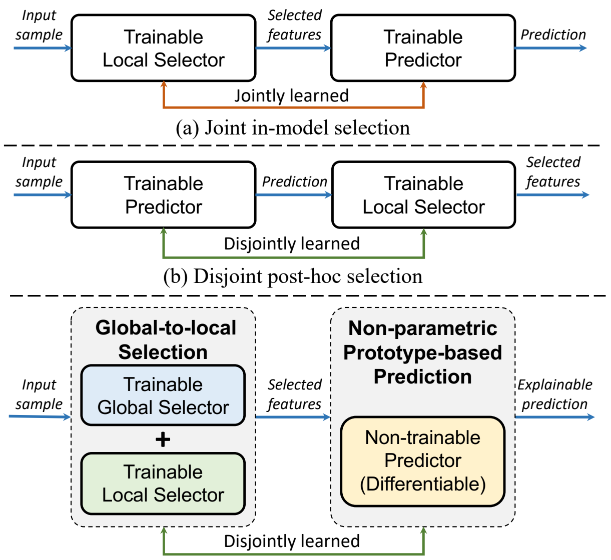
ProtoGate: Prototype-based Neural Networks with Global-to-local Feature Selection for Tabular Biomedical Data

Xiangjian Jiang [1] Andrei Margeloiu [1] Nikola Simidjievski [1 2] Mateja Jamnik [1]

Abstract

Tabular biomedical data poses challenges in machine learning because it is often high-dimensional and typically low-sample-size (HDLSS). Previous research has attempted to address these challenges via local feature selection, but existing approaches often fail to achieve optimal performance due to their limitation in identifying globally important features and their susceptibility to the co-adaptation problem. In this paper, we propose ProtoGate, a prototype-based neural model for feature selection on HDLSS data. ProtoGate first selects instance-wise features via adaptively balancing global and local feature selection. Furthermore, ProtoGate employs a non-parametric prototype-based prediction mechanism to tackle the co-adaptation problem, ensuring the feature selection results and predictions are consistent with underlying data clusters. We conduct comprehensive experiments to evaluate the performance and interpretability of ProtoGate on synthetic and real-world datasets. The results show that ProtoGate generally outperforms state-of-the-art methods in prediction accuracy by a clear margin while providing high-fidelity feature selection and explainable predictions. Code is available at https://github.com/SilenceX12138/ProtoGate.



(c) Disjoint in-model selection (ProtoGate)

Figure 1. **An overview of the proposed model.** ProtoGate introduces a novel disjoint in-model selection method. It balances global and local feature selection, and makes explainable prototypical predictions. In contrast to (a), ProtoGate integrates a trainable feature selector with a non-trainable predictor (i.e., no trainable parameters in the predictor), which allows for disjointly learned feature selector and predictor, thus mitigating the co-adaptation problem. In contrast to (b), ProtoGate makes predictions with the selected features, preserving their in-model explainability.

1. Introduction

In biomedical research, tabular data is frequently collected (Baxevanis et al., 2020; Lesk, 2019) for a wide range of applications such as detecting marker genes (Hsu et al., 2003) and performing survival analysis (Fan et al., 2022). Clinical trials, whilst collecting large amounts of high-dimensional data using modern high-throughput sequencing technologies, often consider a small number of patients due to practical reasons (Levin et al., 2022). The resulting tabular datasets are thus often high-dimensional and typically low-sample-size (HDLSS). Moreover, given the inherent heterogeneity of biomedical data, important features often vary from sample to sample – even in the same dataset (Yang et al., 2022; Yoon et al., 2018). Such scenarios have proven challenging for current machine learning approaches, including deep tabular models (Shwartz-Ziv & Armon, 2022; Yang et al., 2022; Margeloiu et al., 2023a).

Prior work (Remeseiro & Bolon-Canedo, 2019; Arik & Pfister, 2021; Chen et al., 2018; Yoon et al., 2018; Yang et al., 2022; Yoshikawa & Iwata, 2022) has attempted to address such challenges with local feature selection: rather

[1]Department of Computer Science and Technology, University of Cambridge, UK [2]Department of Oncology, University of Cambridge, UK. Correspondence to: Xiangjian Jiang <xj265@cam.ac.uk>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

Table 1. **Model design comparison between ProtoGate and prior local feature selection methods.** ProtoGate has different design rationales to the benchmark methods: a novel learning paradigm with global-to-local feature selector and prototype-based predictor.

Methods	Learning paradigms	Feature Selection		Prediction		Co-adaptation avoidance
		Global Selector	Local Selector	Explainability	Clustering assumption	
TabNet (Arik & Pfister, 2021)	Joint in-model selection	✗	✓	✗	✗	✗
L2X (Chen et al., 2018)	Joint in-model selection	✗	✓	✗	✗	✗
INVASE (Yoon et al., 2018)	Joint in-model selection	✗	✓	✗	✗	✗
LSPIN (Yang et al., 2022)	Joint in-model selection	✗	✓	✗	✗	✗
LLSPIN (Yang et al., 2022)	Joint in-model selection	✗	✓	✓	✗	✗
REAL-X (Jethani et al., 2021)	Disjoint post-hoc selection	✗	✓	✗	✗	✓
ProtoGate (Ours)	Disjoint in-model selection	✓	✓	✓	✓	✓

than selecting a general subset of features across all samples, local methods select specific subsets of features for each sample and these subsets may vary from sample to sample.

However, existing methods still have three main limitations, which are summarised in Table 1: *(i) Neglecting globally important features.* In many real-world HDLSS tasks, even simple models – such as MLP and Lasso – can outperform the advanced methods (see Table 2 for more details). We hypothesise that this is because the mainstream methods (Figure 1(a) and Figure 1(b)) only perform local feature selection without explicitly capturing globally important features, provided the competitive global methods support their existence. *(ii) Low-fidelity feature selection.* The selected features by these methods can be uninformative even when prediction accuracy is high. For instance, L2X (Chen et al., 2018) achieves 96% accuracy in digit classification on MNIST by using only one pixel as input (Jethani et al., 2021). Although REAL-X mitigates this problem via disjointly learning a predictor to prevent it from overfitting on the selected features (Figure 1(b)), it sacrifices the in-model explainability of selected features. *(iii) Insufficient explainability and inappropriate inductive bias.* In the biomedical domain, the clustering assumption, which states similar samples likely belong to the same class (Chapelle et al., 2006), has been shown effective (Kolodner, 1992; Li et al., 2018; Bichindaritz & Marling, 2006; Bichindaritz, 2008; Lu et al., 2021). However, prior work may inappropriately incorporate such inductive bias. For instance, LSPIN (Yang et al., 2022) selects similar features for similar samples, but it cannot guarantee that samples of the same class cluster together after selection (see Section 4.1 for more details). In this paper, we aim to overcome these challenges by introducing a novel global-to-local method that selects accurate and high-fidelity features with explainable predictions.

We propose ProtoGate (Figure 1(c)), a simple yet effective feature selection method for tabular biomedical data. Our approach is distinguished by three core concepts. Firstly, ProtoGate selects features in a global-to-local manner to adaptively balance global and local feature selection. Thus ProtoGate can effectively capture both global and local information across samples. Secondly, ProtoGate elegantly mitigates the co-adaptation problem through a unique learning paradigm for local feature selection: ProtoGate selects features with a non-parametric predictor. The predictor is non-trainable, and it can only evaluate whether the selected features lead to accurate predictions, rather than jointly learn to overfit the selected features for high classification accuracy. With the resistance to co-adaptation problem, ProtoGate can safely make predictions with the selected features. Therefore, ProtoGate can improve the fidelity of selected features while still preserving the in-model explainability of selected features. Thirdly, ProtoGate encodes the clustering assumption into feature selection via prototype-based prediction, which promotes feature selection that clusters samples of the same class together. This design not only confers ProtoGate an inductive bias aligned with the clustering assumption, but also the capability to make explainable prototypical predictions. In addition, we propose a hybrid sorting strategy to expedite the prototype-based prediction.

Contributions. More broadly, our contributions are: (i) We present ProtoGate, a novel feature selection method for tabular biomedical data (Section 3). The proposed method is characterised by three novelties: a global-to-local feature selection approach, a unique learning paradigm to tackle co-adaptation, and a prototype-based prediction mechanism that aligns with the clustering assumption. (ii) We show that ProtoGate generally outperforms 14 benchmark methods in seven biomedical classification tasks with fewer selected features and shorter runtimes (Section 4.1). (iii) We further show that ProtoGate can generalise well in non-biomedical domains (Section 4.1). (iv) We conduct comprehensive ablation studies to show that each component of ProtoGate makes complementary contributions to the model performance (Section 4.2). (v) We quantitatively and qualitatively demonstrate that ProtoGate can provide robust interpretability for the selected features and predictions (Section 4.3). (vi) We provide an open-source implementation of ProtoGate to facilitate future research and applications.

2. Related Work

Local Feature Selection. Recent work attends to the heterogeneity across samples by selecting instance-wise features for making predictions (Chen et al., 2018; Arik & Pfister, 2021; Jethani et al., 2021; Yoon et al., 2018; Yang et al., 2022; Yoshikawa & Iwata, 2022). Its main focus is on designing differentiable optimisation to generate sparse masks for feature selection. L2X uses mutual information with Concrete distribution (Chen et al., 2018), and INVASE models each feature's mask value with independent Bernoulli distributions (Yoon et al., 2018). Yamada et al. (2020) propose a continuous relaxation of discrete random variables, which was further extended to the exact formulation by LSPIN (Yang et al., 2022). In this work, we employ this exact formulation for computing differentiable ℓ_0-regularisation Table 1 and Appendix A show that there have been few mechanisms to explicitly promote globally important features in local feature selection. Instead, ProtoGate proposes to address the gap by adaptively balancing global and local feature selection, leading to flexible feature selection behaviours across different datasets.

Co-adaptation Problem. In local feature selection, co-adaptation refers to the situation where the model achieves high prediction accuracy with low-fidelity features (Jethani et al., 2021; Adebayo et al., 2018; Hooker et al., 2019; Samek et al., 2016). Jethani et al. (2021) prove that co-adaptation stems from jointly learning a trainable selector and a trainable predictor, where the predictor learns to overfit on poorly selected features for high accuracy. Post-hoc explanation methods (Figure 1(b)) address the problem by training a local selector to explain a pre-trained predictor. Instead, we propose to mitigate the problem via a non-trainable predictor. In ProtoGate, the predictor has no trainable parameters and cannot adapt to the feature selector for high accuracy, eliminating the possibility of co-adaptation while preserving the in-model explainability.

Prototype-based Machine Learning. Prototype-based models (Biehl et al., 2016) in machine learning are closely related to metric learning (Goldberger et al., 2004) and case-based reasoning (Kolodner, 1992). They are built upon the clustering assumption and aim to represent data through prototypical exemplars (e.g., KNN (Fix, 1985)) or a set of prototypical centroids (e.g., k-means (Ball & Hall, 1965)) that capture the fundamental characteristics of the data. These core ideas parallel similar concepts from cognitive psychology and neurosciences, and it is a pervasive behaviour in everyday human problem-solving (Kolodner, 1992; Li et al., 2018; Bichindaritz & Marling, 2006; Bichindaritz, 2008; Lu et al., 2021). Thus, in ProtoGate we leverage the efficacy of prototype-based prediction for feature selection, leading to more generally applicable inductive biases and improved explainability in predictions.

3. Method

Figure 2 illustrates the architecture of ProtoGate. We first describe our problem setup (Section 3.1). Then we present the core components of ProtoGate: a global-to-local feature selection approach (Section 3.2), and a non-parametric prototype-based prediction mechanism (Section 3.3). Furthermore, we define a disjoint objective function to optimise ProtoGate end-to-end (Section 3.4). We present the pseudocode for training and inference in Appendix B.

3.1. Problem Setup

We consider the classification task on tabular biomedical data with $\mathcal{Y}$ classes. Let $X := \left[\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}\right]^\top \in \mathbb{R}^{N \times D}$ be the data matrix consisting of N samples $\boldsymbol{x}^{(i)} \in \mathbb{R}^D$ with D features, and let $Y := \left[y^{(1)}, \ldots, y^{(N)}\right]^\top \in \mathbb{R}^{N \times 1}$ be the corresponding labels. We denote $x_d^{(i)}$ as the d-th feature of the i-th sample. To simplify the notation, we assume all samples in X are used for training. We mainly consider the HDLSS datasets, where the number of features is much greater than the number of samples (i.e., $D \gg N$).

3.2. Global-to-local Feature Selection

Our proposed global-to-local feature selection is implemented via a gating network $S_{\mathbf{W}}\colon \mathbb{R}^D \to [0,1]^D$ that sequentially performs soft global selection and local selection (Figure 2A). It takes as input a sample $\boldsymbol{x}^{(i)}$ and generates a mask (also referred to as "gate") $\boldsymbol{s}_{\text{local}}^{(i)} := [s_1^{(i)}, \ldots, s_D^{(i)}] \in [0,1]^D$ for local feature selection. The d-th feature is selected if and only if the mask value is positive ($s_d^{(i)} > 0$) (Figure 2B).

Soft Global Selection. ProtoGate first captures the globally important features via applying ℓ_1-regularisation on $\mathbf{W}^{[1]}$, the weights of the first layer in the gating network. The regularisation promotes sparsity in $\mathbf{W}^{[1]}$ via reducing some weights to zero. When all weights connected to the same input neuron are reduced to zero, the corresponding input feature is dropped (e.g., x_2 and x_3 in Figure 2A). Mathematically, the sparsity in $\mathbf{W}^{[1]}$ transforms the first layer's computation as $\mathbf{W}^{[1]}\boldsymbol{x}^{(i)} = \mathbf{W}^{[1]}(\boldsymbol{x}^{(i)} \odot \boldsymbol{s}_{\text{global}}^{(i)})$, where $\odot$ represents element-wise multiplication and $\boldsymbol{s}_{\text{global}}^{(i)} \in \{0,1\}^D$ is a binary mask, signifying dropped features as zeros. The first layer is shared across samples, and thus the mask is consistent for all samples (i.e., $\boldsymbol{s}_{\text{global}} = \boldsymbol{s}_{\text{global}}^{(i)} = \boldsymbol{s}_{\text{global}}^{(j)}$). Furthermore, the global selection by $\mathbf{W}^{[1]}$ is "soft" because ProtoGate uses $\boldsymbol{s}_{\text{global}}$ to explicitly emphasise the existence of globally important features, serving as a foundation rather than replacing subsequent local feature selection. Specifically, the initially dropped features can be recovered in the following local selection process (e.g., x_3 in Figure 2A). We further illustrate the interplay between soft global and local selection with a real-world example in Appendix C.

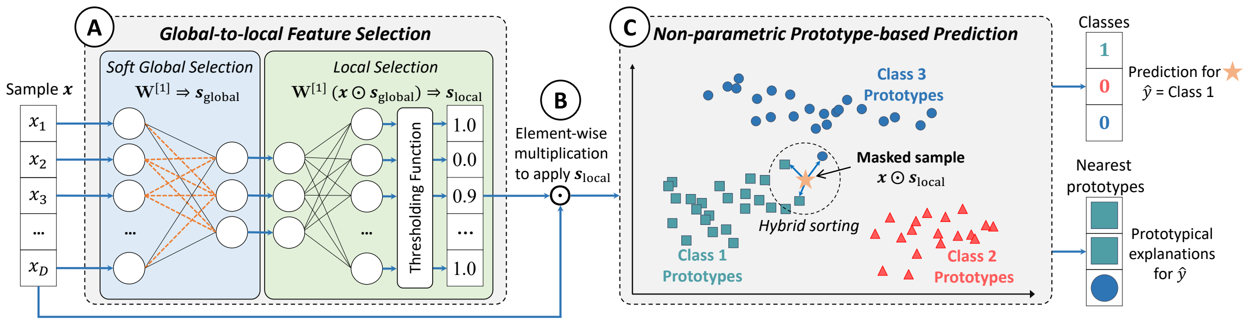

A
Global-to-local Feature Selection
Soft Global Selection
$\mathbf{W}^{[1]} \Rightarrow \boldsymbol{s}_{\text{global}}$
Local Selection
$\mathbf{W}^{[1]} (\boldsymbol{x} \odot \boldsymbol{s}_{\text{global}}) \Rightarrow \boldsymbol{s}_{\text{local}}$
Sample $\boldsymbol{x}$
x_1
x_2
x_3
...
x_D
Thresholding Function
1.0
0.0
0.9
...
1.0
B
Element-wise multiplication to apply $\boldsymbol{s}_{\text{local}}$
C
Non-parametric Prototype-based Prediction
Class 3 Prototypes
Masked sample $\boldsymbol{x} \odot \boldsymbol{s}_{\text{local}}$
Hybrid sorting
Class 1 Prototypes
Class 2 Prototypes
Classes
1
0
0
Prediction for ★ $\hat{y}$ = Class 1
Nearest prototypes
Prototypical explanations for $\hat{y}$

Figure 2. **The architecture of ProtoGate. (A)** Given a sample $\boldsymbol{x} \in \mathbb{R}^D$, the global-to-local feature selection performs soft global feature selection in the first layer of the gating network. The orange dashed lines denote sparsified weights (i.e., reduced to zero) in $\mathbf{W}^{[1]}$ under ℓ_1-regularisation. The neural network then computes the instance-wise mask $\{s_d\}_{d=1}^D \in [0,1]^D$ with a thresholding function for local feature selection. **(B)** The local mask $\boldsymbol{s}_{\text{local}}$ is applied to the sample for local feature selection by element-wise multiplication. **(C)** The non-parametric prototype-based prediction further classifies $\boldsymbol{x} \odot \boldsymbol{s}_{\text{local}}$ by retrieving the K nearest prototypes in base $\mathcal{B}$ via hybrid sorting. The majority class is used as the predicted label $\hat{y}$, and the exemplars (i.e., the nearest prototypes) provide prototypical explanations.

Local Selection. After the initial selection of globally important features, ProtoGate further refines $\boldsymbol{s}_{\text{global}}$ to generate instance-wise masks $\boldsymbol{s}_{\text{local}}^{(i)}$ for local feature selection. This stage involves processing the embedded output $\mathbf{W}^{[1]}(\boldsymbol{x}^{(i)} \odot \boldsymbol{s}_{\text{global}})$ through subsequent network layers, yielding the vector $\boldsymbol{\mu}^{(i)}$. Given the range of $\boldsymbol{\mu}^{(i)}$ is not necessarily $[0,1]^D$, we threshold it to compute the instance-wise masks and then apply ℓ_0-regularisation to promote sparse masks. However, the non-differentiability of $||\boldsymbol{s}_{\text{local}}^{(i)}||_0$ poses a challenge for efficient optimisation via backpropagation. Building on prior work (Louizos et al., 2018; Rolfe, 2016; Yamada et al., 2020; Yang et al., 2022), we re-formalise the local mask as a vector of random variables, defined by $\boldsymbol{s}_{\text{local}}^{(i)} = \max(0, \min(1, \boldsymbol{\mu}^{(i)} + \boldsymbol{\epsilon}^{(i)}))$, where $\boldsymbol{\epsilon}^{(i)}$ is Gaussian noise sampled from $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$. This formulation allows for the differentiable estimation of ℓ_0-regularisation through its expectation (i.e., $||\boldsymbol{s}_{\text{local}}^{(i)}||_0 \propto \mathbb{E}[||\boldsymbol{s}_{\text{local}}^{(i)}||_0]$). In addition, the thresholded mask values are continuous (i.e., $\boldsymbol{s}_{\text{local}}^{(i)} \in [0,1]^D$), which are considered to be more expressive than binary masks (Gros et al., 2021).

The integration of these two phases is encapsulated in ProtoGate's weighted sparsity regularisation for feature selection

$$\mathcal{R}_{\text{select}} := \mathcal{R}_{\text{global}}(\boldsymbol{s}_{\text{global}}) + \mathcal{R}_{\text{local}}(\boldsymbol{s}_{\text{local}}^{(i)}) \\ \propto \lambda_{\text{global}} \left\| \mathbf{W}^{[1]} \right\|_1 + \lambda_{\text{local}} \sum_{d=1}^{D} \int_{\hat{\mu}_d^{(i)}}^{\infty} \exp(-\frac{t^2}{2}) dt \tag{1}$$

where $(\lambda_{\text{global}}, \lambda_{\text{local}})$ is a pair of hyperparameters to balance the regularisation strength between global and local feature selection, and $\hat{\mu}_d^{(i)}$ is the standardised output of $S_{\mathbf{W}}$. Comprehensive theoretical justifications are in Appendix D.

Complementary regularisations. ProtoGate employs different regularisations for its two-stage selection process to address distinct objectives. Firstly, the soft global selection focuses on efficiently identifying a globally important lower-dimension feature set (i.e. $\mathbf{s}_{\text{global}}$). Therefore, ProtoGate promotes sparsity in $\mathbf{s}_{\text{global}}$ via the faster ℓ_1-regularisation because it does not require the computation of complex surrogates like ℓ_0-regularisation. Secondly, the local selection aims to identify locally important features for accurate predictions and high interpretability, which prioritises selecting fewer features (i.e., $\mathbf{s}_{\text{local}}^{(i)}$ contains more "exact zeros"). Note that small mask values (e.g., 0.1) still signify a selected feature. Therefore, we choose ℓ_0-regularisation for local selection because it generally leads to more exact zeros (Louizos et al., 2018). Note that, while ℓ_1-regularisation can also promote exact zeros with greater regularisation strength, we find that it can detrimentally impact model performance, aligning with findings from previous study (Louizos et al., 2018).

3.3. Non-parametric Prototype-based Prediction

The non-parametric prototype-based prediction (Figure 2C) is implemented via a differentiable k-Nearest Neighbours (KNN), denoted as $F_\theta : \mathbb{R}^D \to \mathcal{Y}$, which takes as input the masked sample $\boldsymbol{x}^{(i)} \odot \mathbf{s}_{\text{local}}^{(i)}$ and predicts its label $\hat{y}^{(i)}$. Note that θ here refers to hyperparameters for a non-parametric model, rather than trainable parameters as $\mathbf{W}$ of $S_{\mathbf{W}}$.

During the training period, ProtoGate first constructs a prototype base $\mathcal{B}$ with the masked training samples. Recall that we assume all $x^{(i)}$ are used as training samples. Thus ProtoGate retains the pairs of the masked sample and its label $p^{(i)} := \left(\boldsymbol{x}^{(i)} \odot \mathbf{s}_{\text{local}}^{(i)}, y^{(i)}\right)$ as prototypes in the base. With the acquired prototypes, ProtoGate can classify a randomly picked query sample $\boldsymbol{x}_{\text{query}} \in X$ by retrieving similar prototypes in the base $\mathcal{B}$. Specifically, ProtoGate sorts the prototypes by their similarities (i.e., Euclidean distance) to the masked query sample and makes predictions according to the majority class of the K nearest prototypes. Technically, during training, we use the masked samples from the same batch as prototypes; during inference, we use the masked samples from the whole training set as prototypes.

Hybrid Sorting. The standard sorting operation (e.g., QuickSort (Hoare, 1961)) is non-differentiable and the model cannot be optimised via backpropagation. In contrast, the differentiable sorting operation (e.g., NeuralSort (Grover et al., 2018)) allows backpropagation, but the high time complexity can be a bottleneck. With these in mind, we take a step further and propose *hybrid* sorting. We integrate two kinds of sorting operations to amalgamate their strengths: ProtoGate first trains with NeuralSort for backpropagating gradients from the sorted prototypes, and then substitutes NeuralSort for QuickSort during inference. During the training phase, given a random training sample as query sample $\boldsymbol{x}_{\text{query}} := \boldsymbol{x}^{(i)} \in X$, we sort the base $\mathcal{B}$ via computing a row-stochastic matrix $\widehat{\mathbf{P}} \in \mathbb{R}^{N \times N}$, and $\widehat{\mathbf{P}}[n, m]$ denotes the probability that the m-th prototype in $\mathcal{B}$ is the n-th nearest to query sample. Given $n \sim \mathcal{U}\{1, K\}$, we can improve the accuracy via $\max \mathbb{E}_{\widehat{\mathbf{P}}[n,:]}[\mathbb{1}(y^{(m)} = y_{\text{query}})]$ where $\widehat{\mathbf{P}}[n, :]$ is the n-th row of $\widehat{\mathbf{P}}$, and $\mathbb{1}(\cdot)$ is the indicator function (i.e., maximising the number of prototypes that have the same label as $\boldsymbol{x}_{\text{query}}$). Therefore, the prediction loss per query sample is given by:

$$\mathcal{L}_{\text{pred}} := K - \sum_{n=1}^{K} \mathbb{E}_{\widehat{\mathbf{P}}[n,:]} \left[\mathbb{1}(y^{(m)} = y_{\text{query}}) \right] \tag{2}$$

where $y_{\text{query}} = y^{(i)}$. Note that we set $m \neq i$ to avoid using x_{query} itself as a prototype. Equation (2) estimates the number of prototypes that have different labels to $\boldsymbol{x}_{\text{query}}$ among the K nearest prototypes (we provide theoretical analysis in Appendix D). By optimising the gating network, ProtoGate learns to generate local masks that cluster samples of the same class together. During inference, the prototypes are fixed and query samples are from unseen test data. We find that the hybrid strategy decreases the inference time by almost half while preserving the identical predictive performance as only using differentiable sorting (Appendix H.3).

Differentiability vs. Non-trainability. The two concepts are not contradictory. For hybrid sorting, the differentiability allows backpropagating gradients from sorted prototypes. On the other hand, its non-parametric nature leads to a non-trainable predictor F_θ that consistently evaluates whether the masked samples align with the clustering assumption. And F_θ cannot overfit on $\boldsymbol{s}_{\text{local}}^{(i)}$ from $S_{\mathbf{W}}$ through training, leading to disjointly learned $S_{\mathbf{W}}$ and F_θ. Therefore, ProtoGate can be resistant to the co-adaptation problem.

3.4. Disjoint Training Loss

We further incorporate the core components into the estimator of ProtoGate's training loss:

$$\mathcal{L}_{\text{total}}(\mathbf{W}) := \mathbb{E}_{(X,Y)} \left[\mathcal{L}_{\text{pred}} + \mathcal{R}_{\text{select}} \right]. \tag{3}$$

We can see that the gating network $S_{\mathbf{W}}$ is the only trainable component, denoting disjointly learned feature selector and predictor (i.e., ProtoGate's training objective only needs to optimise $\mathbf{W}$). Moreover, the training loss is fully differentiable, allowing ProtoGate to be optimised end-to-end in a single stage with standard gradient-based approaches.

4. Experiments

We evaluate ProtoGate by answering three questions:

Performance (Q1): Can ProtoGate achieve competitive accuracy in classification and sparsity in feature selection against those of the benchmark methods?

Ablation Impacts (Q2): What contributions to model performance do individual components of ProtoGate make?

Interpretability (Q3): Can ProtoGate improve the interpretability of selected features and predictions?

To answer these questions, we first compare ProtoGate against 16 benchmark methods on real-world classification tasks (Section 4.1). Next, we investigate the effectiveness of global-to-local feature selection and non-parametric prototype-based prediction with ablation studies (Section 4.2). Finally, we quantitatively and qualitatively evaluate ProtoGate's interpretability by analysing the selected features and predictions (Section 4.3).

Real-world datasets. We use seven open-source HDLSS tabular biomedical datasets (Margeloiu et al., 2023a), which contain 2,000-5,966 features with 62-197 samples of 2-4 different classes. We also select four challenging non-HDLSS and non-biomedical tabular datasets from the TabZilla benchmark (McElfresh et al., 2023), which contain 19-857 features with 846-2,000 samples of 4-100 classes. Full descriptions are in Appendix E.1.

Synthetic datasets. We generate three challenging synthetic datasets (Syn1, Syn2 and Syn3) by adapting those in prior studies (Yang et al., 2022; Yoon et al., 2018; Jethani et al., 2021). In our settings, each dataset has 200 samples of 100 features, which is only 10% of the samples and 10 times more features compared to Yang et al. (2022). Each dataset has two classes, and we introduce an imbalance by generating 50 and 150 samples, respectively. The exact synthetic datasets are described in Appendix E.2.

Benchmark methods. We compare ProtoGate against 16 benchmark methods, including four standard baselines: Ridge Regression (Ridge) (Cox, 1958), SVM (Cortes & Vapnik, 1995), KNN (Fix, 1985) and MLP (Gorishniy et al., 2021); six global feature selection methods (also referred to as "global methods"): Lasso (Tibshirani, 1996), Random Forest (RF) (Breiman, 2001), XGBoost (Chen & Guestrin, 2016), CatBoost (Prokhorenkova et al., 2018), LightGBM (Ke et al., 2017) and STG (Yamada et al., 2020); six local feature selection methods (also referred to as "local methods"): TabNet (Arik & Pfister, 2021), L2X (Chen et al., 2018), INVASE (Yoon et al., 2018), REAL-X (Jethani et al., 2021) and LSPIN/LLSPIN (Yang et al., 2022).

Table 2. **Classification accuracy (%) on seven HDLSS real-world tabular datasets.** We report the mean ± std balanced accuracy and average accuracy rank across datasets. A higher rank implies higher accuracy. Note that "−" denotes failed convergence, and the rank is computed with the balanced accuracy of other methods. We highlight the **First**, **Second** and **Third** ranking accuracy for each dataset. ProtoGate consistently ranks Top-3 across datasets and achieves the best overall performance.

	Methods	colon	lung	meta-dr	meta-pam	prostate	tcga-2y	toxicity	**Rank**
Baseline	Ridge	77.50$_{\pm 9.43}$	92.77$_{\pm 6.45}$	**59.19**$_{\pm 9.78}$	92.16$_{\pm 7.69}$	87.22$_{\pm 8.56}$	57.73$_{\pm 5.83}$	**91.88**$_{\pm 5.49}$	6.71$_{\pm 2.75}$
	SVM	70.75$_{\pm 13.93}$	72.77$_{\pm 8.33}$	50.64$_{\pm 2.24}$	61.46$_{\pm 7.65}$	85.75$_{\pm 6.63}$	52.63$_{\pm 4.02}$	66.75$_{\pm 7.86}$	15.14$_{\pm 1.68}$
	KNN	71.65$_{\pm 12.03}$	91.06$_{\pm 5.41}$	54.64$_{\pm 7.92}$	82.79$_{\pm 7.95}$	78.78$_{\pm 9.20}$	58.83$_{\pm 6.71}$	83.86$_{\pm 7.07}$	10.86$_{\pm 3.98}$
	MLP	80.00$_{\pm 8.70}$	**96.47**$_{\pm 2.69}$	57.89$_{\pm 9.63}$	**96.17**$_{\pm 2.59}$	87.22$_{\pm 7.41}$	**60.18**$_{\pm 7.24}$	**94.48**$_{\pm 4.28}$	**3.86**$_{\pm 3.34}$
Global	Lasso	79.40$_{\pm 10.18}$	**94.47**$_{\pm 4.39}$	58.58$_{\pm 9.17}$	95.15$_{\pm 2.83}$	**91.18**$_{\pm 6.39}$	56.99$_{\pm 5.21}$	91.86$_{\pm 6.03}$	**5.00**$_{\pm 2.89}$
	RF	**80.05**$_{\pm 10.37}$	91.73$_{\pm 6.61}$	51.48$_{\pm 3.41}$	88.73$_{\pm 6.24}$	90.38$_{\pm 7.31}$	58.70$_{\pm 6.84}$	79.78$_{\pm 7.10}$	8.71$_{\pm 5.02}$
	XGBoost	72.60$_{\pm 12.59}$	86.61$_{\pm 8.72}$	58.09$_{\pm 8.65}$	92.65$_{\pm 5.40}$	82.55$_{\pm 10.22}$	55.91$_{\pm 6.97}$	70.13$_{\pm 7.85}$	11.71$_{\pm 2.69}$
	CatBoost	72.65$_{\pm 10.12}$	91.57$_{\pm 5.74}$	57.08$_{\pm 5.74}$	95.93$_{\pm 4.39}$	90.24$_{\pm 6.87}$	55.09$_{\pm 7.52}$	81.95$_{\pm 7.47}$	9.00$_{\pm 3.56}$
	LightGBM	76.60$_{\pm 11.67}$	93.42$_{\pm 5.91}$	58.23$_{\pm 8.56}$	94.98$_{\pm 5.19}$	**91.38**$_{\pm 5.71}$	57.09$_{\pm 7.87}$	81.98$_{\pm 6.25}$	6.43$_{\pm 3.21}$
	STG	79.55$_{\pm 10.53}$	93.30$_{\pm 6.28}$	58.15$_{\pm 8.67}$	76.13$_{\pm 8.19}$	89.38$_{\pm 5.85}$	57.04$_{\pm 5.76}$	87.95$_{\pm 5.01}$	7.43$_{\pm 3.60}$
Local	TabNet	56.75$_{\pm 15.20}$	80.14$_{\pm 12.23}$	53.87$_{\pm 7.31}$	82.66$_{\pm 11.56}$	66.55$_{\pm 15.33}$	52.16$_{\pm 8.20}$	41.68$_{\pm 9.03}$	15.29$_{\pm 1.60}$
	L2X	57.60$_{\pm 13.48}$	50.02$_{\pm 14.26}$	52.54$_{\pm 8.30}$	62.64$_{\pm 13.75}$	61.78$_{\pm 13.69}$	52.30$_{\pm 6.29}$	31.72$_{\pm 9.11}$	14.29$_{\pm 0.70}$
	INVASE	–	91.22$_{\pm 6.16}$	–	91.70$_{\pm 6.84}$	–	55.98$_{\pm 6.45}$	79.94$_{\pm 6.60}$	11.29$_{\pm 0.95}$
	REAL-X	76.75$_{\pm 12.21}$	93.27$_{\pm 4.32}$	**60.01**$_{\pm 7.12}$	95.59$_{\pm 3.04}$	86.75$_{\pm 6.68}$	**59.30**$_{\pm 7.49}$	90.79$_{\pm 4.75}$	5.86$_{\pm 3.18}$
	LSPIN	**81.30**$_{\pm 7.97}$	76.92$_{\pm 9.38}$	53.98$_{\pm 8.00}$	**97.18**$_{\pm 3.16}$	87.75$_{\pm 6.74}$	55.95$_{\pm 7.45}$	83.47$_{\pm 8.59}$	8.29$_{\pm 5.19}$
	LLSPIN	79.35$_{\pm 7.74}$	70.10$_{\pm 12.31}$	56.77$_{\pm 9.65}$	95.50$_{\pm 3.60}$	88.71$_{\pm 5.98}$	57.88$_{\pm 6.02}$	81.67$_{\pm 9.01}$	9.00$_{\pm 3.65}$
	ProtoGate (Ours)	**83.95**$_{\pm 9.82}$	**93.44**$_{\pm 6.37}$	**60.43**$_{\pm 7.62}$	**95.96**$_{\pm 3.93}$	**90.58**$_{\pm 5.72}$	**61.18**$_{\pm 6.47}$	**92.34**$_{\pm 5.67}$	**2.00**$_{\pm 1.00}$

Experimental setup. For each dataset, we use 5-fold cross-validation and repeat it for five times, summing up to 25 runs. For the training fold, we randomly select 10% of samples as the validation set. For each model and each dataset, we tune its hyperparameters according to the aggregated performance on validation sets across 25 runs. Our setup with validation sets leads to less training data, but it is more realistic than those in prior work (Yang et al., 2022), where *no* validation sets are used for model selection on "colon" and "toxicity" datasets. Note that the reported results are averaged over 25 runs on test sets by default. When aggregating results across datasets, we use the average distance to the minimum (ADTM) metric via affine renormalisation between the top-performing and worse-performing models (Grinsztajn et al., 2022). Full experimental details are in Appendix E.

4.1. Prediction on Real-world Datasets (Q1)

We compare ProtoGate against benchmark methods in real-world scenarios. We evaluate the models from four aspects: (i) classification accuracy, (ii) feature selection sparsity, (iii) computation efficiency and (iv) model generalisability.

Classification accuracy. We evaluate the performance in real-world classification tasks via *balanced accuracy* and the corresponding *rank*. Typically, higher classification accuracy denotes that the selected features are more informative.

Table 2 shows that ProtoGate exhibits robust performance in real-world HDLSS tasks, with notably high results on "colon", "meta-dr" and "tcga-2y" datasets. It consistently ranks in the top three for balanced accuracy across datasets, outperforming the benchmark methods in average rank. In particular, ProtoGate surpasses the best-performing benchmark local method, REAL-X, by a clear margin: from 5.86→2.00 in average rank. In addition, ProtoGate provides *in-model* explainability of selected features, while REAL-X only provides *post-hoc* explainability. Although other local methods can achieve high classification accuracy (e.g., LSPIN on "meta-pam" dataset), they are susceptible to co-adaptation problem and hence the interpretability of their selected features cannot be fairly compared to ProtoGate's (see Section 4.3). Another important finding is that simple Lasso and MLP can outperform local methods on HDLSS datasets, but not ProtoGate.

The results suggest that ProtoGate can be a robust feature selection method for HDLSS tasks. We attribute ProtoGate's competitive performance to three main reasons. Firstly, ProtoGate takes advantage of both global and local selection. ProtoGate selects instance-wise features *after* explicitly capturing the globally important features, which adaptively balances global and local feature selection. Secondly, ProtoGate meticulously attends to the similarity in high-dimensional space. For instance, LSPIN has constraints to generate similar masks for similar samples in the original input space. However, the poor accuracy of KNN shows that the similarity between samples *before* feature selection can be misleading for predictions, thus harming the model performance. In contrast, ProtoGate inherently leverages the similarity between samples *after* feature selection. Furthermore, ProtoGate is better aligned with the clustering assumption because it encourages samples of the same class

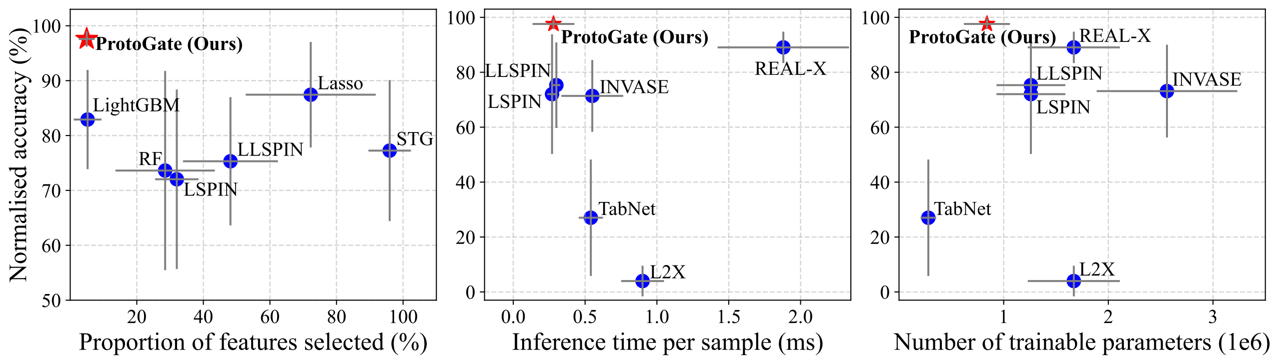


Figure 3. **Left:** Mean normalised feature selection sparsity vs. mean normalised balanced accuracy. We exclude the outliers (TabNet, L2X and INVASE) due to their suboptimal results or failed convergence. **Middle:** Median runtime vs. mean normalised balanced accuracy. **Right:** Median model size vs. mean normalised balanced accuracy. ProtoGate generally achieves higher accuracy and fewer selected features with higher computation efficiency than other local methods.

to have similar representations, *not* similar masks. Thirdly, ProtoGate is parameter-efficient. In contrast to other local methods that can easily be over-parameterised due to complex network-based architectures, ProtoGate has a sparsified gating network and a non-parametric predictor, which notably reduces the number of trainable parameters.

Feature selection sparsity. We evaluate the sparsity via *proportion of features selected per sample* for in-model selection methods. In Figure 3 (Left), we plot the "sparsity vs. accuracy" by aggregating the results across datasets. The numerical results and visualisation of selected features are in Appendix F.1. Overall, ProtoGate stably achieves higher accuracy with a lower proportion of features selected per sample than benchmark methods, suggesting that the features selected by ProtoGate are easier to interpret by humans.

Feature selection behaviour. We further analyse the behaviour of ProtoGate via the *composition of the selected features* (i.e., the proportions of "both selected" and "locally recovered" features in $\mathbf{s}_{\text{local}}^{(i)}$). We find that ProtoGate exhibits adaptive feature selection behaviour. On most datasets (e.g., the "colon" dataset), the majority proportion of $\mathbf{s}_{\text{local}}^{(i)}$ is generally included in $\mathbf{s}_{\text{global}}$ (i.e., the proportion of "both selected" features generally exceeds that of "locally recovered" features), suggesting ProtoGate can behave more globally in feature selection. In contrast, on the other datasets (e.g., the "meta-dr" dataset), ProtoGate tends to recover more features locally than reusing the globally selected features, highlighting a more pronounced local selection strategy. Full results and discussion are in Appendix F.1.3 and Appendix G.2.

Computation efficiency. We evaluate the speed and model size via *inference time* and *number of trainable parameters*. In Figure 3 (Middle), the trade-off between accuracy and inference speed shows ProtoGate excels benchmark local methods in maintaining high accuracy while keeping the inference time short. In Figure 3 (Right), the trade-off between accuracy and model size further shows ProtoGate is more parameter-efficient than benchmark local methods. More results on computation efficiency are in Appendix F.2. The results suggest that ProtoGate can be a highly efficient local method for HDLSS tasks.

Model generalisability. We further evaluate classification performance on four non-HDLSS and non-biomedical datasets. Although ProtoGate is designed for the HDLSS regime, the evaluation results show that ProtoGate generalises well on non-HDLSS real-world datasets, which contain a much larger number of samples and classes than the considered HDLSS datasets. Specifically, ProtoGate can achieve competitive accuracy against benchmark methods, including the gradient boosting trees. Detailed results and discussion are in Appendix F.3.

4.2. Ablation Studies (Q2)

We investigate the efficacy of ProtoGate's two main components: (i) Global-to-local Feature Selection and (ii) Non-parametric Prototype-based Prediction.

Soft global selection and local selection are complementary, rather than interchangeable. We illustrate how ProtoGate balances global and local feature selection by performing ablation experiments on the interplay between λ_{global} and λ_{local}. We compare ProtoGate ($\lambda_{\text{global}} \neq 0, \lambda_{\text{local}} \neq 0$) against two variants: (i) remove $\mathcal{R}_{\text{local}}(\boldsymbol{s}_{\text{local}}^{(i)})$: ProtoGate-global ($\lambda_{\text{global}} \neq 0, \lambda_{\text{local}} = 0$) and (ii) remove $\mathcal{R}_{\text{global}}(\boldsymbol{s}_{\text{global}})$: ProtoGate-local ($\lambda_{\text{global}} = 0, \lambda_{\text{local}} \neq 0$). Detailed experimental setup is in Appendix G.

Table 3 shows that ProtoGate consistently outperforms its variants in accuracy, suggesting that either removing global selection (ProtoGate-local) or removing local selection (ProtoGate-global) leads to performance degradation. Therefore, the two stages are complementary for more accurate selection results.

Table 3. **Classification accuracy (%) of ProtoGate and its two variants.** We **bold** the highest accuracy for each dataset. ProtoGate consistently achieves the best accuracy across all datasets, showing the complementary effects of global and local selection.

Methods	colon	lung	meta-dr	meta-pam	prostate	tcga-2y	toxicity
ProtoGate-global	$75.95_{\pm 11.63}$	$90.20_{\pm 6.67}$	$50.66_{\pm 4.08}$	$92.80_{\pm 6.00}$	$89.02_{\pm 5.38}$	$56.46_{\pm 6.15}$	$70.27_{\pm 10.31}$
ProtoGate-local	$81.45_{\pm 10.87}$	$90.51_{\pm 8.36}$	$58.88_{\pm 8.88}$	$92.60_{\pm 6.07}$	$89.58_{\pm 5.96}$	$59.26_{\pm 6.34}$	$87.44_{\pm 6.84}$
ProtoGate (Global-to-local)	$\mathbf{83.95}_{\pm 9.82}$	$\mathbf{93.56}_{\pm 6.29}$	$\mathbf{60.43}_{\pm 7.62}$	$\mathbf{95.96}_{\pm 3.93}$	$\mathbf{90.58}_{\pm 5.72}$	$\mathbf{61.18}_{\pm 6.47}$	$\mathbf{92.34}_{\pm 5.67}$

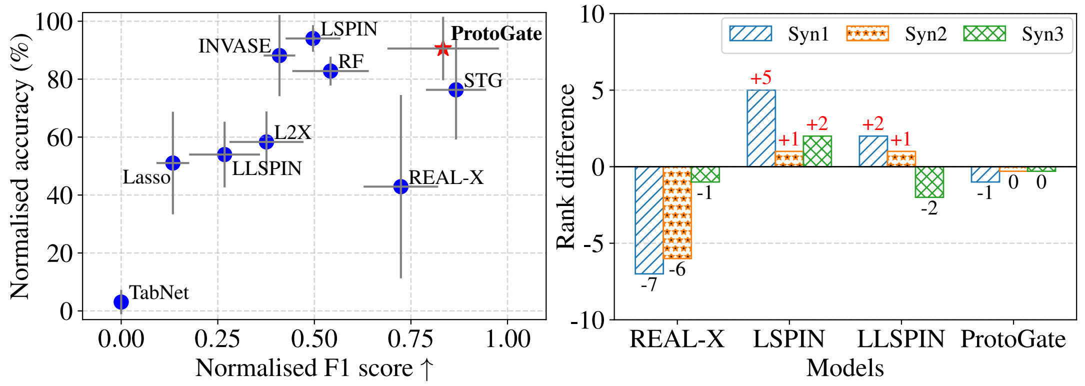


Figure 4. **Fidelity evaluation of selected features on three synthetic datasets. Left:** Mean normalised F1 score of selected features ($F1_{select}$) vs. mean normalised balanced accuracy (ACC_{pred}). **Right:** Rank difference between $F1_{select}$ and ACC_{pred}. A positive value (highlighted in red) indicates low-fidelity feature selection. Note that we plot short bars at zeros for visual clearance. ProtoGate has competitive trade-off between $F1_{select}$ and ACC_{pred}, and consistently non-positive rank differences, showing robustness to co-adaptation.

Table 3 further shows the effects of explicitly leveraging global information in feature selection. Note that we do not claim that the purely local methods, such as ProtoGate-local, omit global information completely. Indeed, they can promote global selection by increasing the regularisation strength for local selection to render smaller feature sets per sample. However, the ablation studies on feature selection sparsity (Appendix G) show that even when ProtoGate and ProtoGate-local achieve similar sparsity, the accuracy gap remains. This demonstrates that ProtoGate selects different features to ProtoGate-local (we will also see this in Figure 5). On the other hand, they can also reduce the regularisation strength to render larger feature sets per sample (i.e., increasing the likelihood of selecting the same features for different samples). However, larger feature sets do *not* guarantee that the additionally selected "common" features are informative, leading to lower accuracy. In contrast, ProtoGate explicitly promotes global information by applying ℓ_1-regularisation on $\mathbf{W}^{[1]}$ and thus generates smaller feature sets per sample. Therefore, the soft global selection and local selection are complementary to help ProtoGate exhibit unique feature selection behaviours.

Non-parametric prototype-based prediction is appropriate for biomedical tasks. We investigate how the prototype-based predictor impacts classification accuracy. Specifically, we compare ProtoGate against two variants with different predictors and investigate different numbers of different nearest neighbours. The further studies (Appendix H) show the efficacy and robustness of non-parametric prototype-based prediction.

4.3. Interpretability Evaluation (Q3)

We evaluate ProtoGate's interpretability by focusing on three aspects of the selected features and predictions: (i) fidelity of selected features, (ii) transferability of selected features and (iii) explainability of predictions.

Note that our criteria are more realistic and comprehensive than prior work (Alvarez Melis & Jaakkola, 2018; Jethani et al., 2021; Yang et al., 2022). In addition to fidelity and transferability of selected features, we also evaluate the explainability of predictions for local feature selection methods. More discussion on the metrics for interpretability evaluation is in Appendix I.1.

Fidelity of selected features. This criterion focuses on *"Can the classification accuracy denote feature selection performance, i.e., does the model achieve high accuracy with poorly selected features?"*, and we quantitatively evaluate it on synthetic datasets via the *rank difference* between feature selection and prediction (Figure 4). Specifically, we first compute the F1 score of selected features ($F1_{select}$) and the balanced accuracy of predictions (ACC_{pred}). Then we compute the rank difference between $F1_{select}$ and ACC_{pred}. A *positive* difference demonstrates the model makes accurate predictions with relatively low-fidelity features. The complete numerical results are in Appendix I.2.

Figure 4 (Left) shows that ProtoGate achieves competitive trade-off between $F1_{select}$ and ACC_{pred} against the benchmark methods. In contrast, some existing methods, such as LSPIN, can achieve high accuracy in prediction, while

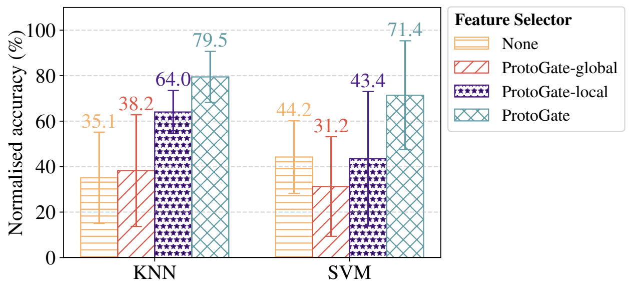


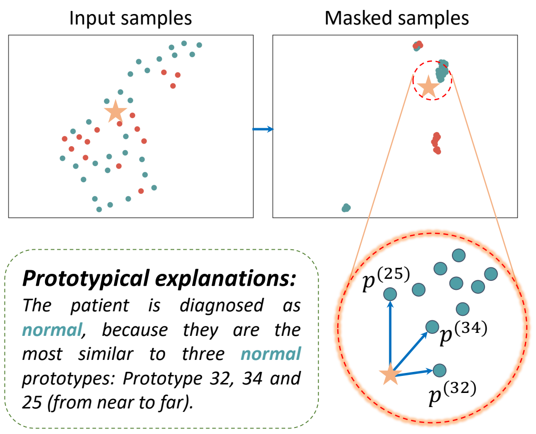
Figure 5. **Normalised balanced accuracy (%) of simple models with different feature selectors.** ProtoGate, with global-to-local selection, selects highly transferable features that generally improve the classification performance of KNN and SVM.

the feature selection is relatively inaccurate, indicating poor fidelity of selected features. Figure 4 (Right) further demonstrates that ProtoGate is the only in-model local feature selection method that consistently has non-positive rank differences between $\text{F1}_{\text{select}}$ and ACC_{pred}. Although REAL-X also has non-positive values across datasets, its post-hoc explainability of selected features cannot improve the prediction performance, leading to poor classification accuracy. In contrast, the in-model explainability helps ProtoGate to achieve high fidelity of selected features while guaranteeing high classification accuracy.

Furthermore, the results raise concerns over the reliability of high accuracy achieved by other benchmark local methods in Section 4.1. For instance, LSPIN achieves the best classification accuracy on Syn1, but the correctness of selected features is much worse, with a rank of six out of ten methods. Consequently, LSPIN's high accuracy may not correspond to selecting the real informative features, and this is also true for other local methods that are susceptible to co-adaption. In contrast, ProtoGate's well-aligned performance between feature selection and classification guarantees the fidelity of selected features.

Transferability of selected features. This criterion focuses on *"Can the selected features improve the performance of other simpler models?"*. We evaluate the balanced accuracy of downstream classifiers trained on the features selected by ProtoGate. Specifically, we first process the HDLSS datasets by applying $\boldsymbol{x} \odot \boldsymbol{s}_{\text{local}}$ – where the mask $\boldsymbol{s}_{\text{local}}$ is generated by ProtoGate, ProtoGate-global or ProtoGate-local – and then train a KNN and SVM on the masked datasets. For each simple model, we aggregate the balanced accuracy across datasets with the ADTM metric.

In Figure 5, the features selected by ProtoGate generally improve the performance of KNN and SVM across datasets. In contrast, the benefits conferred by ProtoGate-global and ProtoGate-local are limited, even leading to lower accuracy than vanilla SVM. The fine-grained results on the accuracy improvements per dataset (see Appendix I.3 for more details) reveal that features selected by ProtoGate



Prototypical explanations:
*The patient is diagnosed as **normal**, because they are the most similar to three **normal** prototypes: Prototype 32, 34 and 25 (from near to far).*

Figure 6. **Visualisation of ProtoGate's inference on the "colon" dataset.** The circles in "Masked samples" denote prototype base $\mathcal{B}$ learned by ProtoGate, containing **normal** and **cancer** prototypes $p^{(i)}$. The star is a new patient (i.e., a sample in the test set). ProtoGate provides an explainable diagnosis for the new patient.

consistently avoid performance degradation for simple models, while ProtoGate-global and ProtoGate-local can cause a considerable drop in accuracy. These findings also support that ProtoGate selects different features to its variants that rely solely on either global or local selection, showing the efficacy of global-to-local feature selection.

Explainability of predictions. This criterion focuses on *"Can the predictions be easily understood by practitioners?"*, and we qualitatively evaluate it by providing specific examples of predictions. Figure 6 presents UMAP (McInnes et al., 2018) on the "colon" dataset to show ProtoGate's inference process. In this split, ProtoGate classifies the new sample by explicitly pointing out the three ($K = 3$) nearest prototypes as evidence. ProtoGate's prototypical predictions are easy to interpret because they resemble human behaviour (Kolodner, 1992; Lu et al., 2021).

5. Conclusion

We introduce ProtoGate, a prototype-based neural model for feature selection in the high-dimensional and low-sample-size regime. ProtoGate proposes to select features in a global-to-local manner and perform non-parametric prototype-based prediction via hybrid sorting. The experimental results on real-world datasets demonstrate that ProtoGate generally improves the accuracy and feature selection sparsity of local methods on tabular datasets without sacrificing computation efficiency. The interpretability evaluation further validates that ProtoGate can effectively guarantee the fidelity and transferability of selected features by elegantly mitigating the co-adaptation problem. ProtoGate's robust performance and the ability to generate human-understandable explanations for its prototypical predictions has the potential to enhance practitioners' experience in decision-making processes beyond the biomedical domain.

Impact Statement

This paper presents a novel feature selection method that aims to advance the field of machine learning by addressing challenges in the high-dimensional and low-sample-size regime. Furthermore, ProtoGate offers an elegant solution to the prevalent co-adaptation problem in machine learning models, which helps to pave the path to more robust and applicable feature selection methods. These characteristics can be particularly useful in high-stakes and data-scarce domains like healthcare, such as preclinical drug evaluation in early-stage clinical trials (Bespalov et al., 2016; Morford et al., 2011). Moreover, ProtoGate can provide patient-wise insights with informative features and similar prototypes as explanations, facilitating automatic diagnosis systems (Fansi Tchango et al., 2022; Nazari et al., 2022).

ProtoGate's impact further extends to enabling broader machine learning applications in low-resource regions, such as low-income countries where medical resources are limited. Indeed, improving machine learning models' accuracy and interpretability in low-sample-size regimes can help elevate healthcare quality and foster medical equity (Alami et al., 2020; Ciecierski-Holmes et al., 2022; Mollura et al., 2020). ProtoGate can further facilitate research and enhance machine learning accessibility in various communities. Without foreseeing harmful applications, ProtoGate is designed to enhance the availability and fairness of AI across various societal and scientific domains.

Acknowledgements

AM acknowledges the support from the Cambridge ESRC Doctoral Training Partnership. NS acknowledges the support of the U.S. Army Medical Research and Development Command of the Department of Defense; through the FY22 Breast Cancer Research Program of the Congressionally Directed Medical Research Programs, Clinical Research Extension Award GRANT13769713. Opinions, interpretations, conclusions, and recommendations are those of the authors and are not necessarily endorsed by the Department of Defense.

References

Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., and Kim, B. Sanity checks for saliency maps. *Advances in neural information processing systems*, 31, 2018.

Akiba, T., Sano, S., Yanase, T., Ohta, T., and Koyama, M. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

Alami, H., Rivard, L., Lehoux, P., Hoffman, S. J., Cadeddu, S. B. M., Savoldelli, M., Samri, M. A., Ag Ahmed, M. A., Fleet, R., and Fortin, J.-P. Artificial intelligence in health care: laying the foundation for responsible, sustainable, and inclusive innovation in low-and middle-income countries. *Globalization and Health*, 16:1–6, 2020.

Alvarez Melis, D. and Jaakkola, T. Towards robust interpretability with self-explaining neural networks. *Advances in neural information processing systems*, 31, 2018.

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

Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., and Samek, W. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015.

Bajwa, G., DeBerardinis, R. J., Shao, B., Hall, B., Farrar, J. D., and Gill, M. A. Cutting edge: Critical role of glycolysis in human plasmacytoid dendritic cell antiviral responses. *The Journal of Immunology*, 196(5):2004–2009, 2016.

Ball, G. H. and Hall, D. J. Isodata, a novel method of data analysis and pattern classification. Technical report, Stanford research inst Menlo Park CA, 1965.

Baxevanis, A. D., Bader, G. D., and Wishart, D. S. *Bioinformatics*. John Wiley & Sons, 2020.

Bespalov, A., Steckler, T., Altevogt, B., Koustova, E., Skolnick, P., Deaver, D., Millan, M. J., Bastlund, J. F., Doller, D., Witkin, J., et al. Failed trials for central nervous system disorders do not necessarily invalidate preclinical models and drug targets. *Nature Reviews Drug Discovery*, 15(7):516–516, 2016.

Bhattacharjee, A., Richards, W. G., Staunton, J., Li, C., Monti, S., Vasa, P., Ladd, C., Beheshti, J., Bueno, R., Gillette, M., et al. Classification of human lung carcinomas by mrna expression profiling reveals distinct adenocarcinoma subclasses. *Proceedings of the National Academy of Sciences*, 98(24):13790–13795, 2001.

Bichindaritz, I. Case-based reasoning in the health sciences: Why it matters for the health sciences and for cbr. In *Advances in Case-Based Reasoning: 9th European Conference, ECCBR 2008, Trier, Germany, September 1-4, 2008. Proceedings 9*, pp. 1–17. Springer, 2008.

Bichindaritz, I. and Marling, C. Case-based reasoning in the health sciences: What's next? *Artificial intelligence in medicine*, 36(2):127–135, 2006.

Biehl, M., Hammer, B., and Villmann, T. Prototype-based models in machine learning. *Wiley Interdisciplinary Reviews: Cognitive Science*, 7(2):92–111, 2016.

Blondel, M., Teboul, O., Berthet, Q., and Djolonga, J. Fast differentiable sorting and ranking. In *International Conference on Machine Learning*, pp. 950–959. PMLR, 2020.

Bouckaert, R. R. and Frank, E. Evaluating the replicability of significance tests for comparing learning algorithms. In *Pacific-Asia conference on knowledge discovery and data mining*, pp. 3–12. Springer, 2004.

Breiman, L. Random forests. *Machine learning*, 45:5–32, 2001.

Chapelle, O., Scholkopf, B., and Zien, A. Semi-supervised learning. 2006. *Cambridge, Massachusettes: The MIT Press View Article*, 2, 2006.

Chen, J., Song, L., Wainwright, M., and Jordan, M. Learning to explain: An information-theoretic perspective on model interpretation. In *International Conference on Machine Learning*, pp. 883–892. PMLR, 2018.

Chen, T. and Guestrin, C. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794, 2016.

Ciecierski-Holmes, T., Singh, R., Axt, M., Brenner, S., and Barteit, S. Artificial intelligence for strengthening healthcare systems in low-and middle-income countries: a systematic scoping review. *npj Digital Medicine*, 5(1): 162, 2022.

Cortes, C. and Vapnik, V. Support-vector networks. *Machine learning*, 20:273–297, 1995.

Cox, D. R. The regression analysis of binary sequences. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 20(2):215–232, 1958.

Curtis, C., Shah, S. P., Chin, S.-F., Turashvili, G., Rueda, O. M., Dunning, M. J., Speed, D., Lynch, A. G., Samarajiwa, S., Yuan, Y., et al. The genomic and transcriptomic architecture of 2,000 breast tumours reveals novel subgroups. *Nature*, 486(7403):346–352, 2012.

Defazio, A., Bach, F., and Lacoste-Julien, S. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. *Advances in neural information processing systems*, 27, 2014.

Ding, C. and Peng, H. Minimum redundancy feature selection from microarray gene expression data. *Journal of bioinformatics and computational biology*, 3(02):185–205, 2005.

Falcon, W. and The PyTorch Lightning team. PyTorch Lightning, March 2019. URL `https://github.com/Lightning-AI/lightning`.

Fan, Y., Zhang, S., and Ma, S. Survival analysis with high-dimensional omics data using a threshold gradient descent regularization-based neural network approach. *Genes*, 13 (9):1674, 2022.

Fansi Tchango, A., Goel, R., Martel, J., Wen, Z., Marceau Caron, G., and Ghosn, J. Towards trustworthy automatic diagnosis systems by emulating doctors' reasoning with deep reinforcement learning. *Advances in Neural Information Processing Systems*, 35:24502–24515, 2022.

Fix, E. *Discriminatory analysis: nonparametric discrimination, consistency properties*, volume 1. USAF school of Aviation Medicine, 1985.

Goldberger, J., Hinton, G. E., Roweis, S., and Salakhutdinov, R. R. Neighbourhood components analysis. *Advances in neural information processing systems*, 17, 2004.

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

Grinsztajn, L., Oyallon, E., and Varoquaux, G. Why do tree-based models still outperform deep learning on typical tabular data? *Advances in neural information processing systems*, 35:507–520, 2022.

Gros, C., Lemay, A., and Cohen-Adad, J. Softseg: Advantages of soft versus binary training for image segmentation. *Medical image analysis*, 71:102038, 2021.

Grover, A., Wang, E., Zweig, A., and Ermon, S. Stochastic optimization of sorting networks via continuous relaxations. In *International Conference on Learning Representations*, 2018.

Hoare, C. A. R. Algorithm 64: quicksort. *Communications of the ACM*, 4(7):321, 1961.

Hooker, S., Erhan, D., Kindermans, P.-J., and Kim, B. A benchmark for interpretability methods in deep neural networks. *Advances in neural information processing systems*, 32, 2019.

Hsu, A. L., Tang, S.-L., and Halgamuge, S. K. An unsupervised hierarchical dynamic self-organizing approach to cancer class discovery and marker gene identification in microarray data. *Bioinformatics*, 19(16):2131–2140, 2003.

Jarquín, D., Crossa, J., Lacaze, X., Du Cheyron, P., Daucourt, J., Lorgeou, J., Piraux, F., Guerreiro, L., Pérez, P., Calus, M., et al. A reaction norm model for genomic selection using high-dimensional genomic and environmental data. *Theoretical and applied genetics*, 127:595–607, 2014.

Jethani, N., Sudarshan, M., Aphinyanaphongs, Y., and Ranganath, R. Have we learned to explain?: How interpretability methods can learn to encode predictions in their interpretations. In *International Conference on Artificial Intelligence and Statistics*, pp. 1459–1467. PMLR, 2021.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.-Y. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017.

Kolodner, J. L. An introduction to case-based reasoning. *Artificial intelligence review*, 6(1):3–34, 1992.

Lemhadri, I., Ruan, F., and Tibshirani, R. Lassonet: Neural networks with feature sparsity. In *International Conference on Artificial Intelligence and Statistics*, pp. 10–18. PMLR, 2021.

Lesk, A. *Introduction to bioinformatics*. Oxford university press, 2019.

Levin, R., Cherepanova, V., Schwarzschild, A., Bansal, A., Bruss, C. B., Goldstein, T., Wilson, A. G., and Goldblum, M. Transfer learning with deep tabular models. In *The Eleventh International Conference on Learning Representations*, 2022.

Li, O., Liu, H., Chen, C., and Rudin, C. Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Liberzon, A., Birger, C., Thorvaldsdóttir, H., Ghandi, M., Mesirov, J. P., and Tamayo, P. The molecular signatures database hallmark gene set collection. *Cell systems*, 1(6): 417–425, 2015.

Liu, D. C. and Nocedal, J. On the limited memory bfgs method for large scale optimization. *Mathematical Programming*, 45:503–528, 1989.

Louizos, C., Welling, M., and Kingma, D. P. Learning sparse neural networks through l_0 regularization. In *International Conference on Learning Representations*, 2018.

Lu, Y. Y., Timothy, C. Y., Bonora, G., and Noble, W. S. Ace: Explaining cluster from an adversarial perspective. In *International Conference on Machine Learning*, pp. 7156–7167. PMLR, 2021.

Lu, Z., Pu, H., Wang, F., Hu, Z., and Wang, L. The expressive power of neural networks: A view from the width. *Advances in neural information processing systems*, 30, 2017.

Lundberg, S. M. and Lee, S.-I. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

Lundberg, S. M., Erion, G. G., and Lee, S.-I. Consistent individualized feature attribution for tree ensembles. *methods*, 5(13):25, 2018.

Margeloiu, A., Simidjievski, N., Lio, P., and Jamnik, M. Weight predictor network with feature selection for small sample tabular biomedical data. *AAAI Conference on Artificial Intelligence*, 2023a.

Margeloiu, A., Simidjievski, N., Lio, P., and Jamnik, M. Gcondnet: A novel method for improving neural networks on small high-dimensional tabular data. In *NeurIPS 2023 Second Table Representation Learning Workshop*, 2023b.

McElfresh, D., Khandagale, S., Valverde, J., Ramakrishnan, G., Prasad, V., Goldblum, M., and White, C. When do neural nets outperform boosted trees on tabular data? In *Advances in Neural Information Processing Systems*, 2023.

McInnes, L., Healy, J., Saul, N., and Großberger, L. Umap: Uniform manifold approximation and projection. *Journal of Open Source Software*, 3(29), 2018.

Mollura, D. J., Culp, M. P., Pollack, E., Battino, G., Scheel, J. R., Mango, V. L., Elahi, A., Schweitzer, A., and Dako, F. Artificial intelligence in low-and middle-income countries: innovating global health radiology. *Radiology*, 297 (3):513–520, 2020.

Morford, L. L., Bowman, C. J., Blanset, D. L., Bøgh, I. B., Chellman, G. J., Halpern, W. G., Weinbauer, G. F., and Coogan, T. P. Preclinical safety evaluations supporting pediatric drug development with biopharmaceuticals: strategy, challenges, current practices. *Birth Defects Research Part B: Developmental and Reproductive Toxicology*, 92 (4):359–380, 2011.

Nadeau, C. and Bengio, Y. Inference for the generalization error. *Advances in neural information processing systems*, 12, 1999.

Nazari, M., Kluge, A., Apostolova, I., Klutmann, S., Kimiaei, S., Schroeder, M., and Buchert, R. Explainable ai to improve acceptance of convolutional neural networks for automatic classification of dopamine transporter spect in the diagnosis of clinically uncertain parkinsonian syndromes. *European journal of nuclear medicine and molecular imaging*, pp. 1–11, 2022.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

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

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

Remeseiro, B. and Bolon-Canedo, V. A review of feature selection methods in medical applications. *Computers in biology and medicine*, 112:103375, 2019.

Ribeiro, M. T., Singh, S., and Guestrin, C. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144, 2016.

Rolfe, J. T. Discrete variational autoencoders. In *International Conference on Learning Representations*, 2016.

Samek, W., Binder, A., Montavon, G., Lapuschkin, S., and Müller, K.-R. Evaluating the visualization of what a deep neural network has learned. *IEEE transactions on neural networks and learning systems*, 28(11):2660–2673, 2016.

Shrikumar, A., Greenside, P., and Kundaje, A. Learning important features through propagating activation differences. In *International conference on machine learning*, pp. 3145–3153. PMLR, 2017.

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

Simonyan, K., Vedaldi, A., and Zisserman, A. Deep inside convolutional networks: visualising image classification models and saliency maps. In *Proceedings of the International Conference on Learning Representations (ICLR)*. ICLR, 2014.

Singh, D., Febbo, P. G., Ross, K., Jackson, D. G., Manola, J., Ladd, C., Tamayo, P., Renshaw, A. A., D'Amico, A. V., Richie, J. P., et al. Gene expression correlates of clinical prostate cancer behavior. *Cancer cell*, 1(2):203–209, 2002.

Song, W., Shi, C., Xiao, Z., Duan, Z., Xu, Y., Zhang, M., and Tang, J. Autoint: Automatic feature interaction learning via self-attentive neural networks. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pp. 1161–1170, 2019.

Thompson, R., Dezfouli, A., et al. The contextual lasso: Sparse linear models via deep neural networks. *Advances in Neural Information Processing Systems*, 36, 2024.

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

Tomczak, K., Czerwińska, P., and Wiznerowicz, M. The cancer genome atlas (tcga): an immeasurable source of knowledge. *Contemporary oncology*, 19(1A):A68, 2015.

Tucker, G., Mnih, A., Maddison, C. J., Lawson, J., and Sohl-Dickstein, J. Rebar: Low-variance, unbiased gradient estimates for discrete latent variable models. *Advances in Neural Information Processing Systems*, 30, 2017.

Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Reinforcement learning*, pp. 5–32, 1992.

Yamada, M., Koh, T., Iwata, T., Shawe-Taylor, J., and Kaski, S. Localized lasso for high-dimensional regression. In *Artificial Intelligence and Statistics*, pp. 325–333. PMLR, 2017.

Yamada, Y., Lindenbaum, O., Negahban, S., and Kluger, Y. Feature selection using stochastic gates. In *International Conference on Machine Learning*, pp. 10648–10659. PMLR, 2020.

Yang, J., Lindenbaum, O., and Kluger, Y. Locally sparse neural networks for tabular biomedical data. In *International Conference on Machine Learning*, pp. 25123–25153. PMLR, 2022.

Yoon, J., Jordon, J., and van der Schaar, M. Invase: Instance-wise variable selection using neural networks. In *International Conference on Learning Representations*, 2018.

Yoshikawa, Y. and Iwata, T. Neural generators of sparse local linear models for achieving both accuracy and interpretability. *Information Fusion*, 81:116–128, 2022.

Appendix: ProtoGate: Prototype-based Neural Networks with Global-to-local Feature Selection for Tabular Biomedical Data

Table of Contents

A Summary of Related Work **15**

B Pseudocode for ProtoGate Training and Inference **16**

C Illustration of Global-to-local Feature Selection **17**

D Theoretical Analysis **18**

E Reproducibility **20**

E.1 Real-word Datasets 20

E.2 Synthetic Datasets 20

E.3 Data Preprocessing 21

E.4 Computing Resources 21

E.5 Training Details and hyperparameter Tuning 21

F Additional Results on Real-world Prediction Tasks **25**

F.1 Results on Feature Selection 25

F.1.1 Numerical Results for Feature Selection Sparsity 25

F.1.2 Visualisation of Selected Features 25

F.1.3 Composition of Selected Features 27

F.2 Results on Computation Efficiency 27

F.3 Results on non-HDLSS Classification Tasks 28

G Additional Ablation Results on Global-to-local Feature Selection **29**

G.1 Results on Features Selection Sparsity 29

G.2 Results on Degree of Local Selection 29

H Additional Ablation Results on Non-parametric Prototype-based Prediction **31**

H.1 Results on Different Predictors 31

H.2 Results on Different Number of Nearest Neighbours K 31

H.3 Results on Hybrid Sorting 31

I Additional Results on Interpretability Evaluation **33**

I.1 Alternative Metrics for Interpretability Evaluation 33

I.2 Results on Fidelity of Selected Features 33

I.3 Results on Transferability of Selected Features 34

J Future Work **35**

A. Summary of Related Work

As a supplement to Section 2, we further provide a detailed summary of the related work on local feature selection and highlight the differences between ProtoGate and the prior studies.

Joint in-model selection (Figure 1(a)). This line of work attempts to predict instance-wise masks for local feature selection via jointly learning a local selector and a predictor (Chen et al., 2018; Yoon et al., 2018; Arik & Pfister, 2021; Yang et al., 2022; Yoshikawa & Iwata, 2022). TabNet uses sequential attention to select instance-wise features for different samples (Arik & Pfister, 2021), but it can suffer from overfitting on HDLSS datasets with the complex transformer architecture (Margeloiu et al., 2023a). L2X uses mutual information with Concrete distribution, but it requires specifying the number of selected features (Chen et al., 2018). INVASE addresses such limitation by modelling each feature's mask/gate value with independent Bernoulli distributions (Yoon et al., 2018). However, both L2X and INVASE utilise computationally expensive gradient estimators like REINFORCE (Williams, 1992) or REBAR (Tucker et al., 2017) to generate sparse masks. Based on STG (Yamada et al., 2020) and Localized Lasso (Yamada et al., 2017), LSPIN (Yang et al., 2022) further extends the estimation of ℓ_0-regularisation to the exact formulation. To date, few contributions have been made to investigate the efficacy of globally important features in local feature selection. The recent Contextual Lasso (Thompson et al., 2024) attempts to introduce context information for local feature selection, but it requires domain knowledge to partition the features into contextual features and explanatory features, which is impractical for HDLSS regimes. Although LassoNet (Lemhadri et al., 2021) features penalisation on the residual layer parameters and constraints on the first layer's parameters, it can only select features globally and has an unstable training process, even failed convergence, on HDLSS datasets (Margeloiu et al., 2023a;b). In addition, the paradigm to jointly learn a feature selector and a predictor is susceptible to the co-adaptation problem, leading to a considerable loss in the fidelity of selected features (Jethani et al., 2021; Adebayo et al., 2018; Hooker et al., 2019). Moreover, these methods typically make predictions without a tailored inductive bias for biomedical data, further highlighting their limitations.

These challenges in this relatively underexplored field suggest the complexity and non-trivial nature of proposing seemingly straightforward ideas. Indeed, our work bridges these gaps by (i) exploring the interplay between global and local selection, (ii) tackling the co-adaptation problem via incorporating a trainable feature selector with a non-trainable predictor and (iii) encoding the clustering assumption as an inductive bias by performing prototype-based prediction.

Disjoint post-hoc selection (Figure 1(b)). Another promising line of work addresses the heterogeneity across samples by designing models to explain a trained predictor via local feature selection (Jethani et al., 2021; Ribeiro et al., 2016; Lundberg & Lee, 2017; Shrikumar et al., 2017; Simonyan et al., 2014; Lundberg et al., 2018; Bach et al., 2015). However, these methods can have limited applicability because they can only provide post-hoc analysis of the important features. Furthermore, the features they identify do not enhance prediction performance, given that the predictor has already been trained using all features. In contrast, ProtoGate directly utilises the selected features for predictions, thereby not only boosting the non-parametric predictor's accuracy but also preserving the in-model explainability of selected features. Moreover, ProtoGate surpasses post-hoc selection approaches such as REAL-X in computation efficiency, as it mitigates the need to enumerate various mask possibilities for an accurate approximation of the predictor's decision boundary.

Table 4. **Model design comparison between ProtoGate and prior local feature selection methods.** ProtoGate has novel design rationales and leverages different feature selectors and predictors to the benchmark methods.

Methods	Category	Key ideas	Feature Selection		Prediction		Co-adaptation avoidance
			Global Selector	Local Selector	Explainability	Clustering assumption	
TabNet	Joint in-model selection	Uses sequential attention to select important features	✗	✓	✗	✗	✗
L2X		Mutual Information maximization with Concrete distribution	✗	✓	✗	✗	✗
INVASE		Modelling instance-wise mask value with independent Bernoulli distribution	✗	✓	✗	✗	✗
LSPIN		Formalising the mask value with Normal distribution	✗	✓	✗	✗	✗
LLSPIN		Replace LSPIN's predictor with a linear prediction head	✗	✓	✓	✗	✗
REAL-X	Disjoint post-hoc selection	Decoupling the training objectives of the feature selector and predictor	✗	✓	✗	✗	✓
ProtoGate (Ours)	Disjoint in-model selection	Balances global and local selection and avoiding co-adaptation with non-parametric prototype-based prediction	✓	✓	✓	✓	✓

B. Pseudocode for ProtoGate Training and Inference

Algorithm 1 Training procedure of ProtoGate

```
Input: training samples $X \in \mathbb{R}^{N \times D}$, labels $Y \in \mathbb{R}^{N \times 1}$, global-to-local feature selector $S_{\mathbf{W}}$, prototype-based predictor $F_\theta$,
    sparsity hyperparameters ($\lambda_{\text{global}}$, $\lambda_{\text{local}}$), number of nearest neighbours $K$, total training epochs $E$, learning rate $\alpha$
Output: trained feature selector $S_{\mathbf{W}}$, prototype base $\mathcal{B}$
  $\mathbf{W} \leftarrow$ GaussianInitialisation() {Initialise the weights of feature selector}
  for $e \leftarrow 1$ to $E$ do
    $\mathcal{B} \leftarrow \{\}$ {Initialise the prototype base as an empty set}
    $\boldsymbol{s}_{\text{global}} \leftarrow \mathbf{W}^{[1]}$ {Compute the global mask from $\mathbf{W}^{[1]}$}
    for $i \leftarrow 1$ to $N$ do
      $\boldsymbol{s}_{\text{local}}^{(i)} \leftarrow S_{\mathbf{W}}(\boldsymbol{x}^{(i)} \odot \boldsymbol{s}_{global})$ {Predict the local mask for a specific training sample}
      $p^{(i)} \leftarrow (\boldsymbol{x}^{(i)} \odot \boldsymbol{s}_{\text{local}}^{(i)}, y^{(i)})$ {Construct a prototype with the masked training sample and its label}
      $\mathcal{B} \leftarrow \mathcal{B} \cup \{p^{(i)}\}$ {Retain the prototype in the base}
    end for
    for $j \leftarrow 1$ to $N$ do
      $\boldsymbol{x}_{\text{query}}^{(j)} \leftarrow$ RandomSampling$(X)$ {Randomly pick a training sample as the query sample}
      $\boldsymbol{x}_{\text{masked}}^{(j)} \leftarrow \boldsymbol{x}_{\text{query}}^{(j)} \odot S_{\mathbf{W}}(\boldsymbol{x}_{\text{query}}^{(j)} \odot \boldsymbol{s}_{\text{global}})$ {Perform local feature selection for the query sample}
      $\widehat{\mathbf{P}}_{\text{query}}^{(j)} \leftarrow$ HybridSort$(X_{\mathcal{B}}, \boldsymbol{x}_{\text{masked}}^{(j)})$ {Compute the permutation matrix based on the similarity to the query sample}
      $\hat{y}_{\text{query}}^{(j)} \leftarrow$ MajorityClass $\left((\widehat{\mathbf{P}}_{\text{query}}^{(j)} Y_{\mathcal{B}})[1:K]\right)$ {Predict with the majority class of the $K$ nearest prototypes}
    end for
    $\mathcal{L}_{\text{total}} \leftarrow \mathbb{E}_{(X,Y)}\left[\mathcal{L}_{\text{pred}} + \mathcal{R}_{\text{select}}\right]$ {Compute the training loss according to Equation (1) and Equation (2)}
    $\mathbf{W} \leftarrow \mathbf{W} - \alpha \nabla_{\mathbf{W}} \mathcal{L}_{\text{total}}$ {Update the weights of feature selector}
  end for
  return $S_{\mathbf{W}}, \mathcal{B}$
```

Algorithm 2 Inference procedure of ProtoGate

```
Input: test sample $\boldsymbol{x}_{\text{test}} \in \mathbb{R}^D$
Output: predicted label $\hat{y}_{\text{test}} \in \mathbb{R}$, prototypical explanations $C_{\text{explanation}}$
  $\boldsymbol{s}_{\text{global}} \leftarrow \mathbf{W}^{[1]}$ {Compute the global mask from $\mathbf{W}^{[1]}$}
  $\boldsymbol{s}_{\text{local}} \leftarrow S_{\mathbf{W}}(\boldsymbol{x}_{\text{test}} \odot \boldsymbol{s}_{\text{global}})$ {Predict the local mask for the test sample}
  $\boldsymbol{x}_{\text{masked}} \leftarrow \boldsymbol{x}_{\text{query}} \odot S_{\mathbf{W}}(\boldsymbol{x}_{\text{query}} \odot \boldsymbol{s}_{\text{global}})$ {Perform local feature selection on the test sample}
  $\widehat{\mathbf{P}}_{\text{test}} \leftarrow$ HybridSort$(X_{\mathcal{B}}, \boldsymbol{x}_{\text{masked}})$ {Compute the permutation matrix based on the similarity to the test sample}
  $\hat{y}_{\text{test}} \leftarrow$ MajorityClass $\left((\widehat{\mathbf{P}}_{\text{test}} Y_{\mathcal{B}})[1:K]\right)$ {Predict with the majority class of the $K$ nearest prototypes}
  $C_{\text{explanation}} \leftarrow \left((\widehat{\mathbf{P}}_{\text{test}} X_{\mathcal{B}})[1:K], (\widehat{\mathbf{P}}_{\text{test}} Y_{\mathcal{B}})[1:K]\right)$ {Use the $K$ nearest prototypes as prototypical explanations}
  return $\hat{y}_{\text{test}}, C_{\text{explanation}}$
```

C. Illustration of Global-to-local Feature Selection

In Figure 7, we illustrate the interplay between soft global selection and local selection with a real-world example from the "colon" dataset. Specifically, we randomly pick a data split and visualise the mask values of the first training sample (i.e., $\boldsymbol{s}_{\text{global}}$ and $\boldsymbol{s}_{\text{local}}^{(1)}$). We can see the four different feature selection behaviours: *(i) Both selected:* the 61st feature is selected by both global and local selection; *(ii) Locally dropped:* the 828th feature is globally selected, but it is locally dropped for this sample; *(iii) Locally recovered:* the 1355th feature is globally dropped, but it is locally recovered for this sample; *(iv) Both dropped:* the 1759th feature is dropped globally and locally. These results show ProtoGate's flexible feature selection process, thus allowing the model to adapt to various datasets via balancing global and local selection.

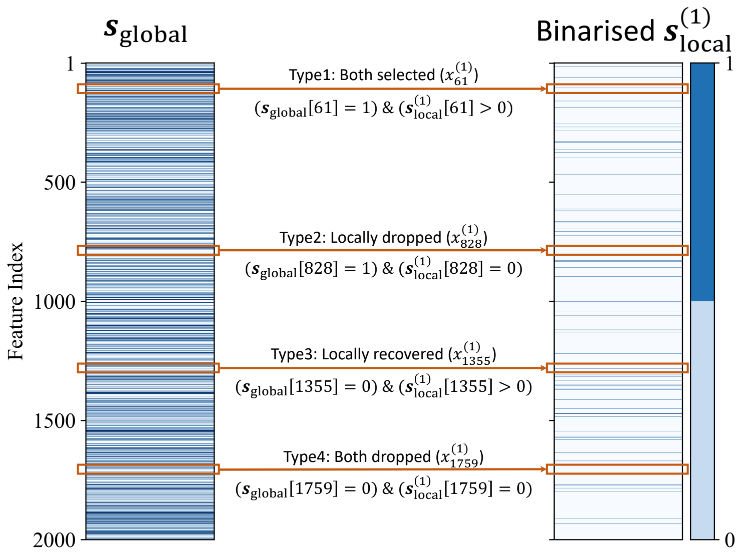


Figure 7. **Illustration of the global-to-local feature selection.** We visualise the mask values of the first training sample (i.e., $\boldsymbol{x}^{(1)}$) from the "colon" dataset. Note that we binarise the local mask via $\mathbb{1}(\boldsymbol{s}_{\text{local}}^{(1)} > 0)$. The heatmap presents four different feature selection behaviours, showing ProtoGate's capability to adaptatively balance global and local selection.

D. Theoretical Analysis

We now provide the proof sketch for readers to understand the key ideas of ProtoGate, which revolves around computing gradients for the approximation of non-differentiable operations. Therefore, $\mathcal{L}_{\text{total}}$ boils down to deriving the empirical estimations of ℓ_0 norm of local masks $\left\|\boldsymbol{s}_{\text{local}}^{(i)}\right\|_0$ and permutation matrix $\mathbf{P}$, which we show below respectively.

Estimation of $\left\|\boldsymbol{s}_{\text{local}}^{(i)}\right\|_0$. The derivation of differentiable ℓ_0-regularisation follows prior work (Louizos et al., 2018; Rolfe, 2016; Yamada et al., 2020; Yang et al., 2022). Recall the local mask is computed by $\boldsymbol{s}_{\text{local}}^{(i)} = \max(0, \min(1, \boldsymbol{\mu}^{(i)} + \boldsymbol{\epsilon}^{(i)}))$, where $\boldsymbol{\epsilon}^{(i)}$ is Gaussian noise sampled from $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$. Given that $\boldsymbol{\epsilon}^{(i)}$ is sampled from a Gaussian distribution with mean vector $\mathbf{0}$ and covariance matrix $\boldsymbol{\Sigma}$, and noting the independence and identical distribution (i.i.d.) of the noise across dimensions, $\boldsymbol{\Sigma}$ simplifies to $\sigma\mathbf{I}$, where $\mathbf{I} \in \mathbb{R}^{D\times D}$ is an identity matrix. The standard deviation $\sigma = 0.5$ is fixed during training, and it is removed during the inference time for deterministic mask values. Subsequently, the relaxed sparsity regularisation for the local mask can be re-formalised as an expectation over the probabilistic distribution induced by the Gaussian noise:

$$\begin{aligned}
\mathcal{R}_{\text{local}}(\boldsymbol{s}_{\text{local}}^{(i)}) &= \lambda_{\text{local}} \left\|\boldsymbol{s}_{\text{local}}^{(i)}\right\|_0 \propto \lambda_{\text{local}} \mathbb{E}\left[\left\|\boldsymbol{s}_{\text{local}}^{(i)}\right\|_0\right] \\
&\propto \lambda_{\text{local}} \mathbb{E}_{\boldsymbol{\epsilon}^{(i)}\sim\mathcal{N}(\mathbf{0},\sigma\mathbf{I})}\left[\left\|\max\left(0, \min\left(1, \boldsymbol{\mu}^{(i)} + \boldsymbol{\epsilon}^{(i)}\right)\right)\right\|_0\right] \\
&\propto \lambda_{\text{local}} \sum_{d=1}^{D} P\left(\mu_d^{(i)} + \epsilon_d^{(i)} > 0\right) \\
&\propto \lambda_{\text{local}} \sum_{d=1}^{D} P\left(\frac{\epsilon_d^{(i)}}{\sigma} > -\frac{\mu_d^{(i)}}{\sigma}\right) \\
&\propto \lambda_{\text{local}} \sum_{d=1}^{D} Q\left(-\frac{\mu_d^{(i)}}{\sigma}\right) \\
&\propto \lambda_{\text{local}} \sum_{d=1}^{D} \int_{\hat{\mu}_d^{(i)}}^{\infty} \exp(-\frac{t^2}{2})dt
\end{aligned} \tag{4}$$

where $\hat{\mu}_d^{(i)} = -\frac{\mu_d^{(i)}}{\sigma}$ is the standardised output of $S_{\mathbf{W}}$. Hence, the complete sparsity regularisation can be approximated via:

$$\begin{aligned}
\mathcal{R}_{\text{select}}(\boldsymbol{s}_{\text{global}}, \boldsymbol{s}_{\text{local}}^{(i)}) &:= \mathcal{R}_{\text{global}}(\boldsymbol{s}_{\text{global}}) + \mathcal{R}_{\text{local}}(\boldsymbol{s}_{\text{local}}^{(i)}) \\
&\propto \lambda_{\text{global}} \left\|\mathbf{W}^{[1]}\right\|_1 + \lambda_{\text{local}} \sum_{d=1}^{D} \int_{\hat{\mu}_d^{(i)}}^{\infty} \exp(-\frac{t^2}{2})dt.
\end{aligned} \tag{5}$$

Estimation of P. The essence of prototype-based prediction is to sort the prototypes based on their similarities to the query sample. Consider the matrix of the prototype features $X_{\mathcal{B}} := [\boldsymbol{x}^{(1)} \odot \boldsymbol{s}_{\text{local}}^{(1)}, \ldots, \boldsymbol{x}^{(N)} \odot \boldsymbol{s}_{\text{local}}^{(N)}]^\top \in \mathbb{R}^{N\times D}$ Sorting the prototypes is equivalent to performing row operations on $X_{\mathcal{B}}$ via left-multiplying by a permutation matrix $\mathbf{P} \in \mathbb{R}^{N\times N}$.

Definition D.1. Let $\boldsymbol{v} \in \mathbb{R}^N$ represent the vector of similarity, where each element $\boldsymbol{v}[n]$ signifies the reciprocal of the distance between the query sample and the n-th prototype (i.e., $\frac{1}{\text{dist}(x_{\text{query}}, \boldsymbol{x}^{(n)}\odot\boldsymbol{s}_{\text{local}}^{(n)})}$). Let $A \in \mathbb{R}^{N\times N}$ denote the matrix of absolute pairwise differences of the distances such that $A[n,m] = |v_n - v_m|$, and $\mathbf{1} \in \mathbb{R}^{N\times 1}$ denote a column vector of all ones. The permutation matrix can be defined as follows:

$$\mathbf{P}[n,m] = \begin{cases} 1, & \text{if } m = \arg\max\left[(N+1-2n)\boldsymbol{v} - A\mathbf{1}\right] \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

To circumvent the non-differentiability of $\arg\max$, we use its continuous approximation $\text{soft}\max$ to compute the relaxed permutation matrix:

$$\widehat{\mathbf{P}}[n,:](\tau) = \text{soft}\max\left[\frac{((N+1-2n)\boldsymbol{v} - A\mathbf{1})}{\tau}\right] \tag{7}$$

where $\tau > 0$ is a temperature parameter.

Lemma D.2. *For the relaxed permutation matrix $\widehat{\mathbf{P}}$, assuming the entries of $\boldsymbol{v}$ are independently drawn from a distribution continuously relative to the Lebesgue measure on $\mathbb{R}$, then the convergence holds almost surely:*

$$\lim_{\tau \to 0^+} \widehat{\mathbf{P}}[n,:](\tau) = \mathbf{P}[n,:], \quad \forall n \sim \mathcal{U}\{1, N\} \tag{8}$$

where $\mathcal{U}$ denotes a discrete uniform distribution. This convergence is substantiated by "Theorem 4" in Grover et al. (2018).

Theorem D.3. *In the context outlined above, the prediction loss of ProtoGate (Equation (2)) estimates the count of prototypes whose labels are different to that of the query sample.*

Proof. Invoking Lemma D.2, the prediction loss limit is computed as follows:

$$\begin{aligned} \lim_{\tau \to 0^+} \mathcal{L}_{\text{pred}} &= K - \lim_{\tau \to 0^+} \sum_{n=1}^{K} \mathbb{E}_{\widehat{\mathbf{P}}[n,:](\tau)} \left[\mathbb{1}(y^{(m)} = y_{\text{query}}) \right] \\ &= K - \sum_{n=1}^{K} \sum_{m=1}^{N} \mathbf{P}[n,m] \mathbb{1}(y^{(m)} = y_{\text{query}}) \end{aligned} \tag{9}$$

Since $\mathbf{P}$ is a binary, row-stochastic matrix, it holds that

$$\sum_{m=1}^{N} \mathbf{P}[n,m] = \begin{cases} 1, & \text{if } (\mathbf{P} Y_{\mathcal{B}})[n] = y_{\text{query}} \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

Here, $(\mathbf{P} Y_{\mathcal{B}})[n]$ represents the label of the n-th nearest prototype. Equation (10) indicates whether the n-th nearest prototype shares the same label as the query sample. Summing over the top K rows of $\mathbf{P}$ then calculates the count of prototypes that belong to the same class as the query sample among the top K nearest prototypes.

Therefore, minimising Equation (2) is equivalent to reducing the number of nearest prototypes with labels differing from the query sample, thereby potentially enhancing classification accuracy. □

E. Reproducibility

E.1. Real-word Datasets

HDLSS datasets. All seven HDLSS datasets are publicly available, and the details are listed in Table 5. Four of them are available online (`https://jundongl.github.io/scikit-feature/datasets`): **lung** (Bhattacharjee et al., 2001), **Prostate-GE** (referred to as "prostate") (Singh et al., 2002), **TOX-171** (referred to as "toxicity") (Bajwa et al., 2016) and **colon** (Ding & Peng, 2005). The methods to build the other three datasets are detailed below.

In accordance with the methodology presented in (Margeloiu et al., 2023a), we derived two datasets from the **METABRIC** dataset (Curtis et al., 2012). We combined the molecular data with the clinical label "DR" to create the **"meta-dr"** dataset, and we combined the molecular data with the clinical label "Pam50Subtype" to create the **"meta-pam"** dataset. Because the label "Pam50Subtype" was very imbalanced, we transformed the task into a binary task of basal vs non-basal by combining the classes "LumA", "LumB", "Her2", "Normal" into one class and using the remaining class "Basal" as the second class. For both "meta-dr" and "meta-pam", we selected the Hallmark gene set (Liberzon et al., 2015) associated with breast cancer, and the new datasets contain 4160 expressions (features) for each patient. We randomly sampled **200** patients while maintaining stratification to create the final datasets, as our focus is on the HDLSS regime.

Following the procedure by Margeloiu et al. (2023a), we also derived **"tcga-2y"** dataset from the **TCGA** dataset (Tomczak et al., 2015). We combined the molecular data and the label "X2yr.RF.Surv" to create the **"tcga-2ysurvival"** dataset. Similar to the previous datasets, we selected the Hallmark gene set (Liberzon et al., 2015) associated with breast cancer, resulting in 4381 expressions (features). We randomly sampled **200** patients while maintaining stratification to create the final datasets, as our primary focus is on the HDLSS regime.

Table 5. Details of seven HDLSS real-world tabular datasets.

Dataset	# Samples	# Features	# Classes	# Samples per class
colon	62	2,000	2	[40, 22]
lung	197	3,312	4	[139, 21, 20, 17]
meta-dr	200	4,160	2	[139, 61]
meta-pam	200	4,160	2	[167, 33]
prostate	102	5,966	2	[52, 50]
tcga-2y	200	4,381	2	[122, 78]
toxicity	171	5,748	4	[45, 45, 42, 39]

non-HDLSS datasets. To show the generalisability of ProtoGate, we select four representative and challenging non-HDLSS datasets from the open-source TabZilla benchmark (McElfresh et al., 2023). Firstly, their dimensionalities are smaller than the number of samples (i.e., $N > D$). Secondly, they are from various domains beyond the biomedical scenarios. For instance, "cnae-9" is relevant to business descriptions. Thirdly, they have many more classes than the HDLSS datasets we mentioned. For instance, "100-plant-texture" has up to 100 classes.

Table 6. Details of four non-HDLSS real-world tabular datasets.

Dataset	# Samples	# Features	# Classes	# Samples per class
100-plants-texture	1,599	65	100	16 per class except 15 of "Class 100"
cnae-9	1,080	857	9	120 per class
mfeat-fourier	2,000	77	10	200 per class
vehicle	846	19	4	[218, 217, 212, 199]

E.2. Synthetic Datasets

The synthetic datasets are adapted from the nonlinear datasets proposed by Yoon et al. (2018). Specifically, we generate three synthetic datasets: Syn1 (also referred to as "Syn1$_{(+)}$"), Syn2 (also referred to as "Syn2$_{(+)}$"), and Syn3 (also referred to as "Syn3$_{(-)}$"), which are designed for the classification task. Each sample is characterised by 100 features, where the feature values are independently sampled from a Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$, with $\mathbf{I}$ representing a 100×100 identity matrix. The ground truth label (target) y for each sample is computed by:

$$y = \mathbb{1}(\frac{1}{1 + \text{logit}(\boldsymbol{x})} > 0.5) \tag{11}$$

where $\mathbb{1}(\cdot)$ is the indicator function. For each samples, the logit$(\boldsymbol{x})$ is computed with a small proportion of its features:

$$\textbf{Syn1}_{(+)}: \ \text{logit} = \begin{cases} \exp(x_1 x_2 - x_3) & \text{if } x_{11} < 0 \\ \exp(x_3^2 + x_4^2 + x_5^2 + x_6^2 - 4) & \text{otherwise} \end{cases} \tag{12}$$

$$\textbf{Syn2}_{(+)}: \ \text{logit} = \begin{cases} \exp(x_3^2 + x_4^2 + x_5^2 + x_6^2 + x_7^2 - 4) & \text{if } x_{11} < 0 \\ \exp(-10\sin(0.2x_7) + |x_8| + x_9^2 + \exp(-x_{10}) - 2.4) & \text{otherwise} \end{cases} \tag{13}$$

$$\textbf{Syn3}_{(-)}: \ \text{logit} = \begin{cases} \exp(x_1 x_2 + |x_9|) & \text{if } x_{11} < 0 \\ \exp(-10\sin(0.2x_7) + |x_8| + x_9^2 + \exp(-x_{10}) - 2.4) & \text{otherwise} \end{cases} \tag{14}$$

Within each dataset, the two classes have a minimum of two informative features in common. For example, in Syn1$_{(+)}$, both class one and class two share (x_3, x_{11}) as the informative features. To introduce class imbalance, we intentionally generate 150 samples for class one and 50 samples for class two.

Note that we purposely design Syn3$_{(-)}$ to examine the clustering assumption in ProtoGate by adding even function $|x_9|$ to Class One. The absolute value function is an even function. Two samples with opposite values of the same feature are likely to have equal logit values, and then they belong to the same class. However, the opposite values mean a long distance between them, and they should not belong to the same class according to the clustering assumption. Therefore, prototype-based models are expected to perform poorly in this regime. We implement it by adding absolute value function $|x_9|$ in the first class of Syn3$_{(-)}$ to observe the performance degradation in ProtoGate. Because of the evenness of absolute value, two samples with opposite values of x_9 are likely to be of Class One, which is against the clustering assumption. Although Syn1$_{(+)}$ and Syn2$_{(+)}$ also contain even functions like square and absolute value, they also have many other informative features that do not utilise the even functions to compute logit value. Therefore, the side effect of even functions is diluted in Syn1$_{(+)}$ and Syn2$_{(+)}$.

Differences to prior work. Compared to previous studies (Yang et al., 2022; Yoon et al., 2018), we focus on more realistic and challenging synthetic datasets by considering four key aspects. Firstly, we only generate 200 samples for each dataset, which is only 10% of the samples in prior work (Yang et al., 2022). Secondly, each sample has 100 features, which is ten times more than that in prior work (Yang et al., 2022). Thirdly, our synthetic datasets are imbalanced, while those in prior work have balanced class distribution (Yang et al., 2022). Lastly, we incorporate a greater number of overlapping informative features between the two classes, while those in prior work may have no overlapping features (Yang et al., 2022).

E.3. Data Preprocessing

Following the methodology presented by Margeloiu et al. (2023a), we perform Z-score normalisation on each dataset before training the models. This normalisation process involves two steps. First, we compute the mean and standard deviation of each feature in the training data. Using these statistics, we transform the training samples to have a mean of zero and a variance of one for each feature. Subsequently, we apply the same transformation to the validation and test data before conducting evaluations.

E.4. Computing Resources

We trained over 15,000 models (including over 3,000 of ProtoGate) for evaluations. All the experiments were conducted on a machine equipped with an NVIDIA A100 GPU with 40GB memory and an Intel(R) Xeon(R) CPU (at 2.20GHz) with six cores. The operating system used was Ubuntu 20.04.5 LTS.

E.5. Training Details and hyperparameter Tuning

Software implementation. We implemented ProtoGate with Pytorch Lightning (Falcon & The PyTorch Lightning team, 2019): the global-to-local feature selector is implemented from scratch, and the relaxed sorting predictor is adapted from the official implementation of NeuralSort (https://github.com/ermongroup/neuralsort). Note that we further optimised the speed of the official implementation with matrix operators in PyTorch (Paszke et al., 2019). We re-implemented LSPIN/LLSPIN because the official implementation (https://github.com/jcyang34/lspin) used a different evaluation setup from ours: we report the mean $\pm$ std number of selected features, while they report the median number of selected features. We implemented XGBoost (https://xgboost.readthedocs.io/en/stable/), CatBoost (https://catboost.ai/) and LightGBM (https://github.com/microsoft/LightGBM) using their open-source implementations. With scikit-learn (Pedregosa et al., 2011), we implemented Random Forest (https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier), KNN (https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier) and Lasso (https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Lasso). For other benchmark methods, we used their open-source implementations: STG (https://github.com/runopti/stg), TabNet (https://github.com/dreamquark-ai/tabnet), L2X (https://github.com/Jianbo-Lab/L2X), INVASE (https://github.com/vanderschaarlab/mlforhealthlabpub/tree/main/alg/invase) and REAL-X (https://github.com/rajesh-lab/realx).

We implemented a uniform pipeline using PyTorch Lightning to ensure consistency and reproducibility. We further fixed the random seeds for data loading and evaluation throughout the training and evaluation process. This ensured that ProtoGate and all benchmark models were trained and evaluated on the same set of samples. The experimental environment settings, including library dependencies, are specified in the associated code for reference and replication purposes.

Note that all the libraries utilised in this study adhere to open-source licenses. Specifically, the scikit-learn and the INVASE implementation follow the BSD-3-Clause license, Pytorch Lightning follows the Apache-2.0 license, and the others follow the MIT license.

Training procedures. In this section, we outline the key training settings for ProtoGate and all benchmark methods. The complete experimental settings can be found in the accompanying code. We made diligent efforts to ensure a fair comparison among the benchmark methods whenever possible. For example, we employed the same predictor architecture in LSPIN, MLP and STG, as these models share similar design principles.

- **ProtoGate** has a three-layer feature selector. The number of neurons in the hidden layer is 200 for real-world datasets and 100 for synthetic datasets. And the activation function is $tanh$ in all layers for a fair comparison against prior work (Yang et al., 2022). The model is trained for 10,000 iterations using early stopping with patience 500 on the validation loss. We used the suggested temperature parameter $\tau = 16.0$ in NeuralSort (Grover et al., 2018). We train the models with a batch size of 64 and utilise an SGD optimiser with a weight decay of $1e-4$.

- **TabNet** has a width of eight for the decision prediction layer and the attention embedding for each mask and 1.5 for the coefficient for feature reusage in the masks. The model is trained with Adam optimiser with momentum of 0.3 and gradient clipping at 2.

- **L2X, INVASE and REAL-X** have the default architecture as published (Chen et al., 2018; Yoon et al., 2018; Jethani et al., 2021). The feature selector network has two hidden layers of $[100, 100]$, and the predictor network has two hidden layers of $[200, 200]$. They all use the $relu$ activation after layers. For convergence and computation efficiency, L2X is trained for 7,000 iterations, INVASE is trained for 5,000 epochs and REAL-X is trained for 1,000 iterations.

- **STG, LSPIN and LLSPIN** have a feature selector with the same architecture as that in ProtoGate. For LSPIN/STG, the predictor is a feed-forward neural network with hidden layers of $[100, 100, 10]$ with $tanh$ activation function. And we used the same architecture of predictor for **MLP**. For LLSPIN, the architecture of the predictor is the same, but the activation functions are removed. In other words, LLSPIN has multiple linear layers with no activations. This is proposed in LLSPIN's original paper (Yang et al., 2022) and the official open-source implementation (https://github.com/jcyang34/lspin/blob/dev/Demo0.ipynb?short_path=69bb17b#L214). That being said, we weren't able to find discussion or reasons behind the implementation choices of the LLSPIN's predictor with multiple linear layers. Nevertheless, we empirically tested and found that LLSPIN with multiple linear layers generally outperforms the variant with only one linear layer in the predictor (see Table 7). The standard deviation σ for injected noise is 0.5. The model is trained for 7,000 iterations using early stopping with patience 500 on the validation loss.

- **Ridge** is trained for 10,000 iterations to minimise the multinomial loss with Limited-memory BFGS solver (Liu & Nocedal, 1989), and the tolerance for early stopping is set as $1e-4$.
- **SVM** is trained for 10,000 iterations with the RBF kernel, and the tolerance for early stopping is set as $1e-3$.
- **KNN** measured the distance across samples with Euclidean distance and used uniform weights to compute the majority class in the neighbourhood.
- **Lasso** is trained for 10,000 iterations to minimise the weighted loss with SAGA solver (Defazio et al., 2014), and the tolerance for early stopping is set as $1e-4$.
- **Random Forest** has 500 estimators, feature bagging with the square root of the number of features, and used balanced weights from class distribution.
- **XGBoost** has 100 estimators. It is trained with a learning rate of 1.0 to minimise the cross-entropy loss. And the ℓ_2-regularization term on weights is set as $1e-5$.
- **CatBoost** has a maximum depth of 6, and it is trained with a learning rate of 0.03 to minimise the cross-entropy loss. The ℓ_2-regularization term of the cost function is 3.0.
- **LightGBM** has 200 estimators, feature bagging with 30% of the features, a minimum of two instances in a leaf. It is trained for 10,000 iterations to minimise the weighted cross-entropy loss using early stopping with patience 100 on validation loss.

Table 7. **Classification accuracy (%) of two variants of LLSPIN.** We **bold** the highest accuracy for each dataset. LLSPIN with multiple linear layers consistently achieves higher accuracy across all datasets.

Methods	colon	lung	meta-dr	meta-pam	prostate	tcga-2y	toxicity
LLSPIN (single linear layer)	73.75	62.50	55.95	88.31	85.45	55.21	74.65
LLSPIN (multiple linear layers)	**79.35**	**70.10**	**56.77**	**95.50**	**88.71**	**57.88**	**81.67**

Hyperparameter tuning. To ensure optimal performance, we initially identified a suitable range of hyperparameters for each model to facilitate convergence. Subsequently, we conducted a grid search within this predefined range to determine the optimal hyperparameter settings. The selection of models was based on their balanced accuracy on the validation sets averaged over 25 runs. It is worth noting that tuning hyperparameters in LSPIN can be challenging, particularly for real-world datasets. Therefore, we followed the recommendations in the original paper (Yang et al., 2022) and employed Optuna (Akiba et al., 2019) to fine-tune the hyperparameters for LSPIN.

Note that the LSPIN/LLSPIN performance on "toxicity" and "colon" datasets in Table 2 is different from its original paper. The mismatched performance stems from the experimental settings. Specifically: the LSPIN paper (Yang et al., 2022) mentions in paragraph 3 of its Appendix B.5.2 that they did **not** perform cross-validation on the "colon" and "toxicity" datasets. Namely, they did **not** use validation sets for model selection. In contrast, we have more realistic evaluation settings. Specifically, we **do** use validation sets for all datasets, leading to smaller train sets. Therefore, the classification accuracy of LSPIN/LLSPIN can be lower than those reported in the original paper.

Table 8 lists the searching ranges of hyperparameters in ProtoGate, and Table 9 lists the searching ranges of hyperparameters in network-based feature selection benchmark methods.

Table 8. Searching ranges of hyperparameters in ProtoGate.

Datasets	Global Sparsity λ_{global}	Local Sparsity λ_{local}	K	Learning Rate α
Real-word	$\{1e-4, 2e-4, 3e-4, 4e-4, 6e-4\}$	$\{1e-3\}$	$\{1,2,3,4,5\}$	$\{5e-2, 7.5e-2, 1e-1\}$
Synthetic	$\{1e-2, 1.5e-2, 2e-2\}$	$\{0, 1e-4, 3e-4\}$	$\{3\}$	$\{1e-1\}$

In line with prior studies (Margeloiu et al., 2023a; Yoon et al., 2018; Yang et al., 2022), we performed hyperparameter searching for other methods within the same ranges for real-world and synthetic datasets. For **Ridge**, we performed a

Table 9. **Searching ranges of hyperparameters in network-based feature selection benchmark methods.** Note that the range for LSPIN/LLSPIN on real-world datasets is an interval instead of a set because we used Optuna to search for the optimal settings.

Datasets	Methods	Sparsity regularisation strength (λ)	Learning Rate
Real-world	STG	$\{35, 40, 45, 50, 55\}$	$\{3e-3\}$
	TabNet	$\{1e-4, 1e-3, 1e-2, 1e-1\}$	$\{1e-2, 2e-2, 3e-2\}$
	L2X	$\{1, 5, 10\}$	$\{1e-4\}$
	INVASE	$\{1, 1.5, 2\}$	$\{1e-4\}$
	REAL-X	$\{1, 5, 10, 30, 50\}$	$\{1e-4\}$
	LSPIN/LLSPIN	$[5e-4, 1.5e-3]$	$[5e-2, 1e-1]$
Synthetic	STG	$\{1, 3, 5\}$	$\{1e-1\}$
	TabNet	$\{1e-2, 1e-1, 5e-1\}$	$\{1e-2\}$
	L2X	$\{1, 5, 10\}$	$\{1e-4\}$
	INVASE	$\{1, 1.5, 2\}$	$\{1e-4\}$
	REAL-X	$\{1, 5, 10, 30, 50\}$	$\{1e-4\}$
	LSPIN/LLSPIN	$\{1e-2, 5e-2, 1e-1\}$	$\{1e-1\}$

grid search of the regularisation strength in $\{1, 1e1, 1e2, 1e3\}$. For **SVM**, we performed a grid search of the regularisation strength in $\{1e-3, 1e-2, 1e-1, 1, 1e1\}$. For **KNN**, we performed a grid search of the number of nearest neighbours in $\{1, 3, 5\}$. For **MLP**, we used Optuna to find the optimal learning rate within $[1e-3, 1e-1]$. For **Lasso**, we performed a grid search of the regularisation strength in $\{1, 1e1, 1e2, 1e3\}$. For **Random Forest**, we performed a grid search for the maximum depth in $\{3, 5, 7\}$ and the minimum number of instances in a leaf in $\{2, 3\}$. For **LightGBM**, we performed a grid search for the learning rate in $\{1e-2, 1e-1\}$ and maximum depth in $\{1, 2\}$. Please refer to the associated code for the full details of the hyperparameter settings and their corresponding ranges.

F. Additional Results on Real-world Prediction Tasks

F.1. Results on Feature Selection

F.1.1. Numerical Results for Feature Selection Sparsity

As a supplement to Figure 3 (Left), we provide detailed numerical results of the feature selection sparsity here. In Table 10, we show the number of selected features per sample. For global methods, the number of selected features is the same across samples, and thus the standard deviation is zero. In Figure 8, we further visualise the proportion of selected features among all features. ProtoGate generally selects fewer features than benchmark feature selection methods, indicating improved interpretability in the selected features.

Table 10. **Feature selection sparsity on real-world datasets.** We report the mean ± std of the number of selected features on test samples, averaged over 25 runs. ProtoGate generally selects fewer features than other feature selection methods.

Methods	colon	lung	meta-dr	meta-pam	prostate	tcga-2y	toxicity
Lasso	$371.40_{\pm 0.00}$	$1618.08_{\pm 0.00}$	$4159.92_{\pm 0.00}$	$4159.40_{\pm 0.00}$	$5434.68_{\pm 0.00}$	$4214.56_{\pm 0.00}$	$2951.28_{\pm 0.00}$
RF	$1629.72_{\pm 0.00}$	$504.76_{\pm 0.00}$	$577.60_{\pm 0.00}$	$1439.20_{\pm 0.00}$	$510.72_{\pm 0.00}$	$887.12_{\pm 0.00}$	$1507.44_{\pm 0.00}$
LightGBM	$70.12_{\pm 0.00}$	$82.44_{\pm 0.00}$	$29.64_{\pm 0.00}$	$336.25_{\pm 0.00}$	$117.58_{\pm 0.00}$	$31.88_{\pm 0.00}$	$1150.48_{\pm 0.00}$
STG	$2000.00_{\pm 0.00}$	$3312.00_{\pm 0.00}$	$4157.96_{\pm 0.00}$	$2992.00_{\pm 0.00}$	$5966.00_{\pm 0.00}$	$4381.00_{\pm 0.00}$	$5748.00_{\pm 0.00}$
L2X	$5.00_{\pm 0.00}$	$1.00_{\pm 0.00}$	$5.00_{\pm 0.00}$	$10.00_{\pm 0.00}$	$10.00_{\pm 0.00}$	$5.00_{\pm 0.00}$	$5.00_{\pm 0.00}$
LSPIN	$1044.32_{\pm 293.67}$	$564.83_{\pm 1236.21}$	$1138.51_{\pm 1545.96}$	$1073.04_{\pm 1661.89}$	$2120.00_{\pm 1968.86}$	$1418.35_{\pm 1936.41}$	$1979.29_{\pm 2387.03}$
LLSPIN	$1311.86_{\pm 209.80}$	$673.27_{\pm 1212.20}$	$3026.77_{\pm 642.02}$	$1180.08_{\pm 1769.59}$	$2151.05_{\pm 1954.80}$	$3486.77_{\pm 696.29}$	$1999.12_{\pm 2398.65}$
ProtoGate	$110.70_{\pm 58.62}$	$177.24_{\pm 173.13}$	$337.21_{\pm 738.81}$	$469.47_{\pm 46.90}$	$91.29_{\pm 7.20}$	$348.79_{\pm 869.23}$	$76.39_{\pm 17.42}$

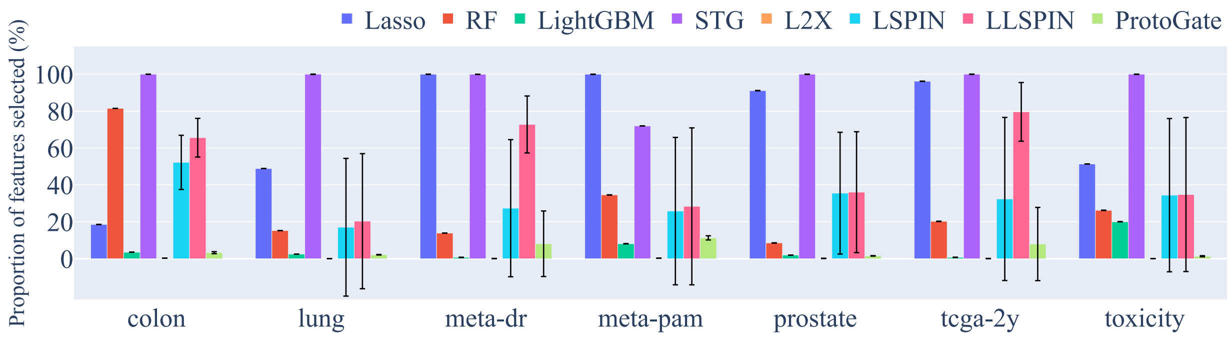
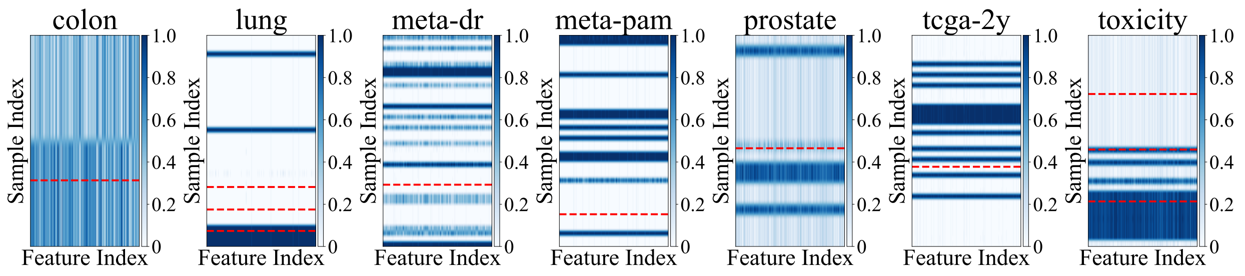


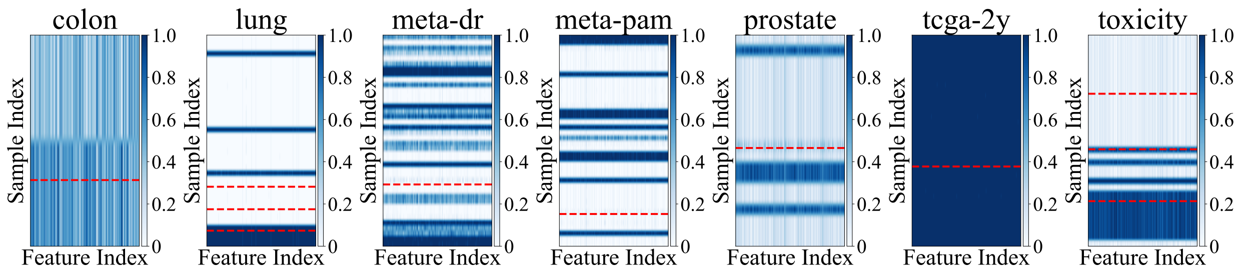
Figure 8. **Visualisation of the feature selection sparsity on real-world datasets.** We report the mean ± std of the proportion of selected features on test samples, averaged over 25 runs. ProtoGate generally selects fewer features than other local feature selection methods.

F.1.2. Visualisation of Selected Features

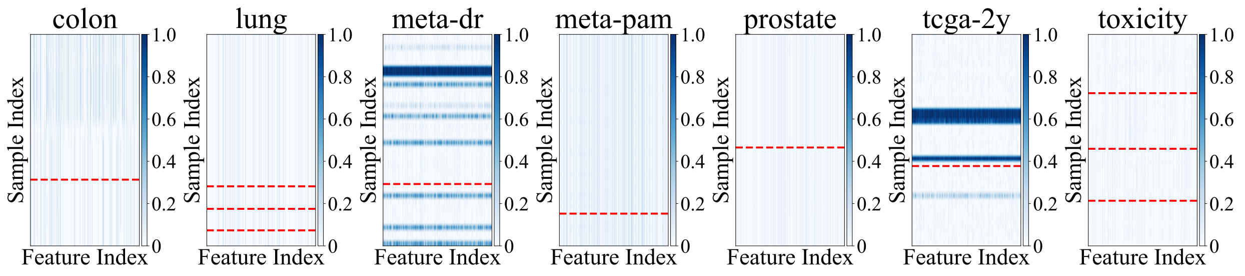
Figure 9 qualitatively shows that ProtoGate generally has smaller mask values than LSPIN and LLSPIN, denoting ProtoGate selects fewer features. Furthermore, we can see different feature selection results (i.e., s_{local}) in ProtoGate's heatmaps (e.g., the "meta-dr" dataset), showing that ProtoGate can select features locally. Additionally, ProtoGate can behave more globally on other datasets (e.g., the "lung" dataset). The results suggest that ProtoGate can adaptively balance global and local feature selection across different datasets, showing the effect of global-to-local feature selection.



(a) Heatmaps of mask values from LSPIN



(b) Heatmaps of mask values from LLSPIN



(c) Heatmaps of mask values from ProtoGate

Figure 9. **Visualisaton of the mask values on real-world datasets.** We plot the heatmaps of predicted mask values $\boldsymbol{s}_{\text{local}}^{(i)}$ of test samples, where the x-axis refers to the indices of features, and the y-axis refers to the indices of samples (this is different to Figure 7). Note that different datasets can have different numbers of samples and features, and the number of features should be more than the number of samples in the HDLSS regime. For visualisation purposes, we align them by adjusting the aspect ratio of heatmaps. The samples are sorted according to their ground truth labels, and the red dash lines separate samples of different classes.

F.1.3. Composition of Selected Features

We further analyse the composition of the final feature selection results (i.e., the proportions of "both selected" and "locally recovered" features in $\mathbf{s}_{\text{local}}^{(i)}$). Table 11 shows that the majority proportion of $\mathbf{s}_{\text{local}}^{(i)}$ is generally included in $\mathbf{s}_{\text{global}}$ on most datasets (i.e., the proportion of "both selected" features is generally greater than "locally recovered" features). For instance, on the "tcga-2y" dataset, 96.61% of $\mathbf{s}_{\text{local}}^{(i)}$ is included in $\mathbf{s}_{\text{global}}$. Table 11 also shows that ProtoGate can adaptively adjust its selection behaviours for different datasets. On some datasets, such as the "toxicity" dataset, ProtoGate locally recovers more features than reusing the ones from global selection. The above results show that ProtoGate always includes some globally selected features in the subsequent local selection results. Therefore, we introduce ProtoGate by "refining the global mask into the local mask". In other words, the global mask $\mathbf{s}_{\text{global}}$ empirically represents a lower-dimensional feature set to effectively determine the local mask $\mathbf{s}_{\text{local}}^{(i)}$.

Table 11. **Proportions (%) of "both selected" and "locally recovered" features in ProtoGate's local selection results (i.e., $\mathbf{s}_{\text{local}}^{(i)}$).** ProtoGate adaptively adjust its selection behaviours to be more global or more local for different datasets.

Selection behaviours	**colon**	**lung**	**meta-dr**	**meta-pam**	**prostate**	**tcga-2y**	**toxicity**
Both selected	56.99	79.54	33.33	69.64	84.56	96.61	10.80
Locally recovered	43.01	20.46	66.67	30.36	15.44	3.39	89.20

F.2. Results on Computation Efficiency

As a supplement to Figure 3 (Middle), we further provide detailed numerical results on the computation efficiency. Table 12 shows that ProtoGate's training time per epoch is consistently shorter than average. Moreover, the training time differences per epoch between ProtoGate and the fastest methods are consistently smaller than 0.1s across all datasets, which is practically insignificant. Table 13 further shows that ProtoGate can efficiently make predictions for new samples. The inference time per sample of ProtoGate is shorter than 1ms, and the time differences between ProtoGate and the fastest methods are consistently smaller than 0.5ms across all datasets.

Table 12. **Training time per epoch (unit: second).** We **bold** the average training time and ProtoGate's training time per batch. ProtoGate's training time per epoch is consistently shorter than average, and the time differences per epoch between ProtoGate and the fastest methods are consistently smaller than 0.1s across all datasets.

Methods	colon	lung	meta-dr	meta-pam	prostate	tcga-2y	toxicity
TabNet	0.03	0.08	0.11	0.11	0.06	0.11	0.13
L2X	0.03	0.16	0.19	0.18	0.08	0.19	0.16
INVASE	0.06	0.10	0.11	0.11	0.12	0.11	0.12
REAL-X	0.03	0.30	0.37	0.40	0.19	0.40	0.33
LLSPIN	0.02	0.05	0.06	0.06	0.04	0.05	0.06
LSPIN	0.02	0.05	0.05	0.05	0.04	0.05	0.06
Avg. w/o ProtoGate	**0.03**	**0.12**	**0.15**	**0.15**	**0.09**	**0.15**	**0.14**
ProtoGate (Ours)	**0.02**	**0.10**	**0.12**	**0.12**	**0.08**	**0.12**	**0.13**

Table 13. **Inference time per sample (unit: millisecond).** We **bold** the average inference time and ProtoGate's inference time per sample. The inference time per sample of ProtoGate is shorter than 1ms, and the time differences per sample between ProtoGate and the fastest methods are consistently smaller than 0.5ms across all datasets.

Methods	colon	lung	meta-dr	meta-pam	prostate	tcga-2y	toxicity
TabNet	0.53	0.39	0.54	0.55	0.57	0.53	0.75
L2X	0.41	0.81	0.93	0.90	0.77	0.93	0.91
INVASE	1.04	0.51	0.54	0.54	1.16	0.55	0.69
REAL-X	0.41	1.55	1.86	1.98	1.88	2.00	1.95
LLSPIN	0.31	0.26	0.30	0.28	0.35	0.27	0.34
LSPIN	0.31	0.27	0.27	0.27	0.37	0.27	0.34
Avg. w/o ProtoGate	**0.50**	**0.63**	**0.74**	**0.75**	**0.85**	**0.76**	**0.83**
ProtoGate (Ours)	**0.17**	**0.19**	**0.28**	**0.28**	**0.61**	**0.44**	**0.59**

Table 14 further shows that ProtoGate can have much fewer trainable parameters than benchmark methods. Note that we focus on the model and exclude the shared training settings such as batch size, training epochs and learning rate when counting the hyperparameters. Because the number of trainable parameters depends on the input dimensionality, we compute the total number of trainable parameters according to the "prostate" dataset (102 samples with 5,966 features). Although the number of trainable parameters can change across datasets, the order remains the same (INVASE>L2X>REAL-X>LSPIN>LLSPIN>ProtoGate>TabNet) in our experimental settings.

Table 14. **Number of hyperparameters and trainable parameters in ProtoGate and other local methods.** We **bold** the number of parameters for ProtoGate. Benchmark methods are sorted according to the number of trainable parameters. ProtoGate can have much fewer trainable parameters than other local methods in the considered experimental settings.

Methods	# hyperparameters	# Trainable Parameters
INVASE	2	3.7M
L2X	3	2.4M
REAL-X	2	2.4M
LSPIN	2	1.8M
LLSPIN	2	1.8M
TabNet	3	0.4M
ProtoGate (Ours)	**3**	**1.2M**

F.3. Results on non-HDLSS Classification Tasks

Table 6 shows that ProtoGate consistently ranks within the top three for all datasets, implying robust performance. Although ProtoGate is primarily designed for HDLSS regimes, we find it can also perform quite well in other challenging tasks, even tied in average rank with LightGBM. Specifically, ProtoGate ranks first on the "100-plants-texture" dataset, surpassing other local methods by a clear margin. Note that REAL-X even failed to converge well on the "100-plants-texture" dataset, highlighting its limitations in managing datasets with a high number of classes. Moreover, ProtoGate can achieve competitive performance on non-biomedical datasets, such as the "mfeat-fourier" dataset. These findings indicate that ProtoGate's applicability may well extend beyond its original scope in HDLSS biomedical applications.

Table 15. **Classification accuracy (%) on four non-HDLSS real-world tabular datasets.** We report the mean ± std balanced accuracy and average accuracy rank across datasets. A higher rank implies higher accuracy. We highlight the **First**, **Second** and **Third** ranking accuracy for each dataset. ProtoGate consistently ranks Top-3 across all datasets and achieves comparable overall performance as the state-of-the-art methods.

Methods	100-plants-texture	cnae-9	mfeat-fourier	vehicle	**Rank**
MLP	$19.10_{\pm 2.03}$	$64.52_{\pm 3.10}$	$79.76_{\pm 1.89}$	**77.87**$_{\pm 3.41}$	$4.50_{\pm 2.29}$
Lasso	$52.89_{\pm 2.46}$	$74.92_{\pm 3.67}$	$76.86_{\pm 1.54}$	$70.62_{\pm 2.68}$	$5.00_{\pm 0.71}$
RF	**64.81**$_{\pm 2.34}$	**87.30**$_{\pm 1.79}$	**80.31**$_{\pm 1.89}$	$72.86_{\pm 2.46}$	**3.25**$_{\pm 0.43}$
LightGBM	**75.73**$_{\pm 2.34}$	**92.96**$_{\pm 1.45}$	**81.96**$_{\pm 1.76}$	**75.87**$_{\pm 2.92}$	**1.75**$_{\pm 0.43}$
REAL-X	$9.89_{\pm 1.33}$	$70.81_{\pm 3.00}$	$63.61_{\pm 1.75}$	$46.81_{\pm 1.99}$	$6.75_{\pm 0.43}$
LSPIN	$49.33_{\pm 1.84}$	$79.54_{\pm 2.47}$	$77.49_{\pm 1.76}$	$68.98_{\pm 2.12}$	$5.00_{\pm 0.71}$
ProtoGate (Ours)	**77.61**$_{\pm 1.42}$	**87.80**$_{\pm 2.59}$	**82.84**$_{\pm 1.60}$	**73.63**$_{\pm 2.63}$	**1.75**$_{\pm 0.83}$

G. Additional Ablation Results on Global-to-local Feature Selection

G.1. Results on Features Selection Sparsity

We perform ablation experiments to illustrate the interplay between λ_{global} and λ_{local}. Note that we summarise the combinations into three cases: ProtoGate (both regularisations: $\lambda_{\text{global}} \neq 0, \lambda_{\text{local}} \neq 0$), ProtoGate-global (only ℓ_1-regularisation: $\lambda_{\text{global}} \neq 0, \lambda_{\text{local}} = 0$) and ProtoGate-local (only ℓ_0-regularisation: $\lambda_{\text{global}} = 0, \lambda_{\text{local}} \neq 0$). Specifically, we perform grid search within ranges of $\lambda_{\text{global}} \in \{0, 1e-4, 3e-4, 5e-4, 7e-4, 1e-3\}$ and $\lambda_{\text{local}} \in \{0, 5e-4, 1e-3, 2e-3\}$.

Table 16 shows that ProtoGate and ProtoGate-local generally selects fewer features than ProtoGate-global. Note that we tried manually increasing λ_{global} in ProtoGate-global to attain a sparsity level similar to ProtoGate, but it substantially degrades its performance. Note that although when ProtoGate and ProtoGate-local achieve similar sparsity, the accuracy gap remains. This demonstrates that ProtoGate selects different features to ProtoGate-local, showing the effects of soft global selection.

Table 16. **Number of selected features of ProtoGate and its two variants.** We report the mean ± std of the number of selected features on test samples, averaged over 25 runs. We **bold** the fewest selected features for each dataset.

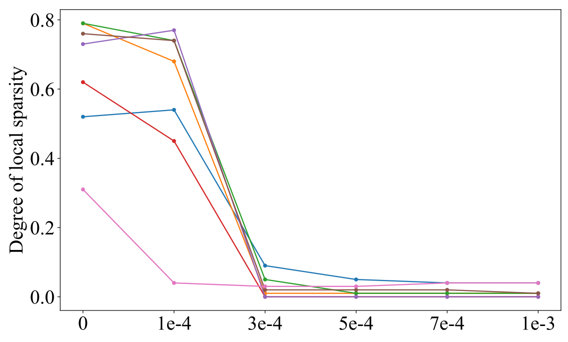
Cases	colon	lung	meta-dr	meta-pam	prostate	tcga-2y	toxicity
ProtoGate-global	$846.48_{\pm 95.32}$	$1673.92_{\pm 250.13}$	$2089.16_{\pm 265.44}$	$2601.15_{\pm 170.17}$	$2542.48_{\pm 108.79}$	$1667.31_{\pm 84.74}$	$2367.68_{\pm 53.67}$
ProtoGate-local	$115.96_{\pm 16.65}$	$\mathbf{82.18}_{\pm 14.53}$	$\mathbf{85.22}_{\pm 12.57}$	$\mathbf{69.85}_{\pm 23.20}$	$291.13_{\pm 58.48}$	$\mathbf{131.43}_{\pm 15.52}$	$154.00_{\pm 25.11}$
ProtoGate	$\mathbf{110.70}_{\pm 58.62}$	$177.24_{\pm 173.13}$	$337.21_{\pm 738.81}$	$469.47_{\pm 46.90}$	$\mathbf{91.29}_{\pm 7.20}$	$348.79_{\pm 869.23}$	$\mathbf{76.39}_{\pm 17.42}$

G.2. Results on Degree of Local Selection

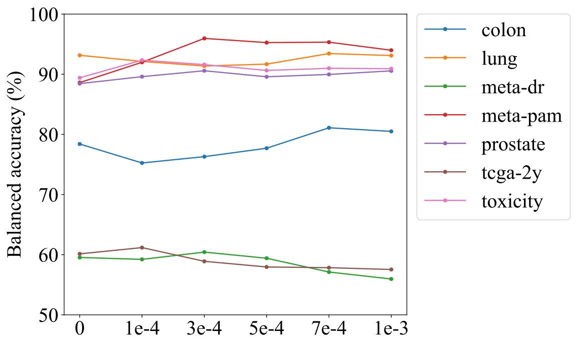
To further distinguish between "similar number of selected features" and "similar selected features", **we introduce a new metric: degree of local sparsity** $\mathcal{Q}$, which is computed by:

$$\mathcal{Q} = \frac{1}{D \cdot N} \sum_{j=1}^{N} \text{card} \left(\bigcup_{i=1}^{N} \text{nonzero}(\boldsymbol{s}_{\text{local}}^{(i)}) - \text{nonzero}(\boldsymbol{s}_{\text{local}}^{(j)}) \right) \tag{15}$$

where card$(\cdot)$ returns the cardinality of a set and nonzero$(\cdot)$ returns the indices of non-zero elements in a vector. $\mathcal{Q}$ measures the difference between the union set of selected features for all samples and the selected features for a specific sample. Intuitively, a smaller $\mathcal{Q}$ denotes the method selects features more globally than a bigger $\mathcal{Q}$. We perform ablation experiments with $\lambda_{\text{local}} = 1e-3$ and $\lambda_{\text{global}} \in \{0, 1e-4, 3e-4, 5e-4, 7e-4, 1e-3\}$, and report the average $\mathcal{Q}$ and average balanced accuracy over 25 runs.



(a) Degree of local sparsity under different λ_{global}



(b) Balanced accuracy under under different λ_{global}

Figure 10. **Comparison of different values of global sparsity hyperparameter λ_{global}.** (a) Degree of local sparsity averaged over 25 runs. Increasing λ_{global} reduces the diversity of selected features across samples. (b) Balanced accuracy (%) averaged over 25 runs. Increasing λ_{global} does not guarantee improvement in the prediction accuracy. Note that confidence intervals are omitted for visualisation purposes.

Table 17. Statistical analysis of performances between ProtoGate with optimal $\lambda^*_{\text{global}}$ and $\lambda_{\text{global}} = 1e-3$.

	lung	meta-dr	meta-pam	prostate	tcga-2y	toxicity	colon	Wilcoxon test
P-values	0.8856	0.0112	0.1638	0.9919	0.1272	0.3517	0.8252	0.0156

Figure 10 shows that increasing λ_{global} enables ProtoGate to select features more globally. Namely, increasing λ_{global} can promote the existence of global important features by performing soft global selection via ℓ_1-regularisation on $\mathbf{W}^{[1]}$. We performed statistical analysis between test accuracy with optimal $\lambda^*_{\text{global}}$ and maximum $\lambda_{\text{global}} = 1e-3$. Specifically, we performed a *corrected* 2-tailed t-test as well as Wilcoxon signed-rank test of the mean accuracies (Nadeau & Bengio, 1999; Bouckaert & Frank, 2004). We find that while models with optimal $\lambda^*_{\text{global}}$ generally tend to perform better than their counterparts, this difference is statistically significant (at $\alpha = 0.05$) **only** in the case of "meta-dr" dataset, which highlights the benefits of leveraging the locally important features across samples. On the other datasets, leveraging the locally important features is less effective, which helps to explain why global methods (e.g., Lasso) can generally outperform some local methods in Table 2. The results show that ProtoGate can easily adapt and effectively balance global and local feature selection for a particular dataset, rather than exclusively regularising for either of them.

H. Additional Ablation Results on Non-parametric Prototype-based Prediction

H.1. Results on Different Predictors

To show the efficacy of clustering assumption in prediction tasks, we replace the differentiable KNN predictor with (i) a linear head and (ii) an MLP, and then tune the hyperparameter for global sparsity λ_{global} by searching within $\{1e-4, 2e-4, 3e-4\}$ for fair comparison between different predictors. Table 18 shows that differentiable KNN consistently outperforms other predictors across datasets, suggesting that the clustering assumption is beneficial for feature selection.

Table 18. **Classification accuracy (%) for different predictors on real-world datasets.** We **bold** the highest accuracy for each dataset. The prototype-based classifier consistently outperforms linear and MLP predictors on all datasets.

Predictors	colon	lung	meta-dr	meta-pam	prostate	tcga-2y	toxicity
MLP	$80.95_{\pm 7.77}$	$69.97_{\pm 9.17}$	$56.00_{\pm 6.37}$	$93.62_{\pm 6.04}$	$89.13_{\pm 6.36}$	$54.74_{\pm 8.11}$	$90.36_{\pm 5.61}$
Linear Head	$79.45_{\pm 6.23}$	$66.51_{\pm 12.45}$	$56.10_{\pm 8.95}$	$93.20_{\pm 6.18}$	$89.87_{\pm 5.80}$	$56.60_{\pm 8.20}$	$90.29_{\pm 5.93}$
Differentiable KNN (Ours)	$\mathbf{83.95}_{\pm 9.82}$	$\mathbf{93.56}_{\pm 6.29}$	$\mathbf{60.43}_{\pm 7.62}$	$\mathbf{95.96}_{\pm 3.93}$	$\mathbf{90.58}_{\pm 5.72}$	$\mathbf{61.18}_{\pm 6.47}$	$\mathbf{92.34}_{\pm 5.67}$

H.2. Results on Different Number of Nearest Neighbours K

To evaluate the robustness of the prototype-based predictor, we further conduct experiments using different numbers of nearest neighbours denoted as K. Considering the limited sample sizes of the datasets under investigation, we set the maximum number of nearest samples to $K = 5$. All experimental settings are kept consistent to ensure fair comparisons.

Table 19 presents the results of the ablation experiments on the number of nearest neighbours, demonstrating that the optimal value of K varies across different datasets. It is observed that using a small value of K can make the predictions more sensitive to noise and outliers, resulting in lower accuracy. Notably, ProtoGate consistently achieves high accuracy across the range of $K \in \{3, 4, 5\}$. This finding supports the validity of the prototype-based prediction for the considered real-world datasets and ProtoGate's robustness in prediction tasks.

Table 19. **Classification accuracy (%) for different numbers of the nearest neighbours.** We **bold** the highest accuracy for each dataset. A small $K \in \{1, 2\}$ can lead to sensitivity to noise, and the model performs stably with $K \in \{3, 4, 5\}$.

# Prototypes	colon	lung	meta-dr	meta-pam	prostate	tcga-2y	toxicity
$K = 1$	$70.40_{\pm 14.45}$	$87.53_{\pm 7.28}$	$50.50_{\pm 6.21}$	$73.02_{\pm 10.90}$	$75.91_{\pm 10.21}$	$57.46_{\pm 6.85}$	$75.85_{\pm 7.02}$
$K = 2$	$77.35_{\pm 13.46}$	$92.30_{\pm 7.28}$	$56.06_{\pm 7.29}$	$90.28_{\pm 6.01}$	$86.93_{\pm 7.33}$	$59.40_{\pm 6.24}$	$88.81_{\pm 7.01}$
$K = 3$	$\mathbf{83.95}_{\pm 9.82}$	$\mathbf{93.56}_{\pm 6.29}$	$57.82_{\pm 8.93}$	$\mathbf{95.96}_{\pm 3.93}$	$89.53_{\pm 5.64}$	$\mathbf{61.18}_{\pm 6.47}$	$91.14_{\pm 5.19}$
$K = 4$	$75.25_{\pm 13.34}$	$90.34_{\pm 7.01}$	$\mathbf{60.43}_{\pm 7.62}$	$95.03_{\pm 4.77}$	$88.85_{\pm 5.87}$	$60.97_{\pm 5.60}$	$91.10_{\pm 4.93}$
$K = 5$	$77.50_{\pm 8.67}$	$91.12_{\pm 6.36}$	$59.23_{\pm 6.88}$	$95.83_{\pm 5.89}$	$\mathbf{90.58}_{\pm 5.72}$	$60.84_{\pm 5.88}$	$\mathbf{92.34}_{\pm 5.67}$

H.3. Results on Hybrid Sorting

To gauge the effect of NeuralSort, we run a proof-of-principle analysis comparing our proposed ProtoGate (w/ HybridSort) to the one where we substitute the sorting mechanisms with only classical QuickSort or differentiable NeuralSort. The results and conclusions are below:

Firstly, Table 20 shows that ProtoGate w/ only QuickSort (as expected) leads to faster training than HybridSort. Nevertheless, given the problem settings we are focusing on (high-dimensional and low-sample-size tabular data), these differences are practically insignificant. Secondly, Table 21 shows the predictive performance of using only QuickSort substantially degrades w.r.t our proposed ProtoGate (w/ HybridSort). This large performance gap indicates that a differentiable sorting operator is important for learning a well-performing feature selector since it allows the clustering assumption to be explicitly encoded into the optimisation procedure (via backpropagating gradients from the sorted samples/prototypes). Thirdly, we also find that using QuickSort for training can be less stable on the considered datasets, resulting in different and sometimes larger selected feature sets (up to 40% of all features), compared to the ones obtained with HybridSort (consistently below 15%).

In a nutshell, HybridSort is an effective sorting operator for prototype-based classification, which reduces the inference time by almost half while preserving the identical predictive performance as only using the computationally expensive differentiable sorting operators.

Table 20. **Training time (per epoch) and inference time (per sample) with different sorting operators (unit: millisecond).** ProtoGate w/ only QuickSort leads to faster training and inference.

Methods	Stages	colon	lung	meta-dr	meta-pam	prostate	tcga-2y	toxicity
QuickSort	training	10.66	36.78	56.72	56.09	61.76	88.91	100.06
	inference	0.17	0.19	0.28	0.28	0.61	0.44	0.59
NeuralSort	training	17.63	99.42	118.44	117.50	80.96	120.31	133.33
	inference	0.28	0.50	0.59	0.59	0.79	0.60	0.78
HybridSort (Ours)	training	17.63	99.42	118.44	117.50	80.96	120.31	133.33
	inference	0.17	0.19	0.28	0.28	0.61	0.44	0.59

Table 21. **Classification accuracy (%) of ProtoGate with different sorting operators.** We **bold** the highest accuracy for each dataset. HybridSort and NeuralSort outperform QuickSort in predictive accuracy by a clear margin.

Methods	colon	lung	meta-dr	meta-pam	prostate	tcga-2y	toxicity
QuickSort	67.25$_{\pm 8.17}$	59.46$_{\pm 12.51}$	49.11$_{\pm 1.79}$	65.78$_{\pm 14.63}$	81.70$_{\pm 11.80}$	55.00$_{\pm 2.89}$	67.57$_{\pm 7.34}$
NeuralSort	**83.95**$_{\pm 9.82}$	**93.56**$_{\pm 6.29}$	**60.43**$_{\pm 7.62}$	**95.96**$_{\pm 3.93}$	**90.58**$_{\pm 5.72}$	**61.18**$_{\pm 6.47}$	**92.34**$_{\pm 5.67}$
HybridSort (Ours)	**83.95**$_{\pm 9.82}$	**93.56**$_{\pm 6.29}$	**60.43**$_{\pm 7.62}$	**95.96**$_{\pm 3.93}$	**90.58**$_{\pm 5.72}$	**61.18**$_{\pm 6.47}$	**92.34**$_{\pm 5.67}$

I. Additional Results on Interpretability Evaluation

I.1. Alternative Metrics for Interpretability Evaluation

We carefully assess the existing metrics before performing the interpretability evaluation, and we find that some of them are debatable, even with potential flaws.

Faithfulness. The faithfulness in feature selection promotes that "all selected features should be significant for predictions". Firstly, faithfulness (Yang et al., 2022; Alvarez Melis & Jaakkola, 2018) can be computationally impractical for high-dimensional datasets. The real faithfulness should be evaluated in four steps: (i) Remove a feature; (ii) Retrain the model on the new datasets without the dropped feature; (iii) Compute the correlation between the accuracy drop and the feature importance; (iv) Repeat the above three steps for all features. On high-dimensional datasets, Step-(ii) requires retraining the model thousands of times. Secondly, we understand that some prior work (Yang et al., 2022) omits Step-(ii) to compute a surrogate of real faithfulness, but omitting Step-(ii) means Step-(iii) evaluates the model with a different distribution to which the model is trained on. Following prior work (Jethani et al., 2021), our experiments are also based on the common assumption in machine learning: models are trained and evaluated on data with the same distribution (i.e., the training and test data are independent and identically distributed). Evaluating the surrogate of faithfulness can violate this key assumption (Jethani et al., 2021; Hooker et al., 2019). Thirdly, the transferability of selected features directly demonstrates whether the selected features are overall informative for accurate predictions with downstream predictors (Yang et al., 2022). Therefore, we choose to provide an analysis of the transferability of selected features to avoid providing possibly misleading guidance for users.

Stability. The stability in feature selection promotes "samples similar in the input space should have similar informative features" (Yang et al., 2022). Firstly, the poor accuracy of vanilla KNN shows that samples of the same class are not always close (i.e., similar) to each other in the original high-dimensional space. In other words, the similarity in the original high-dimensional space can be misleading. Therefore, leveraging the similarity in the original high-dimensional space can harm the model performance. Secondly, in our experiments, we also find that LSPIN without such stability regularisation generally performs better than LSPIN with it. The highest accuracy of LSPIN (reported in Table 2) is consistently achieved with stability strength of 0 (i.e., no stability regularisation) across datasets. The results further show the negative effects of such stability regularisation. Thirdly, we further analyse the source of such negative effects. The real clustering assumption promotes "samples of the same class should have similar representations" (Chapelle et al., 2006). However, "similar masks" is not a sufficient condition for "similar representations". Technically, "similar representations" refer to the similarity between masked samples (i.e., $\mathbf{x}^{(i)} \odot \mathbf{s}_{\text{local}}^{(i)}$), rather than the masks (i.e., $\mathbf{s}_{\text{local}}^{(i)}$). Therefore, similar samples in the input space are not expected to have similar masks. This is also the reason why ProtoGate can achieve high accuracy by attending to the similarity between representations, instead of masks.

Diversity. The diversity in feature selection promotes "samples of different classes should have different informative features". Firstly, we show with vanilla KNN (Table 2) that the similarity (also dissimilarity) in the original high-dimensional space can be misleading for feature selection and prediction. Therefore, leveraging the similarity in the original high-dimensional space can harm the model performance. Secondly, the real clustering assumption promotes "samples of the same class should have similar representations" (Chapelle et al., 2006). However, "similar masks" is not a sufficient condition for "similar representations", and "dissimilar masks" is also not a sufficient condition for "dissimilar representations". In other words, similar samples are not necessarily expected to have similar masks. Likewise, dissimilar samples are not necessarily expected to have dissimilar masks. Therefore, we chose not to evaluate stability in the considered tasks to avoid possibly misleading guidance for users.

I.2. Results on Fidelity of Selected Features

In Table 22, ProtoGate achieves better or comparable performance in feature selection and classification than the benchmark methods on Syn1$_{(+)}$ and Syn2$_{(+)}$. On Syn3$_{(-)}$, ProtoGate performs poorly as expected (see Appendix E.2 for the reason), verifying the inductive bias of the clustering assumption. We also find the LSPIN exhibits visible misalignment in feature selection and prediction. On Syn1$_{(+)}$, LSPIN achieves the best classification accuracy, but the quality of selected features is much worse, with a rank of six out of ten methods. In other words, LSPIN simply overfits the dataset without correctly identifying the informative features, denoting a severe co-adaptation problem and low-fidelity feature selection. In contrast, ProtoGate has consistently non-positive rank differences between F1$_{\text{selec}}$ and ACC$_{\text{pred}}$, showing high-fidelity feature selection. The results demonstrate that ProtoGate can achieve a well-aligned performance of feature selection and classification, guaranteeing the fidelity of selected features.

Table 22. **Evaluation comparison of ProtoGate and nine benchmark methods on three synthetic datasets.** We report the F1 score of selected features ($\mathrm{F1}_{\mathrm{select}}$) and the balanced accuracy for prediction ($\mathrm{ACC}_{\mathrm{pred}}$). "Diff." refers to the difference between the ranks of $\mathrm{F1}_{\mathrm{select}}$ and $\mathrm{ACC}_{\mathrm{pred}}$, and a positive value indicates a high possibility of co-adaptation. We highlight the **First**, **Second** and **Third** performance for each dataset. ProtoGate achieves well-aligned performance for feature selection and prediction.

Methods	$\mathrm{Syn1}_{(+)}$			$\mathrm{Syn2}_{(+)}$			$\mathrm{Syn3}_{(-)}$		
	$\mathrm{F1}_{\mathrm{select}} \uparrow$	$\mathrm{ACC}_{\mathrm{pred}} \uparrow$	**Diff.**	$\mathrm{F1}_{\mathrm{select}} \uparrow$	$\mathrm{ACC}_{\mathrm{pred}} \uparrow$	**Diff.**	$\mathrm{F1}_{\mathrm{select}} \uparrow$	$\mathrm{ACC}_{\mathrm{pred}} \uparrow$	**Diff.**
Lasso	$0.09_{\pm 0.02}$	$54.55_{\pm 6.14}$	2	$0.11_{\pm 0.01}$	$52.42_{\pm 6.69}$	0	$0.09_{\pm 0.02}$	$55.30_{\pm 7.44}$	2
RF	$0.15_{\pm 0.04}$	$57.08_{\pm 6.48}$	3	$0.19_{\pm 0.02}$	**59.44**$_{\pm 5.24}$	1	**0.22**$_{\pm 0.02}$	$56.33_{\pm 9.08}$	-1
STG	**0.27**$_{\pm 0.04}$	**58.65**$_{\pm 9.03}$	-1	**0.22**$_{\pm 0.09}$	$58.28_{\pm 8.36}$	-2	**0.28**$_{\pm 0.18}$	$54.00_{\pm 9.09}$	-7
TabNet	$0.08_{\pm 0.02}$	$48.59_{\pm 6.55}$	1	$0.06_{\pm 0.02}$	$49.57_{\pm 5.38}$	0	$0.06_{\pm 0.02}$	$48.45_{\pm 8.31}$	0
L2X	$0.16_{\pm 0.07}$	$52.89_{\pm 7.51}$	-3	$0.19_{\pm 0.10}$	$55.78_{\pm 6.97}$	-1	$0.10_{\pm 0.09}$	$55.92_{\pm 7.30}$	2
INVASE	$0.18_{\pm 0.05}$	$55.36_{\pm 9.00}$	-1	$0.16_{\pm 0.03}$	**60.28**$_{\pm 8.61}$	6	$0.13_{\pm 0.03}$	**58.75**$_{\pm 8.70}$	5
REAL-X	**0.19**$_{\pm 0.04}$	$47.54_{\pm 9.51}$	-7	**0.23**$_{\pm 0.07}$	$55.20_{\pm 6.38}$	-6	**0.26**$_{\pm 0.06}$	**56.48**$_{\pm 9.34}$	-1
LSPIN	$0.15_{\pm 0.04}$	**59.04**$_{\pm 9.24}$	5	$0.19_{\pm 0.04}$	$59.40_{\pm 8.07}$	1	$0.19_{\pm 0.06}$	**58.09**$_{\pm 6.41}$	2
LLSPIN	$0.11_{\pm 0.02}$	$54.96_{\pm 9.49}$	2	$0.17_{\pm 0.08}$	$56.18_{\pm 5.80}$	1	$0.10_{\pm 0.06}$	$52.35_{\pm 8.32}$	-2
ProtoGate (Ours)	**0.29**$_{\pm 0.07}$	**58.68**$_{\pm 6.28}$	-1	**0.29**$_{\pm 0.09}$	**60.67**$_{\pm 8.21}$	0	$0.17_{\pm 0.06}$	$56.16_{\pm 6.82}$	0

I.3. Results on Transferability of Selected Features

Figure 11 shows that ProtoGate consistently improves the performance of KNN across datasets, while ProtoGate-global and ProtoGate-local can cause performance degradation on some datasets. Although SVM does not have the same inductive bias as ProtoGate (i.e., the clustering assumption), the selected features by ProtoGate do *not* cause performance degradation in SVM, even bringing notable improvements on some datasets (e.g., 25.8% increase in balanced accuracy on the "meta-pam" dataset). In contrast, ProtoGate-global and ProtoGate-local can cause considerable performance degradation (e.g., over 10% drop in balanced accuracy on the "toxicity" dataset), further showing the complementary effects of soft global selection and local selection. In a nutshell, both global and local feature selection contribute to the selection behaviours of ProtoGate, and the features selected by ProtoGate are generally transferable and beneficial for the performance of simple models.

Table 23. **Normalised balanced accuracy (%) of simple models with different feature selectors.** We **bold** the highest accuracy for each simple model. ProtoGate selects features that generally improve the performance of KNN and SVM.

Methods	None	ProtoGate-global	ProtoGate-local	**ProtoGate (Global-to-local)**
KNN	$35.08_{\pm 20.07}$	$38.24_{\pm 24.54}$	$64.02_{\pm 9.47}$	**79.45**$_{\pm 11.22}$
SVM	$44.22_{\pm 15.96}$	$31.24_{\pm 21.90}$	$43.45_{\pm 29.58}$	**71.38**$_{\pm 23.95}$

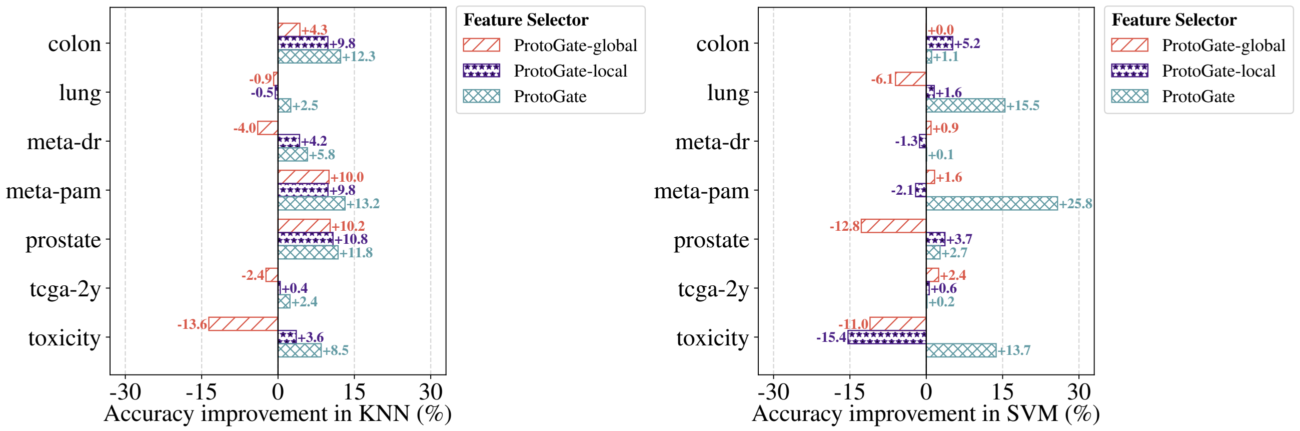


Figure 11. **Left:** Accuracy improvement in KNN (%) with different feature selectors. **Right:** Accuracy improvement in SVM (%) with different feature selectors. ProtoGate selects features that generally improve the performance of KNN and SVM, while ProtoGate-global and ProtoGate-local can cause performance degradation on some datasets.

J. Future Work

We would like to emphasise that no discussed methods, including ProtoGate, dominate the considered datasets. Indeed, ProtoGate could be outperformed by the other top three methods, Lasso and MLP, on some HDLSS datasets, e.g., “lung” dataset. This is likely due to the inherently limited expressiveness of prototype-based models in comparison to well-regularized connectionist models, as discussed in (Lu et al., 2017; Margeloiu et al., 2023a). However, the robustness and prototypical explainability offered by ProtoGate contribute valuable perspectives for advancing reliable HDLSS feature selection methods. A promising direction for future research is the development of more expressive non-parametric models, which can potentially enhance feature selection models with the same learning paradigm as ProtoGate (i.e., disjoint in-model selection). In addition, the clustering characteristics of the samples after feature selection, in conjunction with domain knowledge, could reveal important scientific insights. Moreover, exploring the application of ProtoGate in other high-dimensional fields where explainability is crucial, such as finance (Song et al., 2019), and environmental modelling (Jarquín et al., 2014), could yield important findings.