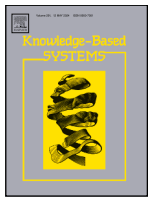
Knowledge-Based Systems 329 (2025) 114384

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys



An explainable multi-task similarity measure: Integrating accumulated local effects and weighted Fréchet distance

Pablo Hidalgo [a,b,*], Daniel Rodriguez [a]

[a] Department of Computer Science, University of Alcala, 28805, Alcalá de Henares, Madrid, Spain
[b] Faculty of Law, Business and Government, Universidad Francisco de Vitoria, Ctra. Pozuelo-Majadahonda Km 1,800, 28223, Pozuelo de Alarcón, Madrid, Spain

ARTICLE INFO

Keywords:
MTL
XAI
ALE
Multi-task similarity
Fréchet distance

ABSTRACT

In many machine learning contexts, tasks are often treated as interconnected components with the goal of leveraging knowledge transfer between them, which is the central aim of Multi-Task Learning (MTL). Consequently, this multi-task scenario requires addressing critical questions: which tasks are similar, and how and why do they exhibit similarity? In this work, we propose a multi-task similarity measure based on Explainable Artificial Intelligence (XAI) techniques, specifically Accumulated Local Effects (ALE) curves.

ALE curves are compared using the Fréchet distance, weighted by the data distribution, and the resulting similarity measure incorporates the importance of each feature. The measure is applicable in both single-task learning scenarios, where each task is trained separately, and multi-task learning scenarios, where all tasks are learned simultaneously. The measure is model-agnostic, allowing the use of different machine learning models across tasks. A scaling factor is introduced to account for differences in predictive performance across tasks, and several recommendations are provided for applying the measure in complex scenarios.

We validate this measure using four datasets, one synthetic dataset and three real-world datasets. The real-world datasets include a well-known Parkinson's dataset and a bike-sharing usage dataset — both structured in tabular format — as well as the CelebA dataset, which is used to evaluate the application of concept bottleneck encoders in a multitask learning setting. The results demonstrate that the measure aligns with intuitive expectations of task similarity across both tabular and non-tabular data, making it a valuable tool for exploring relationships between tasks and supporting informed decision-making.

1. Introduction

In a complex world, addressing tasks as interconnected elements rather than isolated parts makes intuitive sense, especially in a multi-task scenario. This paper specifically focuses on artificial intelligence tasks that can be learned by machine learning models from two perspectives: (i) in a single-task manner, where each task is learned independently by its own model, which may vary across tasks, or (ii) in a multi-task manner, where tasks are learned jointly, leveraging shared information and transferring knowledge between tasks to achieve better results [1,2].

Whether employing single-task or multi-task learning, some critical questions arise depending on the context: (i) which tasks are similar, (ii) to what extent are they similar, and (iii) why are they similar?

Identifying relationships between tasks is highly relevant in various contexts, including cybersecurity [3], manufacturing [4], and healthcare [5]. Understanding task similarities is crucial for determining whether similar tasks exhibit analogous behaviors in actionable contexts. For example, two diseases can be treated as distinct tasks, but their similarity might suggest that they respond similarly to a specific treatment. To leverage such relationships, it is essential not only to identify similar tasks but also to explain the basis of their similarity.

Current multi-task learning techniques compute task similarity in an abstract manner, which may introduce biases in the learning algorithm. For instance, soft parameter-sharing techniques in deep learning impose penalties on the loss function using metrics such as the Frobenius norm [6]. While these approaches often achieve more accurate models in terms of loss minimization, they lack interpretability regarding how and why tasks are similar.

Explainable Artificial Intelligence (XAI) [7] has been a critical tool in understanding the inner workings of black-box models, providing transparency in predictions and enhancing trust. However, to the best of our knowledge, current explainable techniques operate only in single-task scenarios, focusing solely on understanding how a single model works [8].

* Corresponding author.
E-mail address: pablo.hidalgo@ufv.es (P. Hidalgo).

https://doi.org/10.1016/j.knosys.2025.114384
Received 29 January 2025; Received in revised form 9 August 2025; Accepted 29 August 2025
Available online 1 September 2025

The aim of this paper is to propose a *multi-task similarity measure to identify and explain task similarities in an interpretable manner*. The computation of this measure is agnostic to whether tasks were trained using single-task or multi-task learning; it only requires the trained models for each task. Importantly, the measure evaluates the similarities between the models of the tasks, not directly on the datasets. This approach enables us to analyze the mechanisms of the models and their relationships. Since models are simplifications of the underlying complexities of tasks, ensuring sufficient model quality is crucial.

This work makes the following contributions to achieve this objective:

- We define a similarity measure between tasks from an interpretability perspective.
- The similarity measure introduces weights that account for the importance of each variable in a task and the reliability of each segment of the data.

In this work, the chosen interpretability method is Accumulated Local Effects (ALE) curves [9]. ALE curves capture the average influence of each feature on predictions and offer advantages over other explainability tools, such as Partial Dependence Plots (PDPs) [10]. However, other methods can also be easily incorporated into the measure.

To quantify the similarity between tasks, we develop a weighted modification of the Fréchet Distance [11], a standard measure for computing the distance between curves. Furthermore, we propose a scaling factor to modify the similarity to account for differences in predictive performance across task, and several recommendations are provided for applying the measure in complex scenarios such as dealing with non-tabular data (e.g., images) or heterogeneous feature spaces across tasks. These guidelines help ensure the robustness and interpretability of the similarity measure under a wide range of real-world conditions.

The rest of the paper is organized as follows. Section 2 reviews the background and related work, covering key concepts such as multi-task learning, explainable artificial intelligence, and the Fréchet distance. In Section 3, we provide the formal definition of the proposed multi-task similarity measure. Section 4 applies the measure to four datasets: a synthetic dataset, a Parkinson's patient dataset, a bike-sharing usage dataset, and an image dataset (CelebA). Section 5 discusses the main properties, strengths, and weaknesses of the measure. Finally, Section 6 presents the conclusions and outlines future directions for this work.

2. Background and related work

2.1. Multi-task learning

The Multitask Learning (MTL) paradigm [1] is an active research area that aims to leverage shared knowledge between tasks to alleviate data sparsity and reduce learning errors. The main hypothesis of MTL is that simultaneously learning multiple related tasks can enhance the generalization performance across all considered tasks [2]. A critical aspect involves determining which tasks are similar to effectively transfer knowledge between them. Otherwise, multi-task learning may not function optimally, as knowledge transfer from unrelated tasks can detrimentally impact learning and worsen single-task learning.

State-of-the-art techniques in MTL can be broadly categorized into hard parameter sharing, soft parameter sharing, and task-specific modules [6].

Hard parameter sharing approaches, such as those used in early neural networks, involve sharing a common set of parameters between all tasks, thereby reducing the risk of overfitting but limiting task-specific flexibility.

In contrast, soft parameter sharing techniques, such as those based on matrix factorization or attention mechanisms, maintain task-specific parameters while imposing regularization constraints to encourage similarity [2]. Recent advances have introduced novel architectures such as the multigate mixture-of-experts (MMoE), which dynamically allocate resources to tasks and achieve state-of-the-art results in applications ranging from natural language processing to computer vision [12]. Transformer-based models, which leverage attention mechanisms to model inter-task relationships, have also demonstrated superior performance in multi-task learning [13].

Despite their success, a persistent challenge in MTL is understanding and quantifying task relationships, as most methods rely on abstract or implicit measures of similarity, limiting interpretability.

2.2. Explainable AI

The field of Explainable Artificial Intelligence (xAI) [8] has gained substantial attention recently as the deployment of complex machine learning models in real-world applications has become more prevalent in domains such as the medical domain, where explainability is a must. Understanding and interpreting the decisions made by these complex models is crucial for building trust, ensuring accountability, and meeting ethical standards. This requires the development of techniques capable of explaining the diverse range of models employed in various contexts. Furthermore, explainability can be a crucial tool in the refinement of the process of developing a machine learning model [14].

There exist diverse types of explainable methods, each exploiting certain properties of the models. Local methods aim to explain individual predictions or a small subset of data such as Individual Conditional Expectation (ICE) [15], Local Interpretable Model-agnostic Explanations (LIME) [16] or SHapley Additive exPlanations (SHAP) [17]. Global methods, however, try to explain the behavior of the entire model and provide a general picture of the overall trends. An important global model-agnostic method is the ALE plots [9], which allows for an understanding of how one or two features influence the prediction on average. This method is unbiased and less computationally expensive than other alternatives, such as PDP [10]. In this work, the multi-task similarity measure is developed using ALE as the foundation to compute the similarity between features across different tasks.

However, current xAI techniques operate in a single-task manner and are designed to understand only the behavior of individual models and their variables. As a result, they do not effectively address the multi-task paradigm or facilitate an understanding of inter-task relations in order to explain the "big picture".

2.3. Similarity measures

A similarity measure can be defined as a function that assesses the degree of similarity or the relationship between two entities [18]. In this work, the considered entities are tasks represented as mathematical models resulting from a machine learning process. There are many definitions of similarity, depending on the nature of the context at hand and the properties that we want the similarity to satisfy [19]. The Euclidean distance and cosine similarity are two classical measures that capture the relationship between vectors, and they have a broad application in many fields. In natural language processing and bioinformatics, edit distance and string kernels have been essential for measuring the similarity between sequences [20,21]. In statistics, the Kullback-Leibler divergence [22] is an important measure that works with two probability distributions, and it quantifies the differences between them.

In this work, each task is represented by its ALE curves for each variable. Therefore, the similarity between tasks must be computed based on these curves. In the literature, there are several measures to compute the similarity between curves.

One important similarity measure between curves is the Hausdorff distance [23] which for arbitrary bounded sets $A, B \subseteq \mathbb{R}^2$ is defined as:

$$\delta_H(A, B) := max\left(\sup_{a\in A}\inf_{b\in B} d(a, b), \sup_{b\in B}\inf_{a\in A} d(a, b)\right)$$

where d is the Euclidean distance. In other words, the Hausdorff distance measures the maximum distance an adversary can force you to

travel by selecting a point in one set, from which you must then travel to the nearest point in the other set. In simpler terms, it represents the greatest distance from a point in one set to the closest point in the other set. This distance has numerous applications, primarily in image comparison, such as in segmentation algorithms for medical images [24] or in computer graphics [25]. However, the Hausdorff distance only considers the sets of points along both curves and does not capture the trajectories of the curves themselves. In many applications, such as the focus of this paper, understanding the course of the ALE curves is crucial as they summarize the behavior of each variable in each task.

The similarity measure proposed in this paper is based on the Fréchet distance [26], which is a measure of similarity between curves that considers both the location and, importantly, the ordering of the points along the curves. If a curve is defined as a continuous mapping $f : [a, b] \to V$ where $a, b \in \mathbb{R}$ and $a \leq b$ and (V, d) is a metric space then, given two curves $f : [a, b] \to V$ and $g : [a', b'] \to V$, the Fréchet distance between them is defined [11,23] as

$$\delta_F(f, g) = \inf_{\alpha,\beta} \max_{t\in[0,1]} d(f(\alpha(t)), g(\beta(t))) \tag{1}$$

where α and β are both arbitrary continuous non-decreasing functions from $[0, 1]$ onto $[a, b]$ and $[a', b']$, respectively. Usually, this distance is intuitively explained as follows: a person walks a dog on a leash, with the person following one curve, while the dog following another. Both are free to adjust their speed, but they cannot backtrack. The Fréchet distance represents the minimum length of the leash required for traversing both curves [11,23].

The exact Fréchet distance between two polygonal curves can be computed in time $\mathcal{O}(pqlog^2pq)$ [23] where p and q are the number of segments on the polygonal curves.

However, if we only focus on the positions of the endpoints of the line segments defining the polygonal curves, we encounter the discrete Fréchet distance, also known as the coupling distance [11]. It can be proved that this distance serves as an upper bound for the (continuous) Fréchet distance, and the difference between them is bounded by the length of the longest edge of the polygonal curve [11]. As this distance forms the basis for the similarity measure defined here, the formal definition of the discrete Fréchet distance is discussed in Section 3.3. An algorithm based on dynamic programming can compute the discrete Fréchet distance in time $\mathcal{O}(pq)$ [11]. Furthermore, in this work it is convenient to use a variant of the discrete Fréchet distance that uses the sum instead of the maximum as exposed by Eiter and Mannila [11].

In our approach, the tasks are simplified and represented as curves, specifically ALE curves, from an explainable standpoint. With a slight adaptation, the discrete Fréchet distance serves as a suitable foundation for calculating the similarity between these curves. Note that the definition of the similarity concept developed here assumes that lower values imply high similarity and vice versa (in other works, this is referred to as dissimilarity).

In this paper, we propose a post-hoc multi-task similarity measure, assuming that tasks have been trained using a single or multi-task learning paradigm. So, this measure, as defined in this work, is not intended to improve model training directly for multi-task learning or, at least, not as the primary goal. Furthermore, the aim is to compute this similarity through explainable techniques.

3. A multitask similarity measure

3.1. Notation and terminology

We consider a set of tasks $\mathcal{T} = \{\mathcal{T}_1, \dots, \mathcal{T}_T\}$. We assume that each task $\mathcal{T}_t \in \mathcal{T}$ has already been trained, resulting in a learned function $f^t(x) \approx E[Y^t|X^t = x]$ where $\mathbf{X}^t = (X_1^t, X_2^t, \dots, X_{d^t}^t)$ is the vector of d^t predictors for task $\mathcal{T}_t$ and $Y^t \in \mathcal{Y} \subseteq \mathbb{R}$ is a response variable. Each task has an associated dataset $\mathcal{D}_t$ consisting of $n_t \in \mathbb{N}$ samples such that $\mathcal{D}_t = \{\mathbf{x}_j^t = (x_{j,1}^t, \dots, x_{j,d^t}^t), y_j^t\}_{j=1}^{n_t}$. It is worth noting that this dataset $\mathcal{D}_t$ may differ from the training dataset used to derive the function $f^t(x)$, and it is used to build the ALE curves and compute the weights of the weighted Fréchet distance.

$\mathcal{FI}^t : \{1, \dots, d\} \to [0, 1]$ refers to the importance of each variable X_j in the task $\mathcal{T}_t$. This value can be calculated in various ways. There are many techniques to calculate feature importance. Some models allow for direct extraction of the importance of predictor variables, such as linear regression or the Random Forest model [27]. There are also model-agnostic techniques, such as Permutation Feature Importance (PFI) [28] or SHAP values [17]. In either case, it should satisfy that $\sum_{j=1}^{d} \mathcal{FI}^t(j) = 1$ for all $\mathcal{T}_t \in \mathcal{T}$. The construction of this function can be derived from any of the available explainable techniques, or it can be created manually. In the latter case, an expert can specify the values to highlight certain features that may be actionable or relevant for the research. Regardless of the approach, the importance values assigned to each variable will significantly influence the measure.

In the simplest case, all features j of each task, i.e., $X_j^1, \dots, X_j^T$ can express the same characteristic, e.g., arterial blood pressure. However, while this simplification aids in computation, it is not a strict requirement, meaning that, for example, variable X_j^t can express the same concept as variable $X_{j'}^{t'}$ with $j \neq j'$, and the measure defined later is ready to accommodate this issue.

For each variable j of each task $\mathcal{T}_t$, we define a partition $\mathcal{P}_j^t := \{\mathcal{P}_j^t(k) = (z_{k-1,j}^t, z_{k,j}^t] : k = 0, 1 \dots, K^t\}$ consisting of $K^t \in \mathbb{N}$ intervals, where $z_{0,j}^t = \min(x_j^t)$ and $z_{K^t,j}^t = \max(x_j^t)$. For any $x \in \mathcal{P}_j^t(k)$, the index of the interval into which x falls is defined by $k_j^t(x) := \{k : x \in \mathcal{P}_j^t(k)\}$ and the number of observations that fall into interval $\mathcal{P}_j^t(k)$ is defined by $n_j^t(k) := |\{\{x_{ij}^t\}_{i=1}^{n_t} : x_{ij}^t \in \mathcal{P}_j^t(k)\}|$. For the purpose of weighting the distance function and because tasks can have a different number of observations, it is convenient to compute the proportion of observations that fall into interval $\mathcal{P}_j^t(k)$ as $p_j^t(k) := \frac{n_j^t(k)}{\sum_{k=0}^{K^t} n_j^t(k)}$.

Finally, the subset of $d - 1$ predictors of task $\mathcal{T}_t$ excluding $\mathbf{x}_j^t$ is defined as $\mathbf{x}_{\setminus j}^t := (x_k^t : k \in \{1, \dots, d\} \setminus \{j\})$.

3.2. Accumulated local effects (ALE)

Our definition of task similarity is based on the main Accumulated Local Effects (ALE) [9]. The main idea behind ALEs is to compute a curve that accumulates the averaged local effects of each variable, considering the values of the other variables.

Although ALEs of second and beyond orders can be computed, in this work, we only consider the main accumulated local effects, i.e., we will compute the ALE curves for each feature separately. The uncentred ALE curve of the variable j and task $\mathcal{T}_t$ is a function

$$\hat{g}_{j,ALE}^t : \mathcal{P}_j^t \to \mathbb{R}$$

such as

$$\hat{g}_{j,ALE}^t((z_{k-1,j}^t, z_{k,j}^t]) = \sum_{p=1}^{k} \frac{1}{n_j^t(p)} h^t(j, p)$$

where

$$h^t(j, p) = \sum_{\{i : x_{i,j}^t \in \mathcal{P}_j^t(p)\}} \{f^t(z_{p,j}, \mathbf{x}_{i,\setminus j}) - f^t(z_{k-1,j}, \mathbf{x}_{i,\setminus j})\}.$$

The (centre) ALE main effect so that the ALE function has a mean of 0 with respect to the marginal distribution is calculated b

$$\hat{f}_{j,ALE}^t(\mathcal{P}_j^t(k)) = \hat{g}_{j,ALE}^t(\mathcal{P}_j^t(k)) - \frac{1}{n_t} \sum_{k=1}^{K^t} n_j^t(k) \hat{g}_{j,ALE}^t(\mathcal{P}_j^t(k)) \tag{2}$$

In the original paper of ALE plots [9], all the examples used the partition $\mathcal{P}_j^t$ based on the quantiles of the empirical distribution. However, in the similarity measure defined here, we prefer to select an equally spaced partition, similar to the standard method to compute a histogram. This choice allows us to capture the shape of the data distribution and determine how to weight the similarity measure.

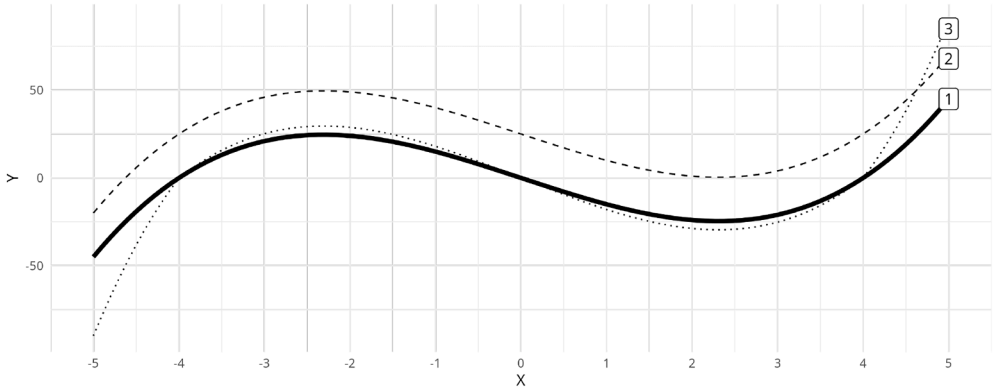


Fig. 1. ALE curves of different features. Intuitively, curves 1 and 3 are more similar than curves 1 and 2.

3.3. Weighted Fréchet distance

The Fréchet distance quantifies the similarity between curves by considering both the positioning and sequence of points along the curves. In our case, the curves are the ALE curves that capture the behavior of the model in each variable.

The estimation of the main ALE curve explained in Section 3.2 is a polygonal curve so, instead of using the definition of the usual Fréchet distance, we employ the discrete version.

In this context, we suppose that (polygonal) ALE curve $\hat{f}^t_{j,ALE}$ is defined as $\sigma(\hat{f}^t_{j,ALE}) = (u_1, \dots, u_{K^t})$ such that $u_k = (z^t_{k,j}, \hat{f}^t_{j,ALE}((z^t_{k-1,j}, z^t_{k,j}])))$.

For the Fréchet distance, we need to define a coupling between two polygonal curves $\hat{f}^t_{j,ALE}$ and $\hat{f}^{t'}_{j',ALE}$ with $t \neq t'$ as a sequence $(u_{a_1}, v_{b1}), (u_{a_2}, v_{b2}), \dots, (u_m, v_m)$ of distinct pairs from $\sigma(\hat{f}^t_{j,ALE}) \times \sigma(\hat{f}^{t'}_{j',ALE})$ such that $a_1 = 1, b_1 = 1, a_m = K^t, b_m = K^{t'}$ and, for all $i = 1, \dots, q$ we have $a_{i+1} = a_i$ or $a_{i+1} = a_i + 1$, and $b_{i+1} = b_i$ or $b_{i+1} = b_i$

The weighted Fréchet Distance is a slight modification of the Discrete Fréchet distance [11]. In our notation, the weighted distance between two ALEs curves of two different tasks is as follows:

$$\delta_F(\hat{g}^t_j, \hat{g}^{t'}_{j'}) := min\{||L|| : L \text{ is a coupling between } \hat{g}^t_j \text{ and } \hat{g}^{t'}_{j'}\} \quad (3)$$

where $||L|| = \sum_{i=1}^{m} w(p^t_{u_{a_i}}, p^{t'}_{v_{b_i}}) d(u_{a_i}, v_{b_i})$, and coupling L is a sequence $(u_{a_1}, v_{b1}), (u_{a_2}, v_{b2})$.

The function d represents a distance function, most commonly the Euclidean distance.

The weighted Fréchet distance can be computed in $\mathcal{O}(pq)$ time [11]. However, if the number of segments is high or varies across tasks, approximation techniques such as Dynamic Time Warping (DTW)[29] or segment aggregation can be used.

This weighted distance can be seen as a similarity between two variables from different tasks. As mentioned in Section 3.1, it is not necessary that the same features occupy the same position in two tasks. In fact, this measure can be applied to two (a priori) unrelated features to identify which feature impacts the response variable similarly. However, if all features are named identically across all tasks (and correspond to the same concept), it simplifies the computation, as we only need to calculate the similarity $d \times (T-1)$ times, as opposed to d^t times (assuming that all tasks have d features).

The typical formulation of the Fréchet distance employs the minimum in Eq. (3), although other alternatives exist that utilizes the sum, as in this work. The rationale for this choice is intuitively explained in Fig. 1. In this figure, curves labeled 1 and 3 are virtually indistinguishable across most of the interval, with any minor differences attributable to the inherent uncertainty of the models. These curves only diverge at the extremes of the domain so they have to be considered as similar. On the other hand, curves 1 and 2, despite exhibiting a similar pattern, are less similar than curves 1 and 3. If we calculate the Fréchet distance between these curves (ignoring the weighted term) using the minimum in Eq. (3), we obtain a value of 45 between curves 1 and 2 as well as between curves 1 and 3, despite the differences exposed previously. However, if we use the sum instead of the minimum, the curves 1 and 2 have a Fréchet distance of 4229.13 and curves 1 and 3 have a distance of 338.13 highlighting the relevance of using the sum in this measure.

To eliminate the artefactual increase in similarity that occurs when ALE curves are discretised at different resolutions, it is necessary to first project every curve onto a common, task-independent grid. Specifically, for each feature, all observations across tasks are pulled to compute the K empirical quantiles $\{q_1, \dots, q_K\}$, and each task's ALE values are linearly interpolated onto these knots. The result is a set of curves with identical abscissae and bin weights, so the weighted Fréchet distance reflects only genuine functional differences rather than arbitrary segment counts. In practice, this resampling step is $O(K)$ per curve and introduces negligible overhead, while ensuring that the multi-task similarity measure remains invariant to uniform curve refinement and numerically stable across datasets.

In the weighted Fréchet distance function, we also introduce a weighted function $w : (0, 1] \times (0, 1] \to [1, \infty)$. The main aim of this function is to serve as a quantifier of the reliability of each segment of the ALE curve because certain segments may have been calculated with very few observations, leading to potentially unimportant or less accurate estimations of the variable of interest. The exact definition of this weighted function can vary, and it has to be related with the context of the environment of the tasks. As a general function, we propose Eq. (4).

$$w(p_{u^t_{a_i}}, p^{t'}_{v_{b_i}}) := \frac{max\{p^t_{u_{a_i}}, p^{t'}_{v_{b_i}}\}}{min\{p^t_{u_{a_i}}, p^{t'}_{v_{b_i}}\}} \quad (4)$$

The purpose of the weighted function is to ponder segments of the two ALE curves that we want to compare, ensuring they have sufficient support and are relevant for the comparison. Upon reviewing Fig. 1 again, it becomes apparent that the differences between curves 1 and 3 at the extremes may be attributed to the uncertainty of the models due to the limited support of observations. Therefore, this region should be assigned less weight in the computation of the similarity measure. Naturally, the function w is symmetrical, i.e., $w(p_{u^t_{a_i}}, p^{t'}_{v_{b_i}}) = w(p^{t'}_{v_{b_i}}, p_{u^t_{a_i}})$

Algorithm 1 shows the Weighted Fréchet distance in pseudo-code.

3.4. Multi-task similarity measure

The weighted Fréchet distance enables us to compare two features from two different tasks. This distance can serve as the foundation for defining a measure that allows us to assess the similarity between two tasks, but before we can formulate the definition of this measure, there are a couple of aspects that we need to consider. On the one hand, one of the primary objectives of this measure is to uncover how specific

Algorithm 1 Weighted discrete Fréchet distance.

Require: Two polygonal ALE curves $\sigma_1 = [(x_1, u_1), \dots, (x_p, u_p)]$ with weights $w_1[1.p]$,
$\sigma_2 = [(y_1, v_1), \dots, (y_q, v_q)]$ with weights $w_2[1.q]$.
Ensure: $\delta_F(\sigma_1, \sigma_2)$.
Initialize $D[1.p][1.q] \leftarrow +\infty$.
for $i = 1$ to p **do**
  for $j = 1$ to q **do**
    $\omega \leftarrow \dfrac{\max(w_1[i],\ w_2[j])}{\min(w_1[i],\ w_2[j])}$
    $d \leftarrow \sqrt{(x_i - y_j)^2 + (u_i - v_j)^2}$
    if $i = 1$ **and** $j = 1$ **then**
      $D[i][j] \leftarrow \omega \cdot d$
    else
      best $\leftarrow \min\big(D[i-1][j],\ D[i][j-1],\ D[i-1][j-1]\big)$
      $D[i][j] \leftarrow$ best $+ \omega \cdot d$
    end if
  end for
end for
return $D[p][q]$

features affect the target similarly. In this context, it is not necessary to compare all the features present in the training dataset for both tasks. Furthermore, there might be features available in one task that are not present in the other task. To address this, the measure can consider only some of the features. Another reason could be that we want to measure the similarity between tasks only for actionable features. On the other hand, generally, we do not assume that the j-th feature in both tasks expresses the same characteristic, i.e., the first feature in one task might represent the blood pressure of the patient, whereas this feature could be the fourth one in another task.

The multi-task similarity measure between a task of interest $\mathcal{T}_t$ and other task $\mathcal{T}_{t'}$ is a function

$$\delta_t : \mathcal{T} \setminus \{\mathcal{T}_t\} \to \mathbb{R}^+$$

is defined as

$$\delta_t(t') := \sum_{j=1}^{d^t} \mathcal{FI}^t(j) \min_{j' \in \{1,\dots,d^{t'}\}} \delta_F(\hat{f}^t_{j,ALE}, \hat{f}^{t'}_{j',ALE}) \tag{5}$$

The pseudocode for the Multi-task Similarity measure is shown in Algorithm 2.

Algorithm 2 Multi-task similarity measure.

Require: Tasks t_0, t_1; feature sets $\mathcal{F}_{t_0}, \mathcal{F}_{t_1}$
ALE curves ALE$[t][f]$, weights prop$[t][f]$, importances imp$[t][f]$.
Ensure: $\delta(t_0, t_1)$.
sim $\leftarrow 0$
for each feature $f \in \mathcal{F}_{t_0}$ **do**
  minDist $\leftarrow +\infty$
  for each feature $g \in \mathcal{F}_{t_1}$ **do**
    $d \leftarrow$ WeightedFrechetDistance$\big($ALE$[t_0][f]$, prop$[t_0][f]$, ALE$[t_1][g]$, prop$[t_1][g]\big)$
    minDist $\leftarrow \min($minDist, $d)$
  end for
  sim $\leftarrow$ sim $+$ imp$[t_0][f] \times$ minDist
end for
return sim

Note that although the weighted Fréchet distance is symmetrical, being Eq. (4) symmetrical, the multi-task similarity measure might not be symmetrical. This is due to the influence of the importance weights in task t, which may not coincide with the feature importance weights of task t'.

3.4.1. Multi-task similarity measure including model performance

The similarity defined in Eq. (5) can be misleading when tasks have very different predictive quality. Such disparities arise when (i) the model assigned to a task lacks sufficient capacity-e.g. using a linear learner for data generated by a nonlinear process-or (ii) some tasks are intrinsically harder.

Throughout the paper we interpret the multi-task similarity measure $\delta_t(t')$ as $\delta_t(t') = 0$ means identical ALE profiles, while larger values indicate tasks that diverge more strongly. This convention matters because the corrective factor introduced below *shrinks* large values when they are attributable to unequal model performance.

Poor performance should not automatically translate into perceived task dissimilarity. To account for this, we multiply the raw discrepancy by a weight $\gamma_t(t') \in [0, 1]$ that depends on the empirical losses $L(t)$ and $L(t')$ (smaller is better):

$$\gamma_t(t') \ = \ f\big(L(t), L(t')\big), \tag{6}$$

with the desideratum that $\gamma_t(t') \approx 1$ when losses are comparable and $\gamma_t(t') \downarrow 0$ as their ratio grows. The *performance-weighted* similarity is then

$$\delta^*_t(t') \ := \ \gamma_t(t') \cdot \delta_t(t'). \tag{7}$$

A simple and scale-free choice that meets the desiderata is

$$\gamma_t(t') \ = \ \frac{\min\{L(t),\ L(t')\}}{\max\{L(t),\ L(t')\} + \varepsilon}, \qquad \varepsilon > 0, \tag{8}$$

where $\varepsilon = 10^{-8}$ (unless stated otherwise) guarantees numerical stability when both tasks fit almost perfectly.

If *both* losses are large, the ratio above is close to 1 even though neither model is trustworthy. We therefore flag any task with $L(t) > \tau$ (e.g., τ could be the median loss across tasks) and recommend inspecting or retraining the corresponding model before drawing conclusions from δ^*_t.

3.4.2. Limitations and recommendations

The Multi-task similarity measure developed in this work has certain limitations. We highlight these limitations here and provide recommendations to help mitigate these issues.

The ALE curves used for the multi-task similarity computations rely on first-order ALE (i.e., they only consider each variable individually). If the true signal is an interaction (e.g., $X_1 \times X_2$), both one-way ALEs can be flat even though the joint surface differs sharply between tasks. The Fréchet distance can be only used between curves parametrized in one dimension (as first-order ALE curves), so the measure cannot be applied to second or higher order ALE curves. In this case, the recommendation is to first learn a shallow autoencoder $Z = f_\theta(X)$ on the union of tasks and compute the ALE curves on Z instead of raw X. The encoder allows for the alignment of heterogeneous feature sets and preserves interactions captured in Z.

However, there is a word of caution regarding this procedure. If the intended purpose is to assess the similarities between tasks in an explainable manner, the output of the encoder may not preserve the interpretability. Nevertheless, the measure values can still be used as a similarity metric between tasks and can serve, for instance, in clustering purposes or as a basis for multi-task learning paradigms.

To address this limitation while preserving interpretability, future work will explore extending the similarity measure to integrate selective second-order effects. One avenue is to compute second-order ALE curves for pre-identified key interaction pairs (via domain knowledge or model-based interaction importance) and aggregate similarity scores from both first-order and selective second-order ALEs.

Another limitation that is important to note is the computational complexity of the measure. There are three main sources contributing to this complexity: (i) the computation of ALE curves, (ii) the calculation of the weighted Fréchet distance between ALE curves, and (iii) the aggregation of similarity scores across all tasks and features. In the worst case, the current implementation scales as $\mathcal{O}(T^2 d K^2)$ where T is

the number of tasks, d the number of features, and K the number of ALE segments.

The first source is related to the size of the data and the speed at which the trained model can produce predictions. A simple optimization is to compute the ALE curves using only a representative sample of the data, which can significantly reduce computational load without severely affecting accuracy.

The second source stems from the Fréchet distance computation between ALE curves, which scales with the number of segments in the curves and the number of pairwise computations. To alleviate this, one could (a) use curves with a reduced or adaptive number of segments, (b) apply approximation techniques or pruning heuristics for the Fréchet distance, or (c) parallelize the pairwise distance calculations.

The third source arises from computing the similarity between each task pair across features. As these computations are independent across task pairs, the most recommended solution is to perform the calculations using parallel processing to take advantage of modern multi-core or distributed computing environments. For a very large number of tasks T, we recommend first clustering tasks with a lightweight metric (e.g., cosine similarity of feature-importance vectors) and computing the full measure only within or between selected clusters, reducing the effective number of pairwise workload from $\mathcal{O}(T^2)$ to roughly $\mathcal{O}(T_c^2)$, where $T_c \ll T$ is the average cluster size.

Although the similarity measure has so far been presented for tabular inputs, many real-world tasks involve non-tabular data, such as image, text, or audio signals. A natural remedy is to use an encoder to transform such data into a latent representation and compute similarity based on these latent features, as proposed at the beginning of this section for heterogeneous task structures. However, this approach introduces a significant limitation: it compromises the interpretability of the similarity measure, as latent features typically lack direct semantic meaning.

To overcome this challenge while preserving compatibility with the ALE Fréchet distance framework, we outline three interpretable transformation strategies that enable the application of our similarity measure to non-tabular data:

1. **Concept-bottleneck encoders.** We propose employing encoders that explicitly predict human-interpretable concepts in their penultimate layer (e.g., edge density, tumor shape, or sentiment polarity) prior to producing the final output as Concept Bottlenecks Models [30]. These concept activations provide a semantically labeled tabular representation on which first-order ALE curves can be computed, thereby retaining the transparency and explainability of the similarity analysis.
2. **Domain-specific feature libraries.** In many fields, there exist well-established libraries of handcrafted, physically meaningful descriptors-such as radiomic features for medical images [31], psycholinguistic or sentiment markers for text [32], and spectral coefficients for audio signals [33]. By extracting these features prior to model training, we avoid reliance on black-box embeddings and ensure that the similarity measure is grounded in features with clear interpretability.
3. **Attention-pooled region features.** Transformer-based architectures naturally provide attention maps that highlight regions (e.g., image patches or text tokens) most influential to a prediction [34]. We propose aggregating these attentions into K coherent superpixels (in vision) or phrase clusters (in NLP), resulting in K interpretable region scores. ALE curves computed on these scores illustrate how specific regions modulate outputs, while weighted Fréchet distances on these curves quantify inter-task differences. The original attention maps can further serve as visual explanations that complement the numerical similarity measure.

Each of these strategies offers a pathway to extend our method to non-tabular data without sacrificing interpretability. Nevertheless, we acknowledge that the transformation step introduces its own assumptions and potential biases, which should be carefully considered in practical applications. We highlight these directions as promising areas for future research, particularly for integrating explainability in complex, multimodal task similarity analysis.

Finally, we acknowledge an inherent limitation in the proposed interpretability framework. The multitask similarity measure is built on Accumulated Local Effects (ALE) curves, which depict how individual features influence model predictions. When the true feature-output relationship is highly non-linear, these curves often exhibit high-frequency oscillations that can overwhelm domain experts and obscure the underlying pattern. We address this concern in two complementary ways. (i) Encoder-based filtering as described earlier in this section, an encoder network can learn a latent representation that captures the salient, shared structure of the ALE curves while discarding task-specific noise. (ii) Spline smoothing of ALE curves. Alternatively, each ALE curve can be regularized directly: we fit a surrogate spline $\hat{g}(x)$ that minimizes a roughness-penalized L_2 loss, thereby preserving the global trend but suppressing idiosyncratic wiggles. Both strategies attenuate fine-grained fluctuations that do not contribute meaningfully to the multitask similarity, yielding explanations that remain faithful yet are considerably easier for experts to interpret.

4. Empirical work

In this section, we apply the similarity measure defined in Section 3.4 in three different contexts.

Firstly, we simulate toy data from various tasks, each with a few features, to comprehend the behavior of the similarity measure. All tasks share features arranged in the same order, making them directly comparable.

Secondly, we apply the multi-task similarity measure to a real data set containing information about Parkinsons' patients. This demonstrate the measure's applicability in a practical context and allow us to discuss its utility in understanding task relationships.

Lastly, we use the multi-task similarity in a bike-sharing dataset from the public service BiciMad in Madrid, Spain. In this case, each task corresponds to one of the 264 stations available to users, with the goal of predicting the number of bikes used per hour based on temporal features (e.g., month, day of the week, hour) and weather conditions (e.g., humidity, temperature). For this dataset, we employ a hybrid parameter-sharing multi-task deep learning algorithm.

In the first dataset, due to its simplicity, we aimed to test whether the tasks intuitively considered similar align with the similarity detected by the measure. In contrast, the other datasets present more complex scenarios, allowing us to demonstrate the full potential of the similarity measure. These cases highlight the benefits and utility of the measure in detecting both general and task-specific behaviors across the datasets.

The details of the model setups can be found respectively in Appendixes A–D. The implementation of the measure is available at github.com/papabloblo/multi-task-similarity.

4.1. Synthetic dataset

The synthetic dataset simulated consists of five tasks, each with five features and $10,000$ observations. For the multitask-measure developed in this work, three key factors must be considered: (1) the distribution of each predictive feature, (2) the relative importance of each feature in the predictive model, and (3) the ALE curves.

Fig. 2 shows the density of each variable across all tasks. It is evident that the tasks exhibit very similar densities for variables X_1, X_2 and X_3, while differences are observed in variables X_4 and X_5. This indicates that even if the patterns of two ALE curves are only similar within a narrow segment, a segment containing the majority of the data points will significantly contribute to a higher similarity. In the definition of the measure, this corresponds to a lower value. The variables X_1 and X_2 are the only predictive variables that are related, following a bivariate

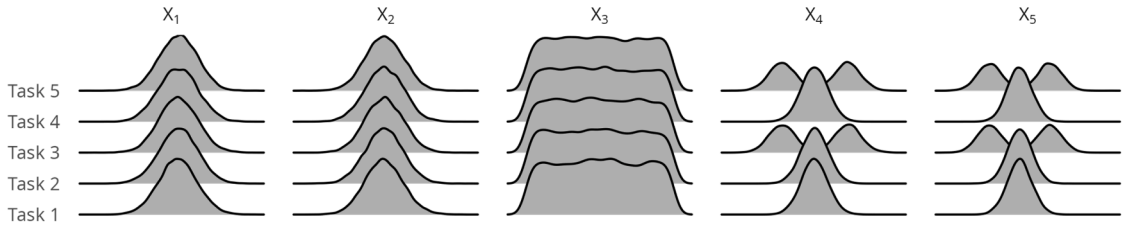


Fig. 2. Density of each predictor variable for each task in Synthetic Dataset 1.

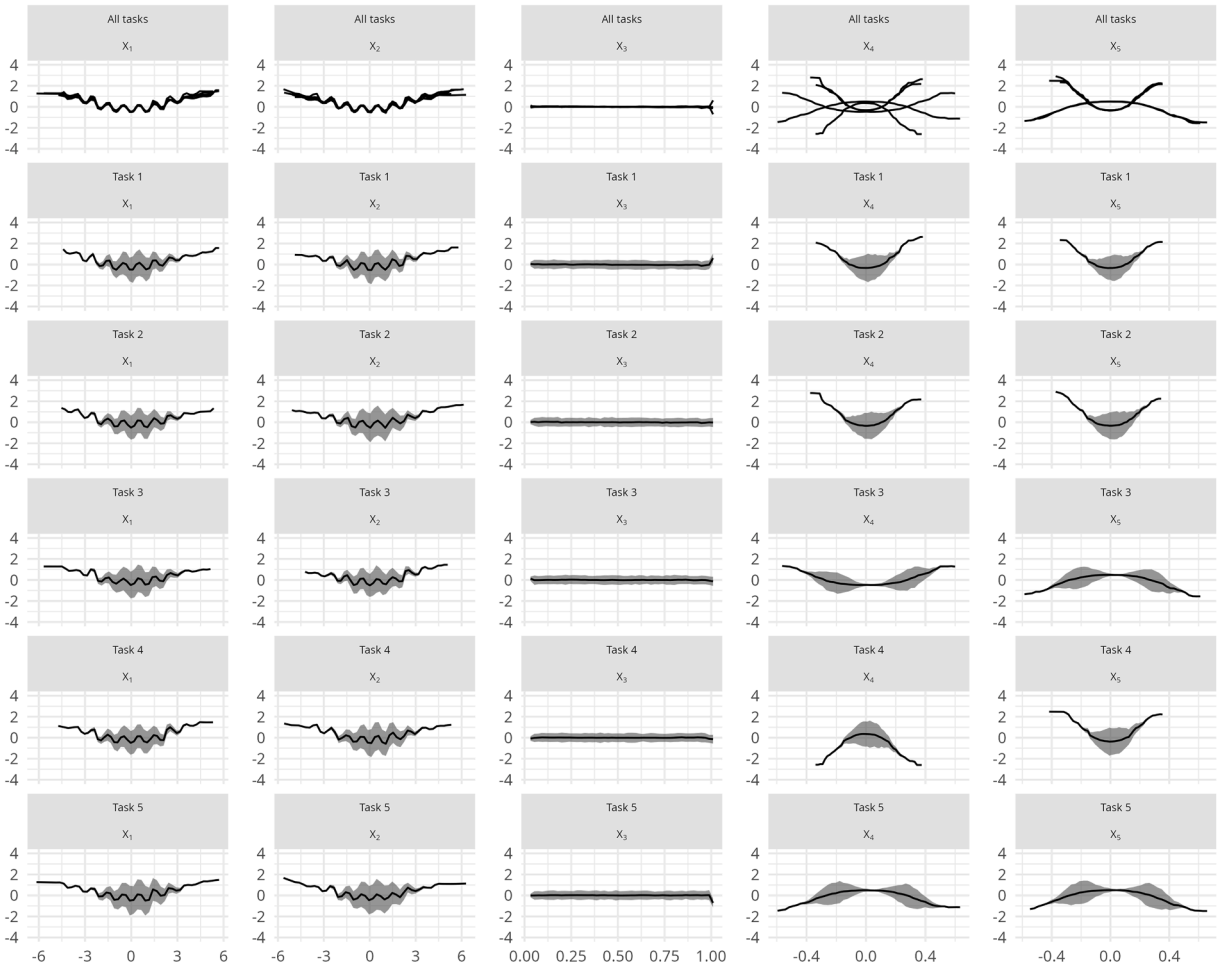


Fig. 3. ALE Curves for the Synthetic Dataset 1. The first row shows all the ALE curves of the different tasks for the same feature. The rest of the rows show ALE curves for each task and feature.

normal distribution, whereas the remaining predictors are unrelated to each other. Details of the data simulation are provided in Appendix A.

Each one of the five tasks was trained independently in a single-task learning fashion using a Random Forest model [27] and the importance measure of each feature was calculated according to the method described by Breiman [27], with the results transformed so that the sum of the feature importance for each of the tasks is 1.

Afterwards, the ALE curve of each feature and task was calculated using an equally spaced partition comprising 50 intervals. Fig. 3 shows the ALE plots for each variable and task, demonstrating similarities in behavior among features across tasks. The marginal distribution of each variable is depicted as the area below the ALE curves. This has a huge impact on the similarity value.

Variables X_1, X_2, and X_3 exhibit nearly identical curves across tasks, albeit for different reasons. In contrast, the variable X_3 exhibits a flat curve across tasks indicating, as previously observed, that this predictor variable is unrelated to the outcome. Finally, features X_4 and X_5 exhibit different behaviors across tasks as illustrated by the ALE plots in Fig. 3.

With these plots, and considering that variables X_1, X_2, and X_3 exhibit similarity across tasks and, therefore, do not contribute to making tasks more or less similar, we can visually determine which tasks are similar. Based on the shape of the curves, it is obvious that tasks 1 and 2 are very similar with respect to all the variables. However, the remaining tasks are only partially similar to each other due to the mixture of shapes in the ALE plots of variables X_4 and X_5, along with the underlying distribution of the predictor variables. This emphasizes the importance of having an objective measure of the similarity between tasks that can consider all the details.

Although the multi-task similarity measure described in this paper provides a value indicating the similarity between tasks, to compute this

value, we first have had to calculate the weighted Fréchet distance between one variable of each task with respect the variables of the other tasks as described in Eq. (3). These intermediate computations help us understand the behavior of the measure and are displayed in Table 1, along with the importance of each variable and the final multi-task similarity measure. It should be noted that, for simplicity, we assumed that all features are named identically across all tasks, and we only considered relations between variables named identically. Consequently, this implies that we only need to calculate the similarity between variables X_1 across tasks, followed by X_2, and so forth. Remember that, by the definition of the measure, lower values indicate that the tasks considered are more similar. A summarize of the similarity between tasks is displayed in Fig. 4.

According to Table 1, variables X_1, X_2, and X_3 exhibit analogous values of similarity across tasks with slight variations. This follows the expected behavior because all the tasks follow the same relationship of X_1 and X_2 to the outcome, and the variable X_3 is unrelated to the outcome. Notice that the weighted Fréchet distance between variables X_3 has lower values than the other variables. This underscores the relevance of considering not only the distance between variables but also the importance of the variables in the multi-task similarity measure. Although the predictor variable X_3 seems to have a similar pattern in all tasks, assessing its importance can capture the insignificance of this feature in the prediction (we know that, in reality, there is no relation with the outcome) and weigh its contribution to the multi-task similarity computation.

Table 1
Multi-task similarity measure values tasks in Synthetic Dataset 1. Additionally, it presents the weighted Fréchet distance between each variable and task, with the last column indicating the feature importance in each task.

	Task 1	Task 2	Task 3	Task 4	Task 5	Imp.
Task 1						
X_1	–	**10.41**	14.31	13.03	11.71	0.19
X_2	–	**9.46**	15.53	13.17	14	0.19
X_3	–	**1.75**	2.22	2.42	3.19	0.09
X_4	–	**7.89**	53.25	133.27	107.53	0.27
X_5	–	4.73	123.69	**3.44**	117.92	0.26
Similarity	–	**7.33**	52.35	42.22	64.84	
Task 2						
X_1	**10.41**	–	13.28	11.19	18.33	0.19
X_2	**9.46**	–	20.99	11.32	10.68	0.19
X_3	1.75	–	1.37	**1.22**	1.89	0.09
X_4	**7.89**	–	50.5	134.59	113.98	0.27
X_5	**4.73**	–	129.62	5.08	140.81	0.27
Similarity	**7.29**	–	54.68	41.78	73.72	
Task 3						
X_1	14.31	**13.28**	–	14.5	13.73	0.19
X_2	**15.53**	20.99	–	24.22	25.47	0.19
X_3	2.22	1.37	–	**0.93**	1.66	0.09
X_4	53.25	**50.5**	–	183.03	90.73	0.27
X_5	123.69	129.62	–	123.33	**2.51**	0.27
Similarity	53.03	54.64	–	89.02	**32.31**	
Task 4						
X_1	13.03	**11.19**	14.5	–	12.31	0.19
X_2	13.17	11.32	24.22	–	**9.51**	0.19
X_3	2.42	1.22	**0.93**	–	1.56	0.08
X_4	133.27	134.59	183.03	–	**41.08**	0.28
X_5	**3.44**	5.08	123.33	–	126.64	0.26
Similarity	**42.76**	**42.76**	90.49	–	49.1	
Task 5						
X_1	**11.71**	18.33	13.73	12.31	–	0.19
X_2	14	10.68	25.47	**9.51**	–	0.18
X_3	3.19	1.89	1.66	**1.56**	–	0.09
X_4	107.53	113.98	90.73	**41.08**	–	0.26
X_5	117.92	140.81	**2.51**	126.64	–	0.27
Similarity	64.94	73.34	**31.76**	49.17	–	

The most significant differences are observed between variables X_4 and X_5. It is evident that the shape of the curves is not the only determinant for the weighted Fréchet distance. As anticipated in the visual inspection, both tasks 1 and 2 have the lowest values of the multi-task similarity measure and in the majority of the weighted Fréchet distances between variables. Therefore, the results of the measure follow the intuition.

Task 3 exhibits two curves with opposite patterns in variables X_4 and X_5. Variable X_5 has a very similar pattern compared to the same variable in Task 5, and the weighted Fréchet distance supports this, resulting in a much lower value compared to the other tasks. However, the variable X_4 of Task 3 has a pattern that could be similar to Task 1 or 2, although it has a different distribution. This is reflected by the values of the weighted distance, as we obtain similarly low values compared to Tasks 1 and 2, but these values are much bigger compared to other patterns that are more similar, such as between Tasks 1 and 2. Ultimately, considering the weighted Fréchet distance of each variable and the importance of each feature, the multi-task similarity measure indicates that Task 5 is the most similar task. The remaining results can be explored in the values collected in Table 1.

As discussed in Section 3.4.1, an alternative version of the multi-task similarity measure can be used, incorporating the predictive quality of each task through Eq. (7). To demonstrate the usefulness of this approach, we simulated new data following the same distribution as Task 1 in the toy dataset, but training a deliberately limited model (specifically, a Random Forest model with only 3 trees and a minimum node size of 750 observations). The ALE curves obtained for this new task (Fig. 5) show that, although there are noticeable differences compared to those of Task 1 (Fig. 3), the overall behavior remains largely similar.

Table 2 presents the multi-task similarity measure between tasks, both with and without incorporating the performance-based scaling factor. The reference task is indicated by the row, and the compared task by the column. Values in parentheses reflect the similarity after adjusting for predictive quality. Notably, the similarities involving Task 6 (the poorly performing task) are considerably reduced when the performance factor is included-demonstrating that the similarity measure appropriately down weights comparisons involving low-quality models. For example, the similarity between Task 1 and Task 6 decreases from 29.73 to 21.34, and between Task 2 and Task 6 from 35.68 to 28.39. Despite the limited model used for Task 6, the similarity values (both raw and scaled) indicate that Task 6 is more similar to Task 1 than to any other task, which is expected given that both tasks share the same data distribution. This further supports the effectiveness of the performance-aware similarity adjustment in preserving meaningful relationships while penalizing unreliable comparisons.

4.2. Parkinson dataset

We also apply the multi-task similarity measure developed to the Parkinson dataset[1] [35], a real-world dataset containing information about 42 patients. The goal is to predict the disease symptom score of Parkinson at different times using 19 biomedical features. In the multi-task scenario, each of the 42 patients represents a task. The dataset consists of a total of 5875 records, and each patient (task) has approximately 200 data points. Although it is not the main goal of this work, this dataset is used as a benchmark in several multi-task learning algorithms [2].

Similarly to the approach followed in the previous synthetic dataset (Section 4.1), we have trained a separate random forest model for each patient (task). Fig. 6 shows the multi-task similarity measure between patients in a matrix format, aiding in the exploration of the re-

[1] https://archive.ics.uci.edu/dataset/189/parkinsons+telemonitoring

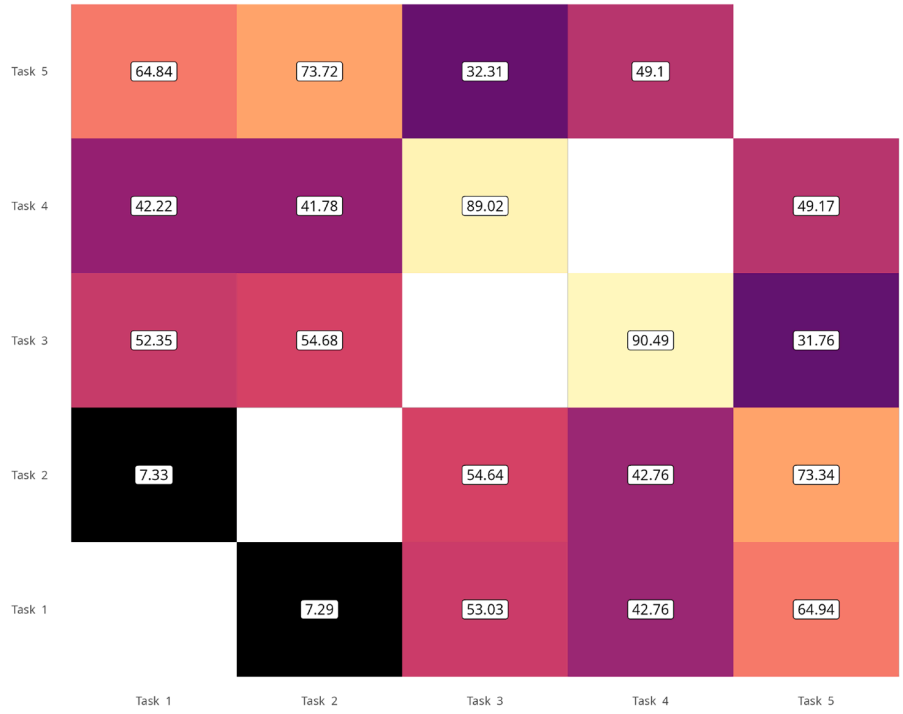


Fig. 4. Multi-task similarity values.

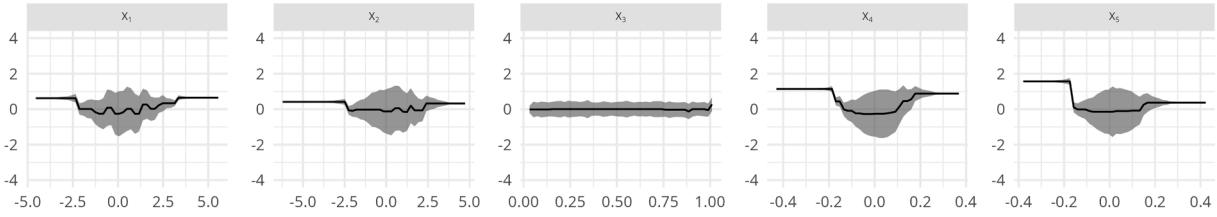


Fig. 5. ALE curves for the task 6.

Table 2
Multi-task similarity measure between a reference task (rows) and another task (columns). The values in parentheses refer to the measure incorporating the performance-based scaling factor.

	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6
Task 1	–	**7.33** (6.61)	52.35 (40.91)	42.22 (34.9)	64.84 (43.88)	29.73 (21.34)
Task 2	**7.29** (6.57)	–	54.68 (47.37)	41.78 (38.28)	73.72 (55.31)	35.68 (28.39)
Task 3	53.03 (41.44)	54.64 (47.33)	–	89.02 (84.16)	**32.31** (27.98)	44.55 (40.93)
Task 4	**42.76** (35.35)	**42.76** (39.18)	90.49 (85.55)	–	49.1 (40.21)	55.04 (47.8)
Task 5	64.94 (43.96)	73.34 (55.02)	**31.76** (27.51)	49.17 (40.26)	–	57.32 (54.04)
Task 6	**32.1** (23.04)	36.92 (29.38)	48.35 (44.42)	62.96 (54.68)	65.73 (61.97)	–

sults and the extraction of conclusions. As the matrix shows, we can use the multi-task measure to identify patients with different behavior compared to the others, not by using the raw observations, but by employing a model that summarizes the general behavior of the tasks (or patients in this context). For example, patient 14 has lower values of similarity (in general) than patient 15, indicating that patient 15 may have special characteristics that make them unique and different from the other patients in the sample. Patient 14, on the other hand, may exhibit a more homogeneous behavior with the rest of the patients.

Table 3 shows the five highest and five lowest similarity values between patients 14 and 15 with respect to the other 41 patients. Fig. 7 shows the ALE plots of the two most important variables for each task among the five most and least similar patients (tasks) to patients 14 and 15. As anticipated in the similarity values in Table 3, it can be observed that the tasks similar to Task 14 have a homogeneous increasing pattern in the most important variable *test_time* (time since recruitment into the trial), in contrast to the most similar tasks to Task 15, where there is a heterogeneous pattern. Furthermore, in the second most important variable (*DFA* or signal fractal scaling exponent), it is clearer that patient 15 exhibits a very different response compared to the other patients considered in the study sample.

With this application, we can see the potential of the measure to serve as an exploratory tool to comprehend not only the general behavior of the tasks but also the relationship and the differences between them.

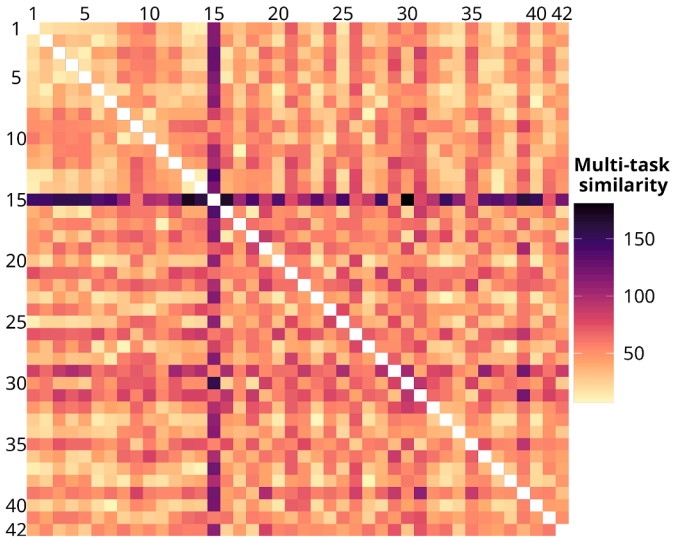


Fig. 6. Matrix of multi-task similarity between patients. The matrix must be interpreted as the similarity between a patient in a column with a patient in a row.

Table 3
Multi-task similarity (MTS) between the features of Task 1 and the features of the other tasks.

Patient 14		Patient 15	
Patient	MTS	Patient	MTS
37	9.33	9	47.16
13	11.52	29	55.64
1	11.66	35	56.24
4	20.43	41	58.90
25	22.21	26	60.87
…	…	…	…
21	76.72	4	126.17
31	81.54	3	129.56
26	88.39	16	136.62
29	98.40	13	139.63
15	159.31	30	154.93

4.3. Bike-sharing BiciMad dataset

This dataset contains information about the public bike-sharing system BiciMad,[2] which operates in Madrid, Spain. The records span between the years 2021 to 2023 and include data from 264 stations where bikes can be locked. The objective is to predict the number of bikes unlocked at each station within a one-hour period. The dataset comprises slightly more than 2 million rows and 10 features, including temporal variables such as month, weekday, and hour, as well as weather conditions such as precipitation and humidity. In Appendix C, Fig. C.2 displays the locations of the stations in Madrid, Spain.

In this case, the model used is a hybrid-parameter sharing deep learning model with three types of layers: (1) layers shared across all tasks with the same parameters (hard parameter sharing), (2) task-specific layers with parameters encouraged to be similar by applying a regularization constraint (soft parameter sharing), and (3) fully task-specific layers with independent parameters for each task. The full details of the model and dataset can be found in Appendix C. The multi-task similarity computation took approximately four minutes on a workstation equipped with an Intel Core i7-8700 (6 cores/12 threads), 16 GB of RAM, an NVIDIA GeForce GTX 1060 (3 GB), and Ubuntu 24.04.2 LTS.

The 264 tasks represent a relatively large number, making it challenging to manually inspect the relationships between them and derive a global interpretation of the model. This highlights the usefulness of the multi-task similarity measure in automatically extracting relationships between tasks. To explore and group tasks, the measure can act as a distance metric, enabling clustering algorithms to group similar tasks effectively.

In this work, we use hierarchical clustering to group the tasks based on the multi-task similarity measure. Fig. 8 presents the dendrogram obtained by applying a hierarchical agglomerative clustering method with Ward's linkage. The dendrogram suggests the presence of six clusters. In this case, all features were considered equally important, so the similarity matrix is symmetric.

Fig. 9 illustrates two of the resulting clusters. It can be observed that most features exhibit similar patterns in their ALE curves. However, there are three variables that present different behaviors. Specifically, the feature "Lag1" shows a sharp positive effect in Cluster 1, while in Cluster 3 it initially increases but then exhibits a slight negative trend. Similarly, "Radiation" reveals a nonlinear relationship that peaks around mid-range values in Cluster 1, whereas in Cluster 3 it shows a steeper and more variable effect with a strong increase toward higher values. Lastly, "Temperature" has a consistently increasing effect in Cluster 1 but a clearly decreasing influence in Cluster 3. These differences suggest that while the overall model behavior is aligned across clusters, these features capture distinct functional dependencies, potentially reflecting different underlying subpopulation dynamics. This phenomenon, where most variables exhibit similar patterns and only a few show differences, highlights the importance of having measures like the one developed in this work.

Similar to the previous dataset, the differing patterns between the most similar and dissimilar tasks can be identified. The first twelve plots in Figs. 10 and 11 show the ALE plots for the most similar and dissimilar tasks, respectively. Without a multitask similarity measure, it would not be straightforward to identify these tasks or to infer the nature of their differences.

The previous results, using a Random Forest model, can be found in Appendix C, showing that the outcomes are very similar.

4.3.1. Autoencoder

As discussed in Section 3.4.2, the relationship between the input and output data can be sufficiently complex that it cannot be captured by first-order ALE curves. One possible remedy discussed in Section 3.4.2 is to use an autoencoder to learn a set of relevant latent features that better represent the underlying structure of the data. For example, in the bike sharing dataset, the effects of the hour of the day and the day of the week do not contribute independently to the prediction, because, as one can intuit, bike usage patterns differ significantly between weekdays and weekends depending on the time of the day. An autoencoder can help capture such interactions by encoding them into a more informative feature space and even discarding irrelevant features. The main drawback of this approach is that the interpretability of ALE curves applied to these new latent features may be reduced. Nevertheless, the representation can still serve as a useful similarity measure between tasks.

For this bike sharing dataset, we trained an autoencoder using all the data in the training set, without distinguish between tasks (i.e., stations unlocked in this context). The encoder outputs six latent features. These transformed features are then used as input to a separate model for each task. Further details are provided in the Appendix C.1.

Fig. 12 shows the ALE curves for each of the six latent features. Although these features lose the interpretability of the original input variables, the dimensionality reduction offers significant benefits-particularly in facilitating manual inspection of the relationships between inputs and outputs and, when possible, identifying some latent features as inherent properties of the data. The main advantage of the autoencoder-based representation is that complex relationships among the original features are captured and summarized within these six latent variables.

[2] https://www.bicimad.com/en/home

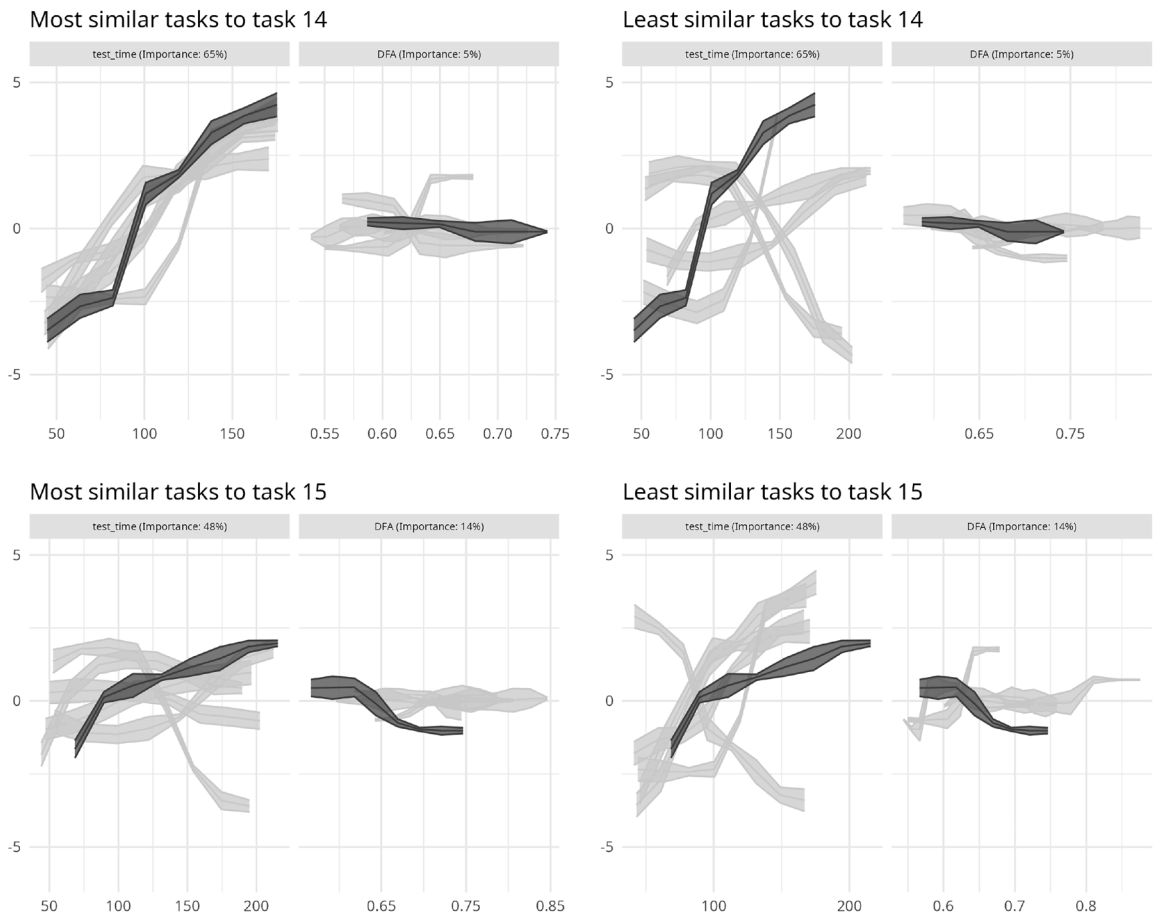


Fig. 7. ALE plots for the five most and least similar tasks to Tasks 14 and 15. The bold line represents the task of interest. In the figure, only the two most important variables for each task of interest are shown.

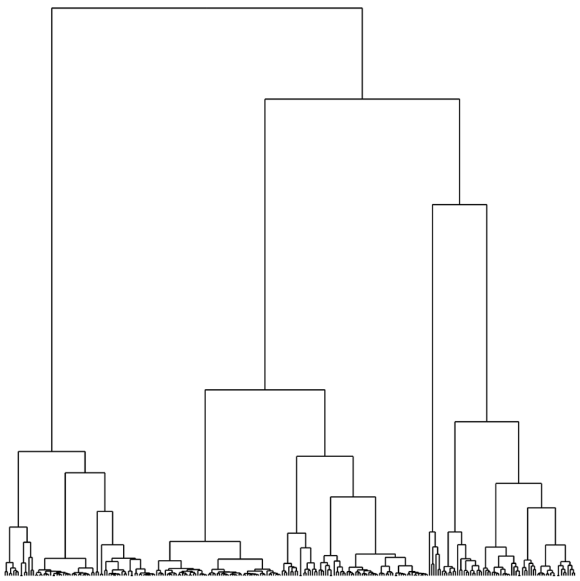


Fig. 8. Dendrogram for the hierarchical agglomerative clustering method using Ward's linkage and the multi-task similarity measure values.

The last row of plots in Figs. 10 and 11 presents the ALE curves for the same tasks previously discussed, but now using the autoencoder-based codification. Notably, the ALE plots exhibit much more distinguishable and structured patterns in the encoded latent features compared to those based on the original input variables.

Table 4
Comparison of multi-task similarity statistics with and without spline smoothing.

Method	Mean Similarity	Standard Deviation
Without Smoothing	23.682	13.281
With Spline Smoothing	26.523	13.342
Similarity differences	−0.945	0.658

4.3.2. *Spline smoothing*

As discussed in Section 3.4.2, the ALE curves can exhibit high-frequency oscillations that can obscure the interpretability of the method. Fig. 13 shows the same ALE curves as in Fig. 9 but with roughness-penalized cubic spline smoothing applied. It can be observed that the rapid fluctuations present in the original curves are considerably attenuated, resulting in smoother and more interpretable profiles. While t he overall trends of the effects are preserved, but the smoothed curves reduce visual noise, enabling for clearer identification of feature influence across clusters.

Although spline smoothing improves the interpretability of ALE curves by reducing high-frequency noise, it does not significantly alter the values of the multi-task similarity measure between tasks. Table 4 shows that both the mean and standard deviation of similarity values remain relatively stable before and after applying spline smoothing. The mean similarity increases slightly from 23.682 to 26.523, and the standard deviation remains nearly unchanged. Furthermore, the root mean square error (RMSE) of the similarity values between the two configurations is 1.151, indicating minimal overall deviation introduced by the

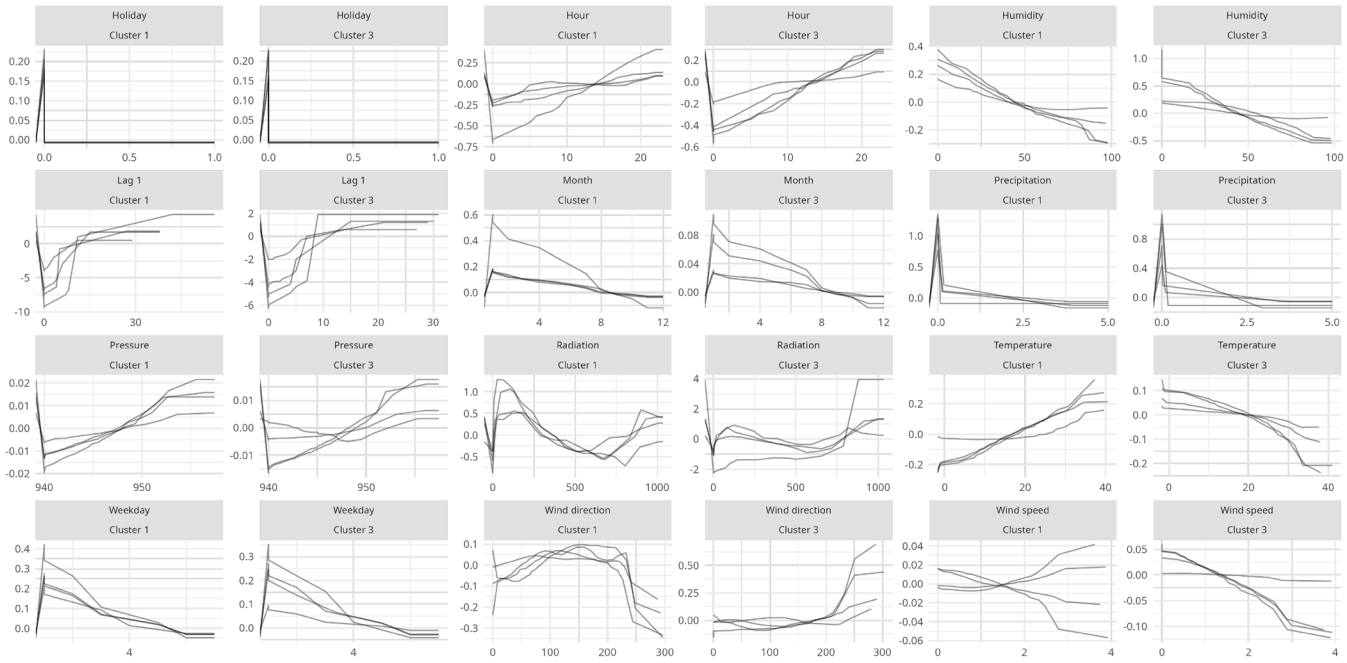


Fig. 9. Centered ALE plots for two of the clusters.

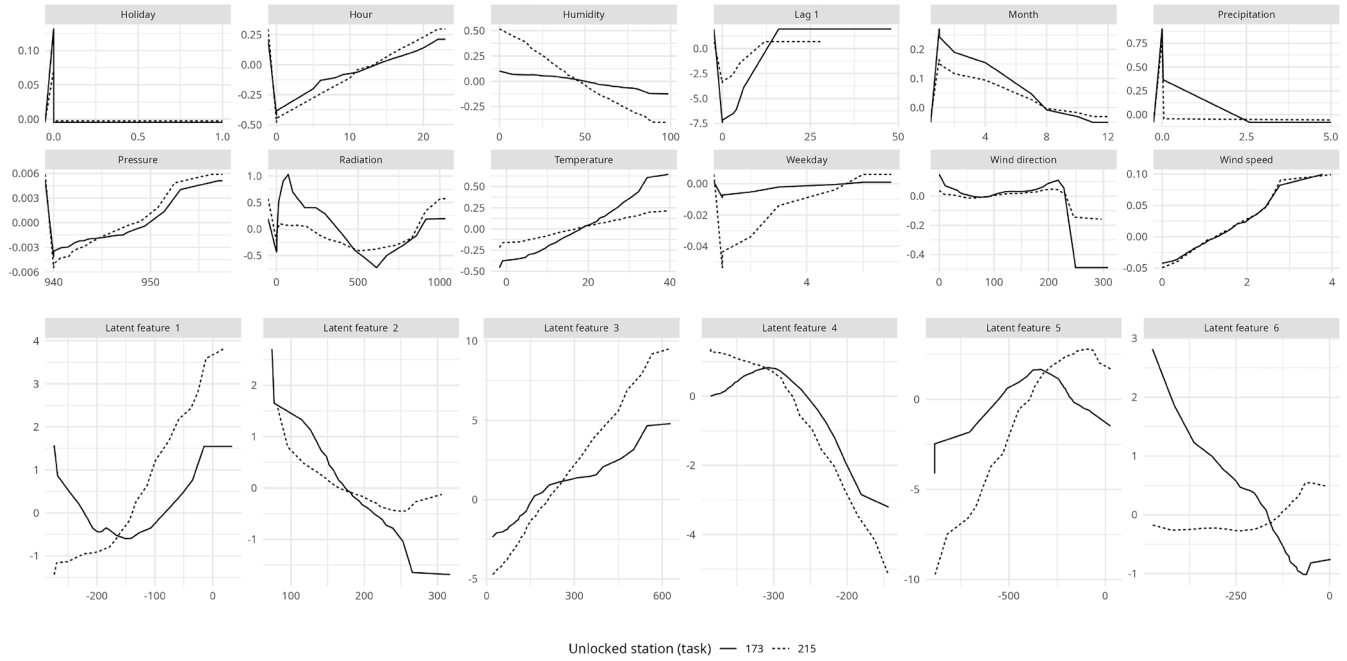


Fig. 10. Centered ALE plots for all features from the two most similar tasks.

smoothing process. Additionally, a Spearman correlation of 0.9953 is obtained.

These results support the robustness of the proposed similarity measure with respect to minor variations in curve smoothness.

4.4. Image dataset

The previous datasets share the common characteristic of being tabular, meaning the information is structured as observations and variables in a table-like format. While this structure is naturally well-suited for use with the multitask similarity measure, the measure can also be applied to other data types, provided the information is appropriately structured.

In Section 3.4.2, we proposed several strategies to apply the measure to non-tabular data. Although the approaches differ in how the latent representation is computed, they all share the same underlying idea: encoding the original data into a latent space.

To provide experimental validation of these ideas, we applied the concept bottleneck encoder to the CelebA image dataset.[3] This dataset contains facial images annotated with multiple binary attributes, which

[3] https://mmlab.ie.cuhk.edu.hk/projects/CelebA.html

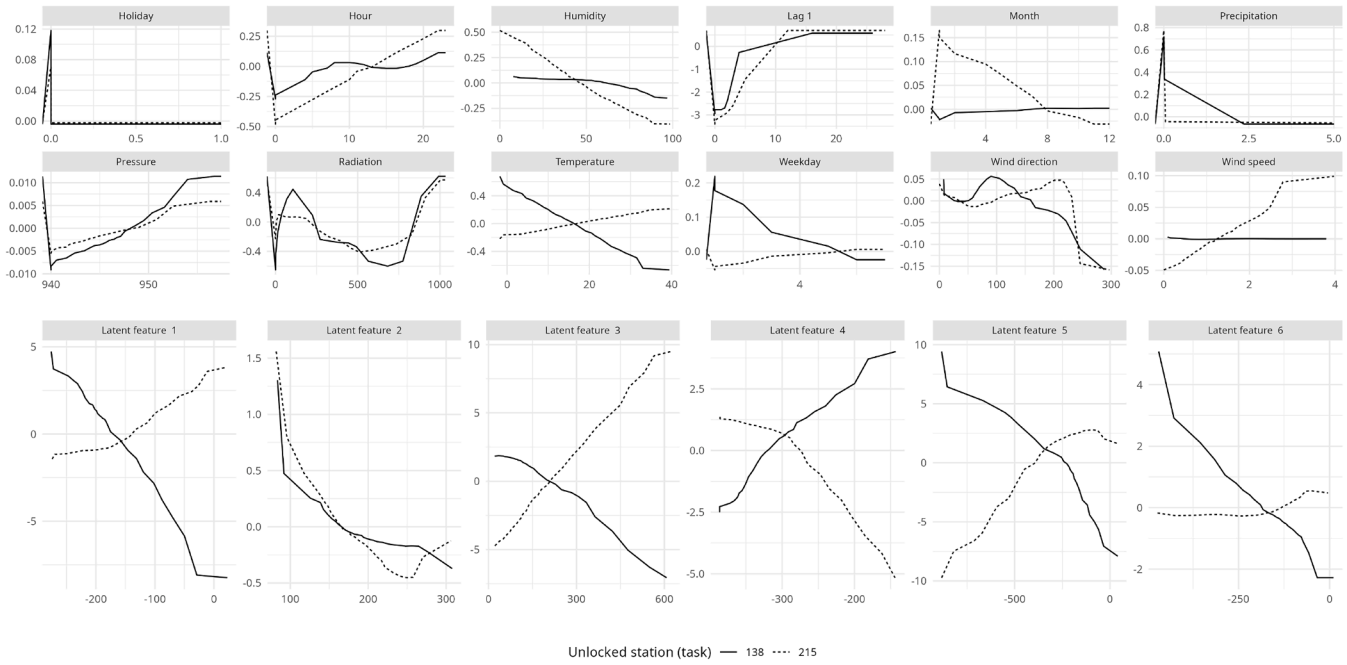


Fig. 11. Centered ALE plots for all features from the two most dissimilar tasks. The first twelve plots correspond to the original data, while the remaining six result from using the autoencoder.

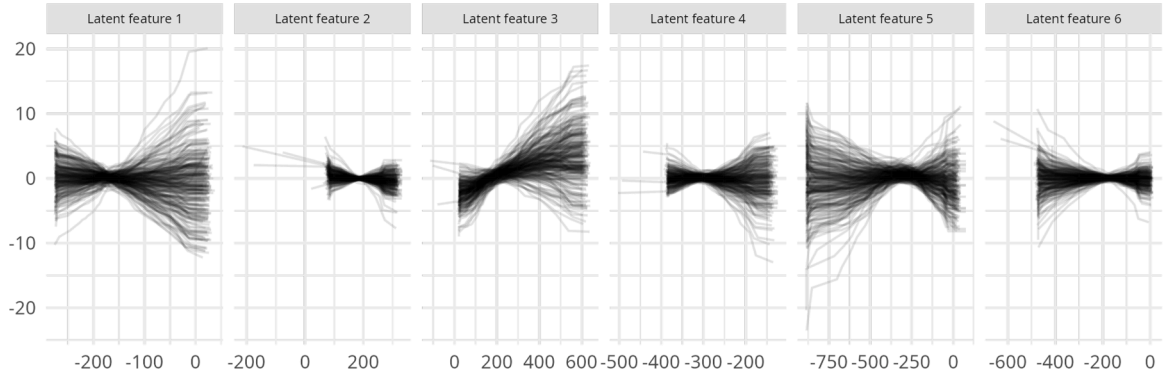


Fig. 12. Centered ALE plots for the four latent features obtained in the autoencoder.

serve as human-interpretable concepts and are well suited for evaluating concept-based representations. Each image is annotated with 40 binary attribute such as *Smiling, Wearing Glasses*, or *Mustache*. It is reasonable to assume that some of these attributes are correlated, for instance, images labeled as *Male* are likely to have a higher probability of also being labeled as *Mustache*.

In our setup, half of the attributes are used as concepts in the concept bottleneck encoder, while the other half serve as separate outputs, each representing an individual task in the multitask learning setting. Further details can be found in Appendix D.

The neural network employed consisted of a Convolutional Neural Network (CNN) used as an encoder to extract a latent concept representation from the input images. This encoder maps each image to a 20-dimensional binary concept vector through a series of convolutional and fully connected layers, using ReLU activations in the hidden layers and sigmoid activations at the output to produce probabilities for each concept.

These predicted concept representations serve as input to a multitask classifier, implemented as a fully connected neural network that independently predicts the 20 target attributes —each corresponding to a separate binary classification task. The full model is trained end-to-end using a combined loss function that includes binary cross-entropy for both concept prediction and final multitask outputs, thereby ensuring the consistency and relevance of the latent representations.

Once trained, the multitask similarity measure is applied to the ALE curves computed for each concept. This allows us to evaluate how variations in the predicted probability of each concept affect the probability of the output attributes across tasks. By quantifying the similarity between tasks in terms of these concept-based ALE curves, we can better understand the functional relationships learned by the model and the role of shared or divergent concept influences.

Table 5 presents the pairwise similarity between four representative target task attributes from the CelebA dataset, as computed by the proposed multitask similarity measure based on ALE curves over the learned concept space. The pairwise distances confirm several intuitive patterns. The *Mustache* and *Male* tasks form the closest pair, with a similarity value of only 0.17, reflecting the strong association between facial hair and gender in CelebA. At the other extreme, *Mustache* and *Wearing Lipstick* are separated by the largest value (0.88), underscoring that these tasks depend on markedly different concept cues. A cosmetically driven relationship is evident between *Wearing Lipstick* and *Wearing Necklace* (0.36), the next-smallest distance in the matrix, suggesting that similar latent

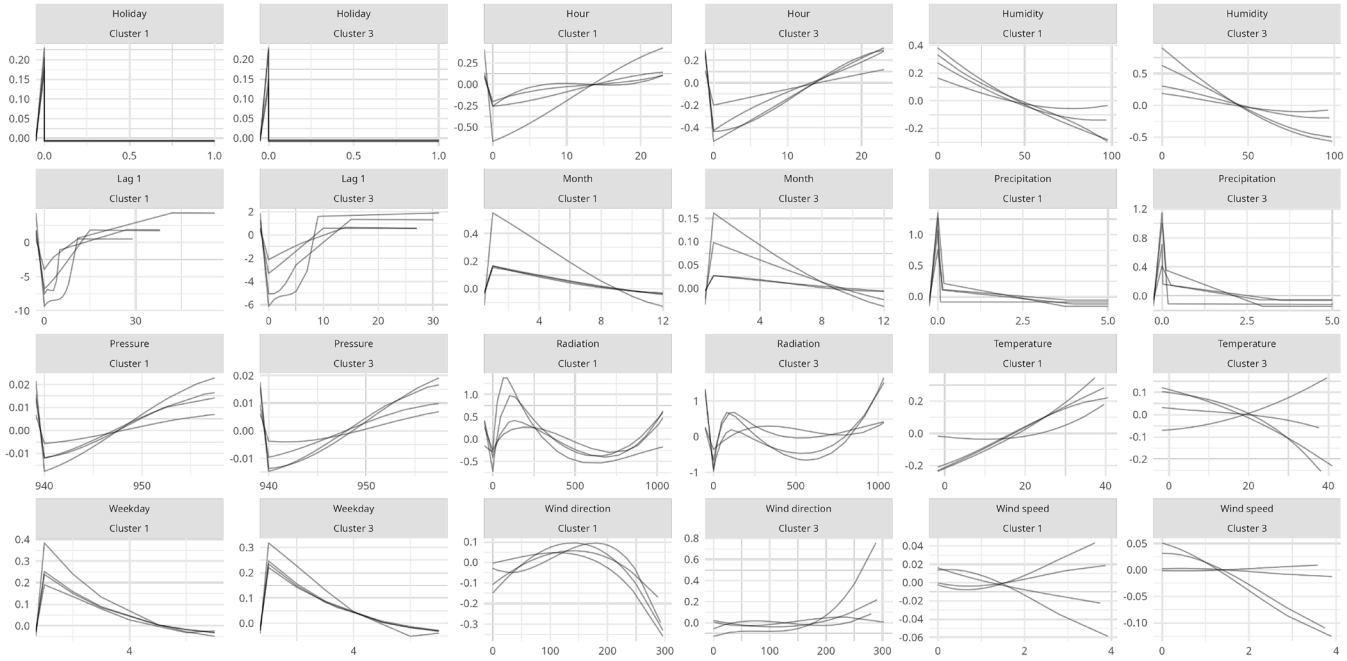


Fig. 13. Centered ALE plots for two of the clusters applying spline smoothing.

Table 5
Pairwise similarity between selected output tasks in the CelebA dataset based on ALE curve comparisons over concept representations. Lower values indicate more similar tasks.

Task	Mustache	Male	Wearing Lipstick	Wearing Necklace
Mustache	–	0.17	0.88	0.73
Male	0.17	–	0.81	0.53
Wearing Lipstick	0.88	0.81	–	0.36
Wearing Necklace	0.73	0.53	0.36	–

features-likely linked to femininity and facial accessories-govern both outputs. Finally, intermediate values (0.53 0.73) for pairs involving *Male* and *Wearing Necklace* signal a moderate overlap of underlying concepts, whereas distances above 0.80 denote tasks with largely distinct functional dependencies. Overall, lower distances consistently correspond to semantically plausible task similarities, validating the usefulness of the ALE-based measure.

Table 6 confirms that the multi-task similarity measure capture semantically meaningful relations between concepts and tasks. Gender-linked cues-especially *Bald*, *Black Hair*, and *Eyeglasses*-exhibit near-zero distances to *Male* (0.01, 0.02, and 0.03, respectively) and similarly small values to *Mustache*, reflecting their shared dependence on male identity. In contrast, the cosmetic concept *Heavy Makeup* aligns most strongly with the femininity cue *Wearing Lipstick* (0.05, the lowest entry in its row), while all concepts show markedly larger distances to the accessory task *Wearing Necklace* (0.11 0.41). Hence lower distances consistently correspond to intuitive task-concept affinities, demonstrating that the multitask-similarity metric surfaces the expected gender and cosmetic patterns while down-weighting weakly informative features.

5. Results and discussion

The application of the multi-task similarity measure across the datasets discussed in Section 4 highlights its utility in identifying and explaining relationships between tasks. The measure's outcomes align with intuitive expectations, enabling the identification of tasks with comparable relationships between features and outcomes from an explainable perspective.

Table 6
Multi-task similarity measure between concepts and two CelebA tasks. Each entry is the weighted Fréchet distance (lower ⇒ stronger relation).

	Mustache	Male	Wearing Lipstick	Wearing Necklace
Mustache				
Bald	–	**0.01**	0.11	0.12
Heavy Makeup	–	0.05	**0.05**	0.07
Black Hair	–	**0.02**	0.17	0.13
Eyeglasses	–	**0.03**	0.13	0.11
…		…	…	…
Male				
Bald	**0.01**	–	0.51	0.33
Heavy Makeup	0.05	–	**0.80**	0.28
Black Hair	**0.02**	–	0.13	0.37
Eyeglasses	**0.03**	–	0.12	0.41
…	…		…	…

The synthetic dataset of Section 4.1 serves an ideal testbed to validate the measure in a controlled environment. The computed similarity values correspond with visual observations of the ALE curves, underscoring the measure's reliability. In this ad hoc dataset, consisting of only five tasks with five variables each, it is straightforward to manually identify similar tasks without requiring a formal measure. Nevertheless, the computed similarity values align with these manual observations, further validating the measure.

Conversely, the relatively small Parkinson's dataset, comprising 42 patients with approximately 200 records per patient, presents a significant challenge in identifying task similarities (patients, in this context) without an appropriate similarity measure. This underscores the importance of the multi-task similarity measure in uncovering such relationships and facilitating the derivation of meaningful conclusions.

Furthermore, the third dataset, used for predicting bike-sharing usage, involves 264 tasks. Identifying task dynamics in this scenario without a summarization tool is particularly challenging. Here, the similarity measure can serve as the foundation for clustering methods to group tasks effectively. Importantly, the measure captures both the overall similarity and variable-specific influences, validating its robustness and interpretability across datasets of varying complexity.

A logical sequence in utilizing the measure involves first computing the multi-task similarity measure between all the tasks, or at least between one task of interest and the remaining tasks. Second, it is beneficial to examine not only the most similar tasks but also the least similar ones. By preserving intermediate calculations necessary for computing the measure, it becomes possible to elucidate why and how tasks are similar or dissimilar. This includes identifying the relevant variables that contribute to the low or high values of the similarity measure.

Additionally, the measure allows for intervention in its behavior to emphasize certain variables. Ideally, the importance of each variable used in the similarity calculation should be determined using a reliable method. However, researchers can manually adjust the variable weights to prioritize those deemed most relevant or actionable for a specific study.

Several multi-task learning scenarios presuppose, either implicitly or explicitly, a degree of homogeneity among the tasks. The proposed measure is robust enough to be applied to heterogeneous tasks, even when the relationships are hidden behind task structures. For instance, if one task's data includes a variable called *age* and another task's data includes *edad* (age in Spanish), the measure can detect the similarity between these variables despite the difference in nomenclature, provided they represent the same information.

A significant advantage of this measure is its model-agnostic nature. It is irrelevant which model has been used to train each task, and it can even handle a mix of models. For example, some tasks may be trained using black-box machine learning models like deep learning, while others use traditional statistical models like linear regression. This is because the measure compares ALE curves derived from the models, and ALE curves are inherently model-agnostic.

However, the validity of the results heavily depends on the quality of the trained models. If the model for each task is not sufficiently capable of capturing the true patterns in the data and the relationships between predictor variables and the target, the resulting measure may not be accurate. Conclusions and subsequent actions are limited if the models are not well-developed. It is critical to ensure that all models used meet appropriate quality standards and exhibit suitable behavior. Furthermore, the measure can serve as an exploratory tool to identify defective models, allowing researchers to refine and improve them.

One challenge in using the measure lies in how the ALE curves are computed. Since the Fréchet distance defined here uses a summation instead of the maximum, significant differences in the number of segments in the ALE curves between tasks can affect the reliability of the measure's results. This issue can be alleviated if all ALE curves have the same number of segments, which makes the measure more robust. At the very least, if it is known which variables to be compared, it is recommended that all of the ALE curves corresponding to these variables have the same number of segments.

Like any machine learning technique, the multi-task similarity measure can help achieve its defined purpose, but it must be used cautiously and not applied uncritically.

6. Conclusions and future work

In this work, we explored the construction of a multi-task similarity measure from an explainable perspective. This approach aims to identify similar tasks in a multi-task scenario and provide insights into why and how they are similar. While several similarity measures exist for comparing tasks, particularly in multi-task learning, — none explicitly facilitate the explanation of task similarities— which is invaluable for researchers.

Throughout the paper, we applied the measure to four datasets: one synthetic dataset designed to demonstrate the behavior of the measure in a controlled environment, a real dataset of Parkinson's patients, a dataset focused on predicting bike-sharing usage, and an image dataset (CelebA) using concept bottleneck models. Our results demonstrate the measure's utility in detecting task similarities, highlighting the intuitive understanding of similarity provided by the multi-task similarity measure.

Furthermore, although there are contexts in which the number of tasks may pose computational challenges, the execution times for the relatively large Bike Sharing Dataset (264 tasks) are reasonable and manageable even on modest hardware.

This research opens up several avenues for future work. Firstly, as demonstrated with the bike sharing dataset, the similarity measure can be integrated into clustering methods to automatically group similar tasks. An important line of research would involve identifying which clustering methods are most suitable for this purpose, considering factors such as scalability, interpretability, and robustness. Additionally, the quality of the measure can be studied across different models with diverse underlying hypotheses (e.g., linear versus non-linear models).

Secondly, the similarity measure can be incorporated into multi-task learning algorithms. For instance, in soft parameter-sharing algorithms, the constraints on shared parameters could be derived directly from the similarity measure. This would allow the algorithm to dynamically adjust parameter sharing based on the quantified similarity between tasks, potentially improving both performance and interpretability.

CRediT authorship contribution statement

Pablo Hidalgo: Writing – review & editing, Writing – original draft, Software, Methodology, Formal analysis, Conceptualization; **Daniel Rodriguez:** Writing – review & editing, Writing – original draft, Conceptualization.

Data availability

Data will be made available on request.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors thank their respective universities for the support and Daniel also acknowledges the project PID2021-125645OB-I00 (PARCHE), funded by MCIN/AEI/10.13039/501100011033/FEDER, EU.

Appendix A. Synthetic dataset 1

We simulate five tasks, each with $10,000$ observations and five features generated as follows:

- X_1 and X_2 are simulated from a bivariate normal distribution with the same mean μ and covariance matrix Σ across all tasks,
- X_3 is simulated uniformly in the $(0, 1)$ interval, and
- X_4 and X_5 are simulated as a mixture of normals, with each task having its own parameters.

The outcome Y is generated as shown in Eq. (A.1).

$$Y = Rastrigin_{std}(X_1, X_2) + q_{std}(X_4, X_5). \tag{A.1}$$

The first part of the sum is computed as

$$Rastrigin(X_1, X_2) = 20 + \sum_{i=1}^{2}(X_i^2 - 10\cos(2\pi X_i)) \tag{A.2}$$

Table A.1
Simulated data generation for each of the five tasks. $\mathcal{N}_2(\mu, \Sigma)$ refers to the bivariate normal of mean μ and standard deviation Σ, $\mathcal{U}(a, b)$ refers to uniform distribution in the interval (a, b) and $\mathcal{N}(\mu_1, \Sigma_1) + \mathcal{N}(\mu_2, \Sigma_2)$ is a mixture of two normals. The a, b and c are the parameters of the quadratic form expressed in Eq. (A.1).

	X_1 and X_2		X_3	X_4 and X_5				Y		
Task	$\mathcal{N}_2(\mu$	$\Sigma)$	$\mathcal{U}(a, b)$	$\mathcal{N}(\mu_1$	$\Sigma_1)+$	$\mathcal{N}(\mu_2$	$\Sigma_2)$	a	b	c
1				0	0.1	0	0.1	1	1	1
2				0	0.1	0	0.1	1	1	−1
3	$(0, 0)$	$\begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$	$(0, 1)$	−0.25	0.1	0.25	0.1	1	−1	1
4				0	0.1	0	0.1	−1	1	1
5				−0.25	0.1	0.25	0.1	−1	−1	1

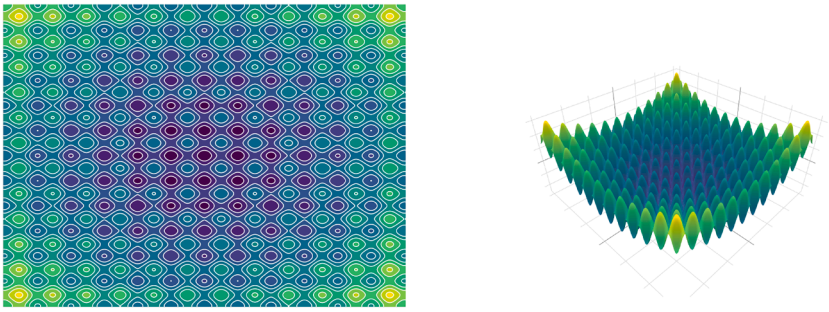


Fig. A.1. Surface plots and contour plots of the relationship between predictor variables X_1 and X_2 and the outcome across all tasks.

The second part is the quadratic form

$$q(X_4, X_5) = aX_4^2 + bX_5^2 + cX_4X_5 \tag{A.3}$$

where the scalars $a, b, c \in \{-1, 1\}$ are dependent of the task.

The functions $Rastrigin_{std}(X_1, X_2)$ and $q_{std}(X_4, X_5)$ appearing in Eq. (A.1) are the normalized versions of the respective functions, i.e., with zero-mean and unit variance. For a detailed breakdown of the simulation, we refer to Table A.1.

The relationship of the predictors with the outcome, as per Eq. (A.1), can be decomposed into two additive components.

The first one is attributed to X_1 and X_2 through the *Rastrigin* function described in Eq. (A.2). This complex function is commonly employed to test the performance of optimization algorithms, such as genetic algorithms [36]. It is characterized by its complexity, featuring numerous local minima and a large search space. The function's graphical representation can be observed in Fig. A.1.

The second component consists of a quadratic form with respect to variables X_4 and X_5. The shape of the quadratic forms will vary depending on the values of $a, b, c \in \{-1, 1\}$ as illustrated in Table A.1. The relation between variables X_4 and X_5 is shown in Fig. A.2. Although there are tasks that share a similar shape with respect to the projection to one of the variables, the variation in the distribution of variables X_4 and X_5 across tasks (see Fig. 2) will cause the similarity measure to vary.

The predictor variable X_3 has no relationship with the outcome. Therefore, if the similarity measure is well-constructed and sufficiently robust, this variable should not affect the values obtained from the measure.

The model used for training was a Random Forest, implemented using the randomForest library of R. To obtain the best model for each task, a grid search strategy was employed, testing combinations of the number of trees (50, 100, 250, and 500) and the number of variables randomly selected at each split (1, 2, 3, 4, and 5). All other hyperparameters were left at their default values. The models were trained on 70 % of the data, with the remaining 30 % used for testing.

The best combination for each model, based on the root mean squared error (RMSE), is represented in Table A.2

Table A.2
RMSE for each task.

	Task 1	Task 2	Task 3	Task 4	Task 5
RMSE (train)	0.686	0.699	0.713	0.685	0.717
RMSE (test)	0.707	0.711	0.741	0.699	0.736
Number of trees	250	250	100	250	100
Features randomly selected	1	3	2	2	4

Appendix B. Parkinson dataset

The model for the Parkinson dataset was trained in a similar manner to the approach described in Appendix A. In this case, the number of variables randomly selected at each split was chosen from the values 1, 5, 10, 15, and 19.

Appendix C. Bike-sharing BiciMad dataset

The dataset is constructed by combining the information available from the Open Data portal of the Empresa Municipal de Transportes de Madrid (EMT) (Municipal Transport Enterprise of Madrid, in English)[4] and weather condition data from the Open Data portal of the Madrid City Hall, Spain.[5] The predictor variables are shown in Table C.1

The proposed neural network is a hybrid-parameter sharing deep learning model with three types of layers, designed for multi-task learning. This architecture leverages shared representations across tasks while preserving task-specific nuances.

The architecture consists of one layer shared across all tasks with the same parameters (hard parameter sharing), followed by task-specific layers with parameters encouraged to be similar by applying a regularization constraint (soft parameter sharing) and (3) fully task-specific

[4] https://opendata.emtmadrid.es/Datos-estaticos/Datos-generales-(1)

[5] https://datos.madrid.es/portal/site/egob/menuitem.c05c1f754a33a9fbe4b2e4b284f1a5a0/?vgnextoid=fa8357cec5efa610VgnVCM1000001d4a900aRCRD&vgnextchannel=374512b9ace9f310VgnVCM100000171f5a0aRCRD&vgnextfmt=default

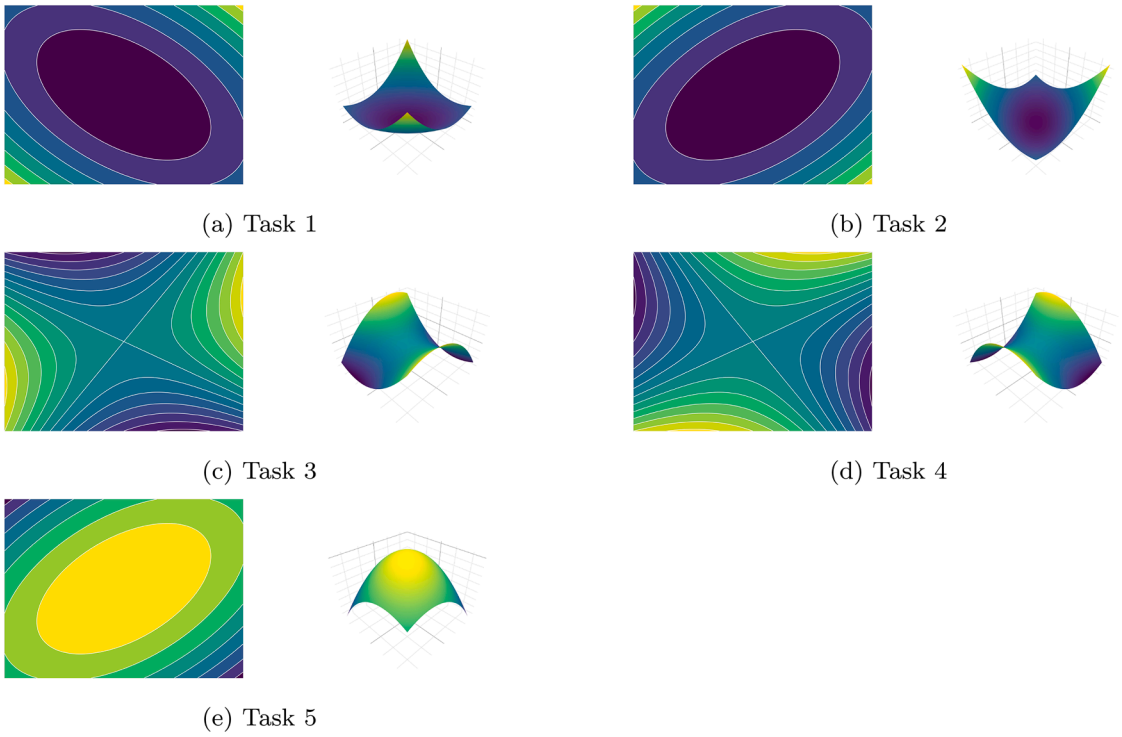


Fig. A.2. The figures depict the contour plots and surface plot of the predictor variables X_4 and X_5 for the different tasks.

Table C.1
Description of predictor variables used in the model.

Variable	Type / Range	Unit	Brief description
Lag 1	numeric	same as target	Target variable at $t-1$ (one hour earlier).
Lag 2	numeric	same as target	Target variable at $t-2$ (two hours earlier).
Lag 3	numeric	same as target	Target variable at $t-3$ (three hours earlier).
Lag 4	numeric	same as target	Target variable at $t-4$ (four hours earlier).
Lag 5	numeric	same as target	Target variable at $t-5$ (five hours earlier).
Wind speed	numeric	$m\,s^{-1}$	Mean horizontal wind speed during the current hour.
Wind direction	numeric (0-360)	degrees	Meteorological wind direction (0° = North, 90° = East).
Temperature	numeric	°C	Ambient 2-m air temperature.
Humidity	numeric (0-100)	%	Relative humidity.
Pressure	numeric	hPa	Surface atmospheric pressure.
Radiation	numeric	$W\,m^{-2}$	Global horizontal solar irradiance.
Precipitation	numeric	mm	Liquid-equivalent precipitation accumulated over the previous hour.
Holiday	binary (0/1)	-	Indicator equal to 1 if the date is a national/public holiday.
Month	integer (1-12)	-	Calendar month (1 = January ... 12 = December).
Hour	integer (0-23)	-	Hour of day in local (solar) time.
Weekday	integer (1-7)	-	Day of week (1 = Monday ... 7 = Sunday).

layers with independent parameters for each task. The details are as follows:

- **Input**: A $264 \times 128 \times 10$ tensor.
- **Hard parameter sharing**:
  - Fully Connected Layer with 264 neurons and ReLU activation.
- **Soft parameter sharing** (×264):
  - Fully Connected Layer with 264 neurons and ReLU activation.
  - Fully Connected Layer with 128 neurons and ReLU activation.
- **Output Layer** (×264): A single neuron with ReLU activation (the output must be a positive number).

Fig. C.1 provides a visual representation of the model.

To obtain the best model, a grid search strategy was employed using a combination of learning rates $(0.1, 0.001, 0.0001,$ and $0.00001)$ and a penalty factor values for the regularization constraint in soft parameter sharing $(0.1, 0.01,$ and $0.001)$.

The ALE curves were calculated with a default size of 30 intervals. However, the variables with fewer than 30 unique values (e.g., the month feature), an appropriate number of intervals was used.

To contextualize the data set, Fig. C.2 displays the locations of the stations Madrid, Spain.

The Multi-task Similarity Measure has been computed for the same dataset, this time using a Random Forest model. Fig. C.3 displays the same two clusters shown in Fig. 9, but generated with the Random Forest model. It can be observed that the behavior of the Centered ALE plots is very similar.

C.1. Autoencoder

The autoencoder consists of a symmetric architecture with an encoder and a decoder. The encoder maps the 16-dimensional input to a 6-dimensional latent representation through two fully connected lay-

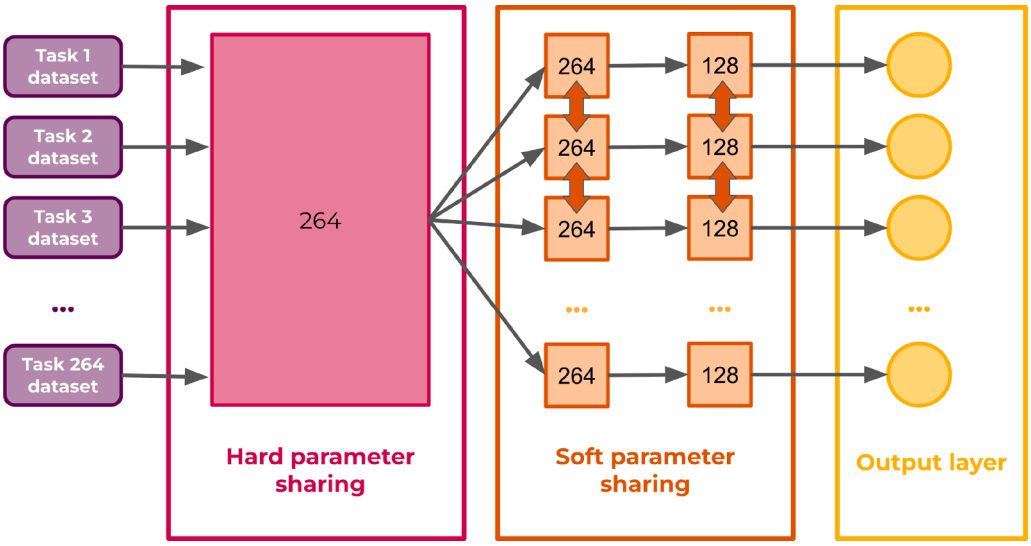


Fig. C.1. Neural network architecture for Bike-sharing dataset.

Fig. C.2. Locations of the stations in the city of Madrid, Spain.

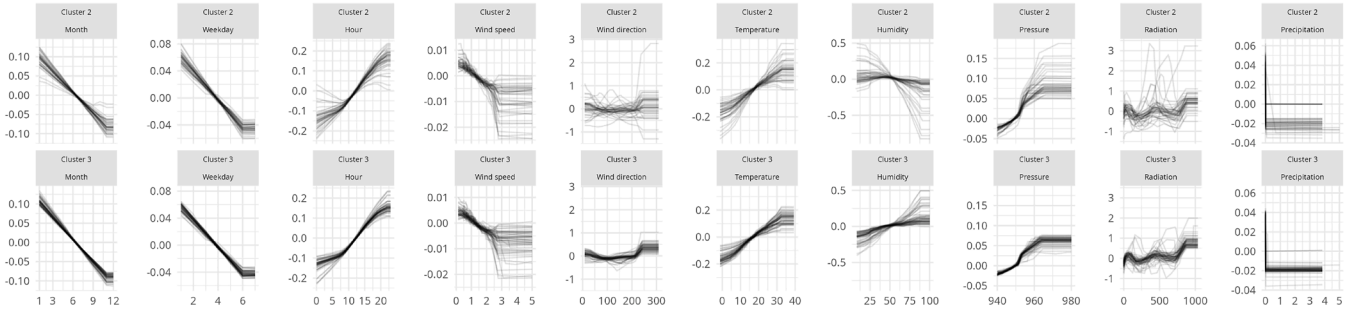


Fig. C.3. Centered ALE plots for all features from two of the clusters using a Random Forest model.

ers of sizes 16 *rightarrow* 10 *rightarrow* 6, using ReLU activations. The decoder reconstructs the original input by mirroring this structure (6 *rightarrow* 10 *rightarrow* 16), also using ReLU activations, except for the output layer, which uses a linear activation.

The model was trained using the mean squared error (MSE) loss function with the Adam optimizer (learning rate = 0.0001, batch size = 128) for 100 epochs. Early stopping was applied with a patience of 10 epochs.

The complete dataset was used without differentiating between tasks, and the target variable (use) was excluded from the input to the autoencoder.

Appendix D. CelebA dataset

In this work, we use the CelebA dataset to validate the application of concept bottleneck encoders in a multitask learning setting. CelebA consists of over 200,000 facial images, each annotated with 40 binary attributes representing various semantic features such as facial traits, accessories, and demographic characteristics.

To structure the multitask setting and the concept bottleneck, we split the 40 attributes into two disjoint sets:

- **Concept attributes (used as bottleneck variables):** These 20 attributes are used as interpretable intermediate representations within the concept bottleneck encoder. They represent observable traits that are expected to capture key visual semantics:

  Arched Eyebrows, Bags Under Eyes, Bald, Bangs, Big Lips, Big Nose, Black Hair, Blond Hair, Brown Hair, Bushy Eyebrows, Double Chin, Eyeglasses, Heavy Makeup, High Cheekbones, Mouth Slightly Open, Pale Skin, Receding Hairline, Smiling, Straight Hair, Wavy Hair.
- **Target task attributes (used as multitask outputs):** These 20 attributes are predicted from the latent concept representations. Each of them defines a separate output in the multitask learning setup:

  5 o'Clock Shadow, Attractive, Chubby, Goatee, Gray Hair, Male, Mustache, Narrow Eyes, No Beard, Oval Face, Pointy Nose, Rosy Cheeks, Sideburns, Wearing Earrings, Wearing Hat, Wearing Lipstick, Wearing Necklace, Wearing Necktie, Young, Blurry.

All attributes were binarized and normalized prior to training. The dataset was randomly split into training, validation, and test sets following the official partitioning provided by CelebA. Images were resized to 64×64 pixels, and standard data augmentation techniques (random horizontal flips and normalization) were applied during training.

This setup allows us to evaluate the capacity of the concept bottleneck encoder to extract meaningful latent features that can support multiple downstream tasks while maintaining interpretability.

References

[1] R. Caruana, Multitask learning, Mach. Learn. 28 (1) (1997) 41–75. https://doi.org/10.1023/A:1007379606734

[2] Y. Zhang, Q. Yang, A survey on multi-task learning, IEEE Trans. Knowl. Data Eng. 34 (12) (2022) 5586–5609. https://doi.org/10.1109/TKDE.2021.3070203

[3] Z. Zhang, H.A. Hamadi, E. Damiani, C.Y. Yeun, F. Taher, et al., Explainable artificial intelligence applications in cyber security: state-of-the-art in research, IEEE Access 10 (2022) 93104–93139. https://doi.org/10.1109/ACCESS.2022.3204051

[4] T.-C.T. Chen, Explainable artificial intelligence (XAI) in manufacturing, in: Explainable Artificial Intelligence (XAI) in Manufacturing, Springer International Publishing, Cham, 2023, pp. 1–11. Series Title: SpringerBriefs in Applied Sciences and Technology, https://doi.org/10.1007/978-3-031-27961-4_1

[5] A. Chaddad, J. Peng, J. Xu, A. Bouridane, et al., Survey of explainable AI techniques in healthcare, Sensors 23 (2) (2023) 634. https://doi.org/10.3390/s23020634

[6] S. Ruder, An overview of multi-task learning in deep neural networks, 2017, arXiv preprint arXiv:1706.05098

[7] A. Barredo Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, F. Herrera, et al., Explainable artificial intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI, Inf. Fusion 58 (2020) 82–115. https://doi.org/10.1016/j.inffus.2019.12.012

[8] F.K. Dosilovic, M. Brcic, N. Hlupic, et al., Explainable artificial intelligence: a survey, in: 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), IEEE, Opatija, 2018, pp. 0210–0215. https://doi.org/10.23919/MIPRO.2018.8400040

[9] D.W. Apley, J. Zhu, Visualizing the effects of predictor variables in black box supervised learning models, J. Royal Stat. Soc.: Ser. B (Stat. Methodol.) 82 (4) (2020) 1059–1086. https://doi.org/10.1111/rssb.12377

[10] J.H. Friedman, Greedy function approximation: a gradient boosting machine, Ann. Stat. 29 (5) (2001) 1189–1232. http://www.jstor.org/stable/2699986.

[11] T. Eiter, H. Mannila, Computing discrete frechet distance, Technische Universitat Wien Technical report (1994). https://www.researchgate.net/publication/22872317

[12] J. Ma, Z. Zhao, X. Yi, J. Chen, L. Hong, E.H. Chi, et al., 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 & Data Mining, ACM, London United Kingdom, 2018, pp. 1930–1939. https://doi.org/10.1145/3219819.3220007

[13] J. Yu, Y. Dai, X. Liu, J. Huang, Y. Shen, K. Zhang, R. Zhou, E. Adhikarla, W. Ye, Y. Liu, Z. Kong, K. Zhang, Y. Yin, V. Namboodiri, B.D. Davison, J.H. Moore, Y. Chen, et al., Unleashing the power of multi-task learning: a comprehensive survey spanning traditional, deep, and pretrained foundation model eras, 2024, https://doi.org/10.48550/arXiv.2404.18961

[14] R. Velioglu, J.P. Göpfert, A. Artelt, B. Hammer, et al., Explainable artificial intelligence for improved modeling of processes, in: H. Yin, D. Camacho, P. Tino (Eds.), Intelligent Data Engineering and Automated Learning - IDEAL 2022, 13756, Springer International Publishing, Cham, 2022, pp. 313–325. Series Title: Lecture Notes in Computer Science, https://doi.org/10.1007/978-3-031-21753-1_31

[15] A. Goldstein, A. Kapelner, J. Bleich, E. Pitkin, et al., Peeking inside the black box: visualizing statistical learning with plots of individual conditional expectation, J. Comput. Graphical Stat. 24 (1) (2015) 44–65. https://doi.org/10.1080/10618600.2014.907095

[16] M.T. Ribeiro, S. Singh, C. Guestrin, et al., "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, ACM, San Francisco California USA, 2016, pp. 1135–1144. https://doi.org/10.1145/2939672.2939778

[17] S.M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, in: Advances in Neural Information Processing Systems, 30, Curran Associates, Inc., 2017. https://proceedings.neurips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html.

[18] D. Ifenthaler, Measures of similarity, in: N.M. Seel (Ed.), Encyclopedia of the Sciences of Learning, Springer US, Boston, MA, 2012, pp. 2147–2150. https://doi.org/10.1007/978-1-4419-1428-6_503

[19] C.D. Manning, P. Raghavan, H. Schütze, et al., Introduction to Information Retrieval, Cambridge University Press, 1 edition, 2008. https://doi.org/10.1017/CBO9780511809071

[20] K. Riesen, H. Bunke, Approximate graph edit distance computation by means of bipartite graph matching, Image Vis. Comput. 27 (7) (2009) 950–959. https://doi.org/10.1016/j.imavis.2008.04.004

[21] C. Leslie, E. Eskin, W.S. Noble, The spectrum kernel: a string kernel for SVM protein classification, in: Biocomputing 2002, WORLD SCIENTIFIC, Kauai, Hawaii, USA, 2001, pp. 564–575. https://doi.org/10.1142/9789812799623_0053

[22] S. Kullback, R.A. Leibler, On information and sufficiency, Ann. Math. Stat. 22 (1) (1951) 79–86. https://doi.org/10.1214/aoms/1177729694

[23] H. Alt, M. Godau, Computing the Fréchet distance between two polygonal curves, Int. J. Comput. Geom. Appl. 05 (01n02) (1995) 75–91. https://doi.org/10.1142/S0218195995000064

[24] F. Morain-Nicolier, S. Lebonvallet, E. Baudrier, S. Ruan, Hausdorff distance based 3D quantification of brain tumor evolution from MRI images, in: 2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, IEEE, Lyon, France, 2007, pp. 5597–5600. ISSN: 1557-170X, https://doi.org/10.1109/IEMBS.2007.4353615

[25] P. Cignoni, C. Rocchini, R. Scopigno, *Metro* : measuring error on simplified surfaces, Comput. Graphics Forum 17 (2) (1998) 167–174. https://doi.org/10.1111/1467-8659.00236

[26] M.M. Fréchet, Sur quelques points du calcul fonctionnel, Rendiconti del Circolo Matematico di Palermo 22 (1) (1906) 1–72. https://doi.org/10.1007/BF03018603

[27] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32. https://doi.org/10.1023/A:1010933404324

[28] A. Fisher, C. Rudin, F. Dominici, et al., All models are wrong, but many are useful: learning a variable's importance by studying an entire class of prediction models simultaneously, J. Mach. Learn. Res. 20 (177) (2019) 1–81. http://jmlr.org/papers/v20/18-760.HTML.

[29] M. Yadav, M.A. Alam, Dynamic time warping (DTW) algorithm in speech: a review, Int. J. Res. Electron. Comput. Eng. 6 (1), (2018), 524–528. https://nebula.wsimg.com/16e4a36aa249dfdaf4ee86deb401e2e6?AccessKeyId=DFB1BA3CED7E7997D5B1&disposition=0&alloworigin=1.

[30] P.W. Koh, T. Nguyen, Y.S. Tang, S. Mussmann, E. Pierson, B. Kim, P. Liang, et al., Concept bottleneck models, in: Proceedings of the 37th International Conference on Machine Learning, PMLR, 2020, pp. 5338–5348. ISSN: 2640–3498, https://proceedings.mlr.press/v119/koh20a.html.

[31] P. Lambin, R.T.H. Leijenaar, T.M. Deist, J. Peerlings, E.E.C. de Jong, J. van Timmeren, S. Sanduleanu, R.T.H.M. Larue, A.J.G. Even, A. Jochems, Y. van Wijk, H. Woodruff, J. van Soest, T. Lustberg, E. Roelofs, W. van Elmpt, A. Dekker, F.M.

Mottaghy, J.E. Wildberger, S. Walsh, Radiomics: the bridge between medical imaging and personalized medicine, Nat. Rev. Clin. Oncol. 14 (12) (2017) 749–762. https://doi.org/10.1038/nrclinonc.2017.141

[32] Y.R. Tausczik, J.W. Pennebaker, The psychological meaning of words: LIWC and computerized text analysis methods, J. Lang. Soc. Psychol. 29 (1) (2010) 24–54. https://doi.org/10.1177/0261927X09351676

[33] S. Davis, P. Mermelstein, Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences, IEEE Trans. Acoust. 28 (4) (1980) 357–366. https://doi.org/10.1109/TASSP.1980.1163420

[34] H. Chefer, S. Gur, L. Wolf, et al., Transformer interpretability beyond attention visualization, in: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Nashville, TN, USA, 2021, pp. 782–791. https://doi.org/10.1109/CVPR46437.2021.00084

[35] M.L. Athanasios Tsanas, Parkinsons telemonitoring, 2009, https://doi.org/10.24432/C5ZS3N

[36] H. Mühlenbein, M. Schomisch, J. Born, et al., The parallel genetic algorithm as function optimizer, Parallel Comput. 17 (6-7) (1991) 619–632. https://doi.org/10.1016/S0167-8191(05)80052-3