box architecture. Furthermore, our approach does not require access to domain knowledge, thus removing the need for costly feature engineering and supplementary data.

3 MTL-Inspired Explainability

In this section, we present our proposed framework that leverages MTL in order to enhance explainability. Our approach addresses both a prediction and an explainability task, each characterized by a distinct loss function. We use a formulation in which these two losses are fused using a convex combination. The goal is to *jointly* train a black-box model and a surrogate model that tries to approximate the predictions of the former. We concurrently update the parameters of the two models using the combined loss objective that consists of the two loss components. The first component represents the predictive training loss of the black-box model, while the second one utilizes an explainability metric to assess the quality of the surrogate model's approximation.

3.1 Background

We consider a supervised learning setting [34], where the objective is to learn a ML model f, namely a mapping from a vector space $\mathcal{X}$ to a target space $\mathcal{Y}$, with $f \in \mathcal{F} : \mathcal{X} \rightarrow \mathcal{Y}$, where $\mathcal{F}$ is the function family, and the target variable $y \in \mathcal{Y}$ can be either a real value (in regression problems) or a categorical value (in classification problems). In ML settings, f is modeled as a DNN parameterized by a set of parameters θ (henceforth f_θ) that is trained with data $\mathcal{D} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N$ using a loss function $\mathcal{L}_{STL}$ in the single-task scenario (e.g., cross-entropy for a classification task - note that STL stands for Single-Task Loss).

In MTL, f_θ is learned with respect to multiple objectives which are most commonly combined in a weighted linear sum:

$$\mathcal{L}_{MTL} = \sum_{j=1}^{m} \alpha_j \mathcal{L}_{STL_j} \tag{2}$$

where $\alpha_j \in \mathbb{R}$ is the weight for the j-th task and m is the number of distinct tasks. In addition, the model is trained using data in the form of $\mathcal{D} = \{\mathbf{x}_i, [\mathbf{y}_{i1}, \ldots, \mathbf{y}_{ij}, \ldots, \mathbf{y}_{im}]\}_{i=1}^N$ where $\mathbf{y}_{ij}$ is the target for the i-th training example and the j-th task.

In this work, we aim to generate explanations in the form of feature importances. Therefore, one of the objectives will be responsible for the explainability, while the other will be responsible for the prediction task. A system that produces such explanations is denoted as $e : \mathcal{X} \times \mathcal{F} \rightarrow \mathcal{E}$, where $\mathcal{E}$ is the family of possible explanations and is defined as $\mathcal{E} = \{g_q \in \mathcal{G} \mid g_q : \mathcal{X} \rightarrow \mathcal{Y}\}$. In this work, $\mathcal{G}$ is the set of linear functions which are suitable for producing feature-based explanations. Therefore, since the explanations will be based on the coefficients of the learned linear function, we have that $\mathcal{E} = \mathcal{G}$. Moreover, q denotes the parameter set (i.e., the coefficients) of g_q.

3.2 Explainability Metrics: Fidelity

Several metrics have been developed to objectively assess the quality of explanations according to different criteria [3]. A common choice for the evaluation

of feature-based explanations is to estimate how accurately g_q approximates the behavior of f_θ for each target sample $\mathbf{x}$ [1,23]. This can be captured through the squared difference:

$$\mathrm{PF}(f_\theta, g_q, \mathbf{x}) = \left(g_q(\mathbf{x}) - f_\theta(\mathbf{x})\right)^2 \tag{3}$$

which is referred to as Point Fidelity [6,29]. The Global Fidelity is obtained as the average of Point Fidelity values, across all N samples,

$$\mathrm{GF}(f_\theta, g_q) = \frac{1}{N}\sum_{i=1}^{N}\left[\mathrm{PF}(f_\theta, g_q, \mathbf{x}_i)\right] . \tag{4}$$

Fidelity is also used in cases that involve *locality*, where it is used to measure how good g_q is in modeling f_θ in a local neighborhood $N_\mathbf{x}$ of point $\mathbf{x}$, which consists of synthetically generated perturbations of $\mathbf{x}$'s feature values [1,23],

$$\mathrm{NF}(f_\theta, g_q, \mathbf{x}) = \frac{1}{|N_\mathbf{x}|}\sum_{\mathbf{x}'\in N_\mathbf{x}}\left[\left(g_q(\mathbf{x}') - f_\theta(\mathbf{x}')\right)^2\right] \tag{5}$$

and is called Neighborhood Fidelity [1]. Similar to Point Fidelity, we can average across all data points to get a 'global' Neighborhood Fidelity (GNF) measure for the entire dataset:

$$\mathrm{GNF}(f_\theta, g_q) = \frac{1}{N}\sum_{i=1}^{N}\left[\mathrm{NF}(f_\theta, g_q, \mathbf{x}_i)\right] . \tag{6}$$

3.3 Optimization Objective

As mentioned above, the intention is to compute the parameters of both the black-box and the explainable models in a way that g_q's predictions are as close as possible to f_θ's ones, while also catering for the latter model's predictive performance.

Specifically, we want to train f_θ and g_q by solving the following optimization problem:

$$(\hat{f}_\theta, \hat{g}_q) = \underset{(f_\theta, g_q)\in\mathcal{F}\times\mathcal{E}}{\arg\min}\ \frac{1}{N}\sum_{i=1}^{N}\left[\alpha\cdot\mathcal{L}_{base}(f_\theta(\mathbf{x}_i), y_i) + (1-\alpha)\cdot PF(f_\theta, g_q, \mathbf{x}_i)\right] \tag{7}$$

where $\hat{f}_\theta, \hat{g}_q$ are the acquired black-box and surrogate models respectively, after the MTL training process. The function $\mathcal{L}_{base}(\cdot)$ is a prediction loss function (e.g., squared error loss for regression, cross-entropy loss for classification, etc.), PF is the point fidelity metric (3) and $\alpha \in (0, 1)$ is a hyper-parameter that controls the relative weight of the two loss functions (Fig. 2). The optimization problem in (7) can be solved using a gradient-based optimization algorithm. The obtained surrogate model $\hat{g}_q$ can be used as a global explanation method regarding the obtained $\hat{f}_\theta$.

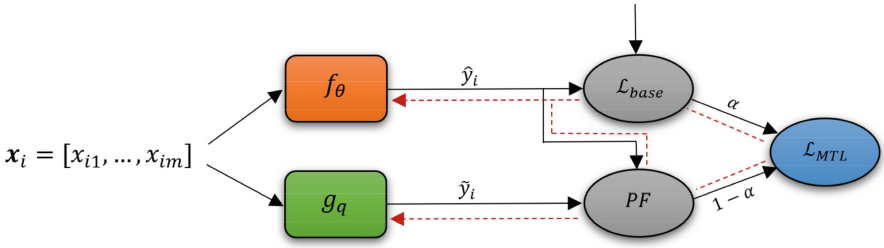

Fig. 2. The proposed MTL framework. We represent a data point as a feature vector $\mathbf{x}_i$ with f_θ and g_q being the black-box and explainable models respectively. Ground-truth response is denoted by $\mathbf{y}_i$, while the black-box's and the linear model's predictions are denoted by $\hat{\mathbf{y}}_i$ and $\tilde{\mathbf{y}}_i$ respectively. Red dashed lines denote the back-propagated gradients which allow the information exchange between the two tasks via the joint optimization of the parameter sets θ and q. (Color figure online)

4 Experimental Results

This section provides results and insights from the experiments that we carried out in order to assess the performance of the MTL-based framework and compare it with state-of-the-art, single-task (STL) approaches. We experimented with global and local explainability performance metrics. For simplicity, we considered experiments on tabular datasets in which attribution is directly awarded on the input features without further processing (e.g., formation of super-pixels for imaging data [6]).

4.1 Model Architectures and Training

For the black-box f_θ, we experimented with Multi-Layer Perceptrons (MLPs). We acquired the final architecture through a tuning process in which the number of hidden layers as well as the number of neurons per layer were selected based on the performance in a held-out validation set. We set *ReLU* [19] as the activation function of the hidden layers. For training, we used SGD with Adam [17] and starting learning rate $\eta = 10^{-3}$. Additionally, we used the *binary cross-entropy* loss for binary classification tasks, the *logarithm of the hyperbolic cosine* for regression tasks and an early stopping criterion. For the MTL paradigm, a linear model was used for g_q.

4.2 Datasets

We tested our models on a variety of regression and binary classification problems from the UCI database [20], the California Housing dataset[1] [27] and the Titanic dataset[2] [32]. Information about characteristics of these datasets can be found

[1] https://www.dcc.fc.up.pt/~ltorgo/Regression/cal_housing.html.
[2] https://www.openml.org/search?type=data&sort=runs&id=40945&status=active.

in Table 1. For each dataset, we standardized numerical features to have mean zero and variance one.

Table 1. Statistics of the datasets.

Dataset	# samples	# features	Type
(Red) Wine Quality [16]	1,599	12	regression
Adult [26]	48,842	14	classification
California Housing [27]	20,640	8	regression
Titanic [32]	1,309	14	classification
AutoMPG [28]	398	7	regression

4.3 Evaluation Measures

For the prediction tasks, we relied on traditional metrics such as *Accuracy* and the F_1 *score* for classification, and *Mean Squared Error (MSE)* for regression, in order to measure the predictive performance of the models. For the explainability task, we used the GF and GNF metrics, defined in (4) and (6), in the experiments regarding global and local explainability respectively.

4.4 Global Explainability Evaluation

Our method provides global explanations through the coefficients of $\hat{g}_q$ in the form of feature importance scores. We compared the models trained in the MTL fashion to the ones obtained using separate, single-task training. For the single-task scenario, we used a global surrogate model to approximate the single-task trained model *after the end of its training*. For classification tasks, the comparison in predictive performance is made based on *Accuracy*, while in regression tasks, *MSE* is used. Table 3 shows the results of the experiments on the test set of each dataset. For α, we experimented with $step = 0.1$ in the range $(0, 1)$, resulting in 9 values. Additionally, for the sake of completeness, we present prediction test scores from a linear model baseline trained with STL in Table 2 in order to justify the use of a non-linear black-box model.

The results show that training by using the MTL setting improves the GF metric. Lower GF is better as it measures the difference of predictions. The improvement holds for all values of α, but especially for the lower values of α it does so by a large margin, compared to STL. This is expected, since for low values of α, the Fidelity loss component has a large coefficient, and the optimization process is highly influenced by it. However, for low values of α, we see that the predictive performance of $\hat{f}_\theta$ decreases only by a small margin. This effect diminishes as α takes on higher values, but so does the margin of the decrease of GF, compared to the STL baseline. This is also anticipated as

Table 2. Comparison of single-task trained MLP and linear models.

Metric	Dataset	Linear	Non-linear (MLP)
ACCURACY/MSE	WINE (MSE)	0.598	0.541
	ADULT (ACC.)	0.824	0.850
	HOUSING (MSE)	0.410	0.237
	TITANIC (ACC.)	0.774	0.785
	AUTOMPG (MSE)	0.176	0.098

Table 3. Comparison of a single-task trained MLP model (STL) with MTL training for various values of α based on the corresponding metric for the predictive task performance and GF for the global explainability task. Results are shown across 5 runs.

Metrics	Datasets	STL	**MTL - parameter α**								
		-	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
ACCURACY/ MSE	WINE (MSE)	0.541	0.569	0.558	0.544	0.544	0.540	0.539	0.536	0.540	0.547
	ADULT (ACC.)	0.850	0.836	0.839	0.842	0.844	0.848	0.849	0.850	0.849	0.850
	HOUSING (MSE)	0.237	0.403	0.381	0.381	0.340	0.307	0.279	0.262	0.221	0.204
	TITANIC (ACC.)	0.785	0.764	0.767	0.775	0.781	0.776	0.776	0.781	0.780	0.776
	AUTOMPG (MSE)	0.098	0.153	0.148	0.137	0.126	0.117	0.110	0.104	0.096	0.105
Global Fidelity (GF)	WINE	0.034	0.001	0.003	0.005	0.009	0.014	0.025	0.036	0.056	0.086
	ADULT	0.033	0.001	0.004	0.007	0.010	0.013	0.016	0.018	0.021	0.021
	HOUSING	0.199	0.0006	0.002	0.006	0.015	0.025	0.038	0.056	0.100	0.152
	TITANIC	0.048	0.001	0.004	0.007	0.011	0.017	0.020	0.026	0.026	0.028
	AUTOMPG	0.093	0.001	0.001	0.004	0.008	0.013	0.024	0.039	0.055	0.083

a higher weight for the predictive loss allows it to affect training to a greater extent and thus increase the predictive performance.

This accuracy-interpretability trade-off for the different values of α is depicted in Fig. 3 for each dataset. The larger sized (circled) points represent the Pareto optimal points (i.e., the optimal trade-offs between the two tasks). The behavior is consistent for all the datasets where a monotonicity of GF is observed, except for the Wine Quality dataset where the fidelity metric is slightly worse than the single-task baseline for large values of α (e.g., $0.7, 0.8, 0.9$). This could be explained by the fact that we treat the target variable of the dataset as continuous, thus solving a regression problem. It could be possible that since the linear model cannot predict the target as accurately as the neural network model, and since for large values of α the Fidelity component takes a small weight in the loss function, the result of the approximation is less accurate.

4.5 Local Explainability Evaluation

We additionally experimented with local explainability, in order to assess if the acquired black-box $\hat{f}_\theta$ could be better explained by local surrogate models. We used a post-hoc local explainability method and specifically, LIME [6]. We evaluated the explanations produced by LIME based on the acquired black-box $\hat{f}_\theta$

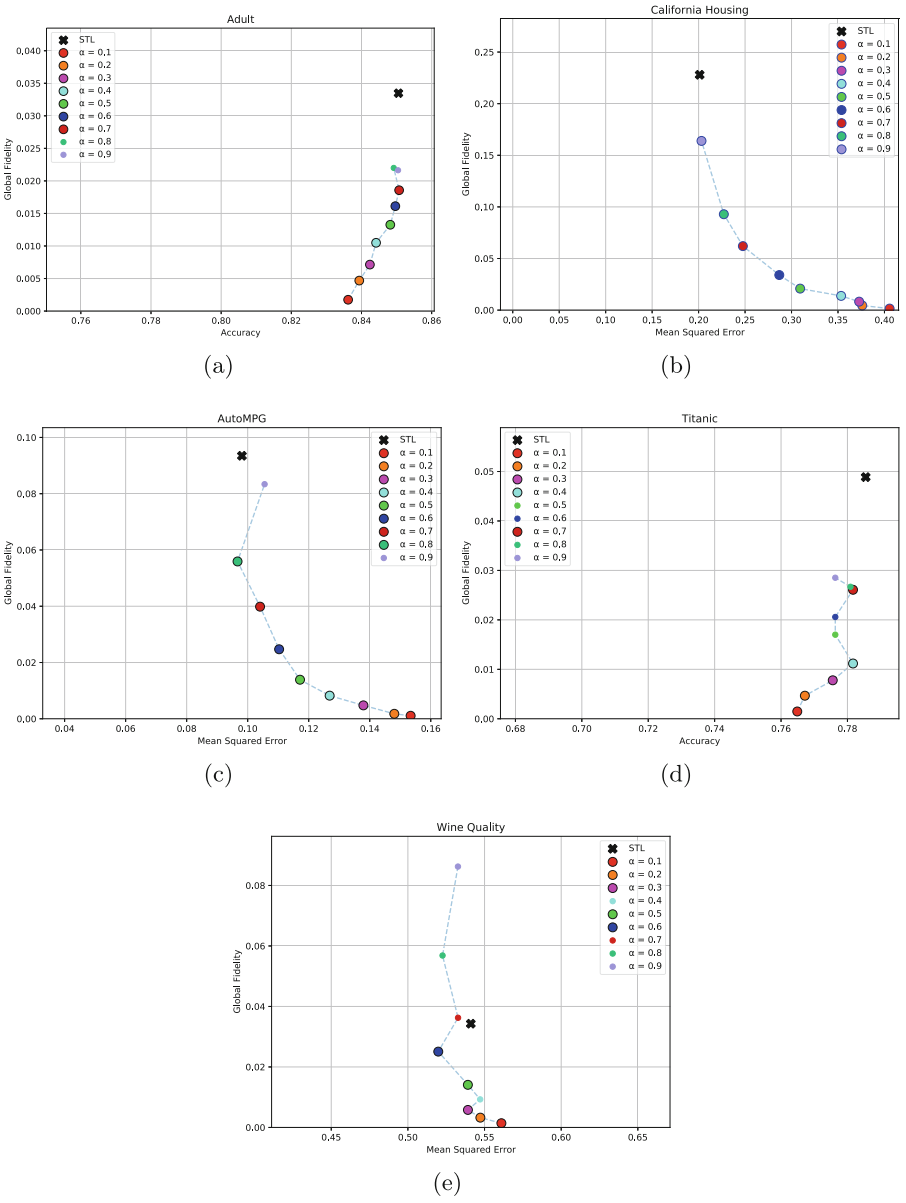

Fig. 3. Visualization of the predictivity-explainability trade-off. Prediction accuracy vs. Global Fidelity results for different values of α on different datasets. Datasets: (a) Adult, (b) California Housing (c) AutoMPG, (d) Titanic, (e) Wine Quality.

using the GNF metric. We again compared a single-task trained black-box model against black-box models trained with MTL ($\alpha \in (0, 1), step = 0.1$).

After the training procedure of $\hat{f}_\theta$ was completed, we used LIME to produce local explanations for each instance in the test set. For the GNF metric, we generated neighbors for $N_\mathbf{x}$ using perturbations stemmed from $\mathcal{N}(\mathbf{x}, \mu, \sigma^2)$ with $\mu = 0, \sigma^2 = 0.1$ and used 10 neighbors ($|N_\mathbf{x}| = 10$) for the evaluation.

Table 4 contains the results of the experiments for all datasets.

Table 4. Comparison of a single-task trained (STL) MLP model with MTL training for various values of α based on the corresponding metric for the predictive task performance and GNF for the local explainability task. Because calculation of GNF is slow due to a separate training of a surrogate model for each instance, results are shown for a single run. In addition, for the ADULT and HOUSING datasets, 500 test points were used.

Metrics	Datasets	STL	MTL - parameter α								
		-	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
ACCURACY/ MSE	WINE (MSE)	0.541	0.584	0.557	0.545	0.537	0.551	0.529	0.518	0.509	0.540
	ADULT (ACC.)	0.850	0.834	0.838	0.842	0.843	0.845	0.850	0.852	0.851	0.852
	HOUSING (MSE)	0.237	0.403	0.391	0.355	0.334	0.348	0.264	0.251	0.234	0.195
	TITANIC (ACC.)	0.785	0.778	0.774	0.774	0.770	0.774	0.767	0.782	0.774	0.771
	AUTOMPG (MSE)	0.098	0.156	0.141	0.135	0.123	0.118	0.113	0.103	0.104	0.112
Global Neighborhood Fidelity (GNF)	WINE	0.019	0.001	0.002	0.003	0.008	0.008	0.014	0.031	0.018	0.029
	ADULT	0.084	0.048	0.047	0.057	0.061	0.067	0.078	0.074	0.051	0.083
	HOUSING	1.260	0.003	0.052	0.085	0.134	0.369	0.230	0.937	0.242	0.616
	TITANIC	0.131	0.048	0.120	0.140	0.164	0.119	0.009	0.153	0.225	0.105
	AUTOMPG	0.111	0.027	0.039	0.016	0.022	0.035	0.033	0.041	0.047	0.126

Results show that GNF is also improved when MTL is employed. This shows that the acquired black-box model $\hat{f}_\theta$ which was trained with regard to having similar predictions to those of a linear model $\hat{g}_q$ can also be more accurately approximated by *local* linear explanations. However, local explainability results seems to be independent regarding the value of α which could be explained by the fact that the objective (7) does not involve a local explainability optimization component. A possible solution would be the incorporation of a component similar to [1] that will also account for local explainability performance during the training process.

4.6 Lessons Learned from the Experiments

Overall, our results showcase that using the proposed MTL training procedure allows the surrogate linear model $\hat{g}_q$ to better approximate the black-box model $\hat{f}_\theta$, compared to the standard baseline of training them sequentially and separately. We also appose Table 5 which contains the R^2 score between the predictions of $\hat{f}_\theta$ and $\hat{g}_q$ in the single-task and multi-task settings on the ADULT dataset.

Table 5. R^2 score between the predictions of the black-box and the surrogate models on ADULT, in single-task and multi-task settings.

Approach	STL	$\alpha = 0.1$	$\alpha = 0.2$	$\alpha = 0.3$	$\alpha = 0.4$	$\alpha = 0.5$	$\alpha = 0.6$	$\alpha = 0.7$	$\alpha = 0.8$	$\alpha = 0.9$
$\mathbf{R^2}$	0.57	0.97	0.93	0.89	0.85	0.83	0.78	0.77	0.72	0.73

The following key points can be observed from our experiments:

- The produced global and local explanations are more accurate than the explanations produced by the single-task trained black-box model. This means that $\hat{f}_\theta$ can be more accurately approximated even from local explainability methods compared to a black-box trained with STL.
- For global explainability, we observe a high improvement in the Global Fidelity metric for low values of α and a slight decrease in the predictive performance of $\hat{f}_\theta$, compared to the baseline of the single-task training. The decrease diminishes as α gets larger and *even disappears on certain datasets*.
- For local explainability, we also observe an improvement on the Fidelity of the local explanations produced by LIME [6], compared to the Fidelity of the same explanations when the black-box neural network is trained in a traditional single-task fashion, but the improvement seems to be independent of the value of α. This could be explained by the fact that the optimization objective manages to make $\hat{f}_\theta$ more 'interpretable' but does not account for local explainability performance per se.

5 Conclusions

In this work, we propose and evaluate a novel Multi-Task Learning framework in which we train a black-box neural network model together with a surrogate linear model in order to obtain Feature Attribution explanations. We use a convex combination of two loss components. The first component assesses the black-box's predictive performance in terms of a training loss function, while the second one evaluates the surrogate's approximation quality through the fidelity metric. We demonstrate that this paradigm improves the quality of the surrogate model's approximation to the black-box, thus resulting in more accurate (fidelity-wise) global explanations on unseen test data compared to the standard used method, which is to train the surrogate model separately from, rather than concurrently with the black-box one. Finally, we also showcase the effectiveness of the framework on a local explainability setting where again, more accurate (fidelity-wise) local explanations are produced.

Future work could generalize the current setting through more explainability metrics such as faithfulness, complexity [9] and stability [1] to the training procedure. We could also consider other forms of optimization like constrained optimization, namely minimize the prediction accuracy subject to a constraint on an explainability metric. The objective would be to optimize the predictive

training loss while enforcing a constraint on the value taken by the fidelity metric in order to keep it below a desired threshold.

Lastly, an area we would like to study is related to user-perception based explainability metrics. In the current work, we use a quantitative metric for explainability, however, the real perceived experience on the end-user is not clear. As explainability of ML models touches upon the end-users more than any other ML model property, the grand objective would be to translate metrics such as fidelity to new ones that are closer to the user perception of what explainability means to them and how it is perceived, and at the same time continue to follow a systematic optimization approach, similar to what we describe in this paper. This of course necessitates that the new metrics are differentiable or can be approximated by differentiable functions, so that they can be incorporated in a Deep Learning-based framework. Learning this mapping from the set of quantitative explainability metrics such as fidelity, faithfulness, complexity, to perceived user experience is a challenging goal which calls for ML methods on crowdsourced datasets collected from human feedback that we intend to pursue in the future.

Acknowledgements. This work was supported by the CHIST-ERA grant CHIST-ERA-18-SDCDN-004 (project LeadingEdge, grant number T11EPA4- 00056) through the General Secretariat for Research and Innovation (GSRI). It was also supported by the Horizon Europe PRE-ACT project, supported by the European Commission through the Horizon Europe Program (Grant Agreement number 101057746), by the Swiss State Secretariat for Education, Research and Innovation (SERI) under contract number 22 00058, and by the UK government (Innovate UK application number 10061955).

References

1. Plumb, G., Al-Shedivat, M., Cabrera, Á. A., Perer, A., Xing, E., Talwalkar, A.: Regularizing black-box models for improved interpretability. Adv. Neural Inf. Process. Syst. **33**, 10526–10536 (2020). Curran Associates Inc.
2. Shrikumar, A., Greenside, P., Kundaje, A.: Learning important features through propagating activation differences. In: Proceedings of the 34th International Conference on Machine Learning (ICML 2017), vol. 70, pp. 3145–3153 (2017)
3. Zhou, J., Gandomi, A.H., Chen, F., Holzinger, A.: Evaluating the quality of machine learning explanations: a survey on methods and metrics. Electronics **10**(5) (2021)
4. Rieger, L., Singh, C., Murdoch, W., Yu, B.: Interpretations are useful: penalizing explanations to align neural networks with prior knowledge. In: Proceedings of the 37th International Conference on Machine Learning, volume 119 of Proceedings of Machine Learning Research, pp. 8116–8126 (2020)
5. Molnar, C.: Interpretable Machine Learning, 2nd edn. (2022)
6. Ribeiro, M.T., Singh, S., 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, San Francisco, 13–17 August 2016, pp. 1135–1144 (2016)

7. Wachter, S., Mittelstadt, B., Russell, C.: Counterfactual explanations without opening the black box: automated decisions and the GDPR. Harvard J. Law Technol. **2**(31), 841–887 (2018)
8. Chen, Z., et al.: Co-attentive multi-task learning for explainable recommendation. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI 2019), pp. 2137–2143. International Joint Conferences on Artificial Intelligence Organization (2019)
9. Bhatt, U., Weller, A., Moura, J.M.F.: Evaluating and aggregating feature-based model explanations. In: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI 2020) (2020)
10. Burkart, N., Huber, M.F.: A survey on the explainability of supervised machine learning. J. Artif. Int. Res. **70**, 245–317 (2021)
11. Ross, A.S., Hughes, M.C., Doshi-Velez, F.: Right for the right reasons: training differentiable models by constraining their explanations. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI 2017), Melbourne, pp. 2662–2670 (2017)
12. Wu, M., Hughes, M.C., Parbhoo, S., Zazzi, M., Roth, V., Doshi-Velez, F.: Beyond sparsity: tree regularization of deep models for interpretability. In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence, AAAI 2018/IAAI 2018/EAAI 2018 (2018)
13. Wu, M., Parbhoo, S., Hughes, M., Kindle, R., Celi, L., Zazzi, M., Roth, V., Doshi-Velez, F.: Regional tree regularization for interpretability in deep neural networks. Proc. AAAI Conf. Artif. Intell. **34**(04), 6413–6421 (2020)
14. Ma, J., Zhao, Z., Yi, X., Chen, J., Hong, L., Chi, E.H.: Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2018), 19–23 August 2018, pp. 1930–1939. ACM, London (2018)
15. Belle, V.I., Papantonis, I.: Principles and practice of explainable machine learning. Front. Big Data **4** (2021)
16. Cortez, P., Cerdeira, A., Almeida, F., Matos, T., Reis, J.: Wine Quality. UCI Machine Learning Repository (2009)
17. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: 3rd International Conference on Learning Representations (ICLR 2015), San Diego, 7–9 May 2015, Conference Track Proceedings (2015)
18. Rumelhart, D., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. Nature **323**, 533–536 (1986)
19. Fukushima, K.: Visual feature extraction by a multilayered network of analog threshold elements. IEEE Trans. Syst. Sci. Cybernet. **5**(4), 322–333 (1969)
20. Kelly, M., Longjohn, R., Nottingham, K.: The UCI Machine Learning Repository. https://archive.ics.uci.edu. Accessed June 2023
21. Vandenhende, S., et al.: Multi-task learning for dense prediction tasks: a survey. IEEE Trans. Pattern Anal. Mach. Intell. **44**(07), 3614–3633 (2022)
22. Misra, I., Shrivastava A., Gupta, A., Hebert, M.: Cross-stitch networks for multi-task learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
23. Amparore, E.G., Perotti, A., Bajardi, P.: To trust or not to trust an explanation: using LEAF to evaluate local linear XAI methods. PeerJ Comput. Sci. **7**, e479 (2021)

24. Belém, C., Balayan, V., Saleiro, P., Bizarro, P.: Weakly supervised multi-task learning for concept-based explainability. In: Proceedings of the 1st Workshop on Weakly Supervised Learning (WeaSuL) - 38th International Conference on Machine Learning (ICML), Online (2021)
25. Sundararajan, M., Taly, A., Yan, Q.: Axiomatic attribution for deep networks. In: Proceedings of the 34th International Conference on Machine Learning (ICML 2017), vol. 70, pp. 3319–3328 (2017)
26. Becker, B., Kohavi, R.: Adult. UCI Machine Learning Repository (1996)
27. Pace, K., Barry, R.: Sparse spatial autoregressions. Statist. Prob. Lett. **33**(3), 291–297 (1997)
28. Quinlan, R.: Auto MPG. UCI Machine Learning Repository (1993)
29. Lundberg, S., Lee, S.: A unified approach to interpreting model predictions. Adv. Neural Inf. Process. Syst. **30**, 4765–4774 (2017)
30. Weinberger, E., Janizek, J., Lee, S.: Learning deep attribution priors based on prior knowledge. Adv. Neural Inf. Process. Syst. **33**, 14034–14045 (2020)
31. Alvarez-Melis, D., Jaakkola, T.: Towards robust interpretability with self-explaining neural networks. Adv. Neural Inf. Process. Syst. **31** (2018)
32. Harrell Jr., F.E., Cason, T.: Titanic dataset. https://www.openml.org/d/40945 (2017)
33. Lee, G., Jin, W., Alvarez-Melis, D., Jaakkola, T.: Functional transparency for structured data: a game-theoretic approach. In: Proceedings of the 36th International Conference on Machine Learning, Volume 97 of Proceedings of Machine Learning Research, pp. 3723–3733 (2019)
34. Mitchell, T.N.: Machine Learning, 1st edn. McGraw-Hill Inc., USA (1997)
35. Štrumbelj, E., Kononenko, I.: Explaining prediction models and individual predictions with feature contributions. Knowl. Inf. Syst. **41**(3), 647–665 (2014)
36. Du, M., Liu, N., Yang, F., Hu, X.: Learning credible deep neural networks with rationale regularization. In: 2019 IEEE International Conference on Data Mining (ICDM), Los Alamitos, pp. 150–159 (2019)
37. Wang, N., Wang, H., Jia, Y., Yin, Y.: Explainable recommendation via multi-task learning in opinionated text data. In: 41st International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 165–174. Association for Computing Machinery (2018)