Active Feature Acquisition Via Explainability-driven Ranking

Osman Berke Guney [1] **Ketan Suhaas Saichandran** [2 3] **Karim Elzokm** [1] **Ziming Zhang** [4]
Vijaya B. Kolachalama [2 3 5]

Abstract

In many practical applications, including medicine, acquiring all relevant data for machine learning models is often infeasible due to constraints on time, cost, and resources. This makes it important to selectively acquire only the most informative features, yet traditional static feature selection methods fall short in scenarios where feature importance varies across instances. Here, we propose an active feature acquisition (AFA) framework, which dynamically selects features based on their importance to each individual case. Our method leverages local explanation techniques to generate instance-specific feature importance rankings. We then reframe the AFA problem as a feature prediction task, introducing a policy network grounded in a decision transformer architecture. This policy network is trained to select the next most informative feature by learning from the feature importance rankings. As a result, features are acquired sequentially, ordered by their predictive significance, leading to more efficient feature selection and acquisition. Extensive experiments on multiple datasets demonstrate that our approach outperforms current state-of-the-art AFA methods in predictive accuracy and feature acquisition efficiency. These findings highlight the promise of an explainability-driven AFA strategy in scenarios where feature acquisition is a concern.

1. Introduction

In traditional machine learning, all features are typically assumed to be available at inference. However, in real-world settings, feature acquisition is often costly, time-consuming, and sequential. Developing models that can make accurate predictions while minimizing feature acquisition is important for efficiency and practical implementation. This can be achieved by selecting a static global subset of features, but it is suboptimal since the important set of features may vary across different instances (Kachuee et al., 2019; Covert et al., 2023b). Furthermore, the chosen subset might not provide sufficient information for some cases, which requires the acquisition of more features to ensure a confident prediction. A more effective strategy is to identify important features sequentially for each individual instance, a technique known as active feature acquisition (AFA), which has gained increasing attention in recent years (He & Chen, 2022; von Kleist et al., 2023; Chattopadhyay et al., 2024).

The literature mainly contains two different ways of approaching AFA: reinforcement learning (RL)-based and greedy-based methods. Both approaches aim to develop a feature selection policy through exploration, as instance-wise feature importance rankings are typically unavailable. RL-based methods (Kachuee et al., 2019; Yin et al., 2020; von Kleist et al., 2023) train policy networks by maximizing different reward functions. While the RL-based approach is intuitive for this sequential task and theoretically capable of finding the optimal policy, empirical evidence shows that RL-based methods often underperform compared to greedy-based methods (Gadgil et al., 2024). Greedy-based methods attempt to predict the next most important available feature by calculating conditional mutual information (CMI). To compute CMI, researchers have proposed generative approaches (Rangrej & Clark, 2021; He et al., 2022) and methods based on the variational perspective (Covert et al., 2023b; Gadgil et al., 2024). However, calculating CMI directly remains challenging, and methods leveraging the variational perspective have demonstrated superior performance compared to generative alternatives.

In this work, we approached the problem by empirically observing that local explanation methods, such as SHapley Additive exPlanations (SHAP) (Lundberg & Lee, 2017)

[1] Department of Electrical & Computer Engineering, Boston University, MA, USA [2] Department of Computer Science, Boston University, MA, USA [3] Department of Medicine, Boston University Chobanian & Avedisian School of Medicine, MA, USA [4] Department of Electrical & Computer Engineering, Worcester Polytechnic Institute, MA, USA [5] Faculty of Computing & Data Sciences, Boston University, MA, USA. Correspondence to: Vijaya B. Kolachalama <vkola@bu.edu>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025. Copyright 2025 by the author(s).

and LIME (Ribeiro et al., 2016), can be utilized to identify instance-wise feature importance rankings. With this insight, we framed the AFA problem as a feature prediction task rather than a feature exploration one. Specifically, we trained our policy network to select the next unacquired feature with the highest importance ranking based on the current observations. This approach contrasts with feature exploration, which would involve searching for relevant features without predefined rankings, and instead focuses on prioritizing and acquiring the most informative feature in a structured, instance-specific manner. Our contributions are listed below:

- We demonstrated that local explanation methods effectively determine instance-specific feature importance rankings. Additionally, we showed that an ideal (oracle) policy network, following a precomputed feature acquisition order based on these rankings, outperformed state-of-the-art (SOTA) AFA techniques for any fixed feature budget. These findings empirically underscore the potential of explainability methods in guiding instance-specific feature acquisition. Similar trends have been observed in the local explanation literature (Petsiuk et al., 2018; Jethani et al., 2021; 2022), where inserting or removing features based on explanation-derived rankings leads to improvements or degradations in model performance, respectively. However, prior work has not formally compared explanation-based feature ranking approaches with AFA techniques, leaving a gap in understanding how these methods align or diverge in sequential feature acquisition tasks. Our study addresses this gap and highlights explanation-based rankings as a principled and effective baseline for AFA.

- We employed a decision transformer architecture (Chen et al., 2021) as our policy network and trained it using a two-stage approach. Our method outperformed SOTA techniques, demonstrating that feature importance rankings can be accurately inferred without direct observation.

2. Related work

Generally, the methods in the AFA literature have two networks: a policy network for feature acquisition and a prediction network for prediction with available subset of features. The AFA methods mainly differ in training their policy networks, so we only highlight those differences.

The AFA problem can be formulated as a Markov decision process (MDP) (Zubek & Dietterich, 2002; Dulac-Arnold et al., 2011); based on this formulation, there have been many RL-based approaches proposed (Dulac-Arnold et al., 2011; Shim et al., 2018; Kachuee et al., 2019; Yin et al., 2020; Li & Oliva, 2021; von Kleist et al., 2023). These methods generally train their policy networks with the objective of maximizing the defined reward functions. Namely, they try to approximate the action-value function (i.e., Q-function). For example, in (Dulac-Arnold et al., 2011), the Q-function is approximated linearly and later it is extended in (Janisch et al., 2019) using a deep Q network (Mnih et al., 2015; van Hasselt et al., 2016). A similar approach was taken by the opportunistic learning (OPL) method in (Kachuee et al., 2019). Another type of mainstream methods (Rangrej & Clark, 2021; He et al., 2022; Covert et al., 2023b; Chattopadhyay et al., 2023; Gadgil et al., 2024) are the greedy-based frameworks. These methods acquire the features by estimating the conditional mutual information (CMI) between the current available subset of features and the unacquired features. For CMI estimation, there are generative approaches (Rangrej & Clark, 2021; He et al., 2022) that use variational autoencoders (Kingma & Welling, 2013), and discriminative approaches (Covert et al., 2023b; Chattopadhyay et al., 2023; Gadgil et al., 2024) directly predicting the feature index with the highest CMI without explicitly calculating CMI. Although, the MDP formulation is theoretically appealing, RL-based methods often underperform compared to discriminative greedy-based approaches (Covert et al., 2023b; Gadgil et al., 2024).

Alternative AFA approaches also exist that avoid policy networks entirely and instead leverage imputation to guide feature acquisition (Beebe-Wang et al., 2023; Valancius et al., 2024). These approaches first identify a set of nearest neighbors from the training data based on the currently available features, which are then used to generate an ensemble of imputed instances. For example, AACO (Valancius et al., 2024) evaluates a set of candidate features by computing a loss function (weighted combination of the predictive loss and feature acquisition cost) over the ensemble of imputed instances, using the true labels of the nearest neighbors. Also, the method in (Beebe-Wang et al., 2023) applies a fixed predictor to the ensemble of imputed instances and then uses an explanation technique, i.e., SHAP (Lundberg & Lee, 2017), to acquire the next feature with the highest variance in importance scores for acquisition. Unlike our proposed framework, the approach in (Beebe-Wang et al., 2023) relies on a local explanation technique (e.g., SHAP) at inference time, which is computationally demanding. In contrast, we used feature rankings derived from explanation methods on the training data to train a policy network, thereby enabling efficient and scalable feature acquisition. Conceptually, our approach is closely related to imitation learning approaches (He et al., 2012a;b), where the policy network is trained to follow trajectories of another policy, such as a greedy one (He et al., 2016a).

In addition to AFA methods, related approaches from the budget learning literature (Trapeznikov & Saligrama, 2013;

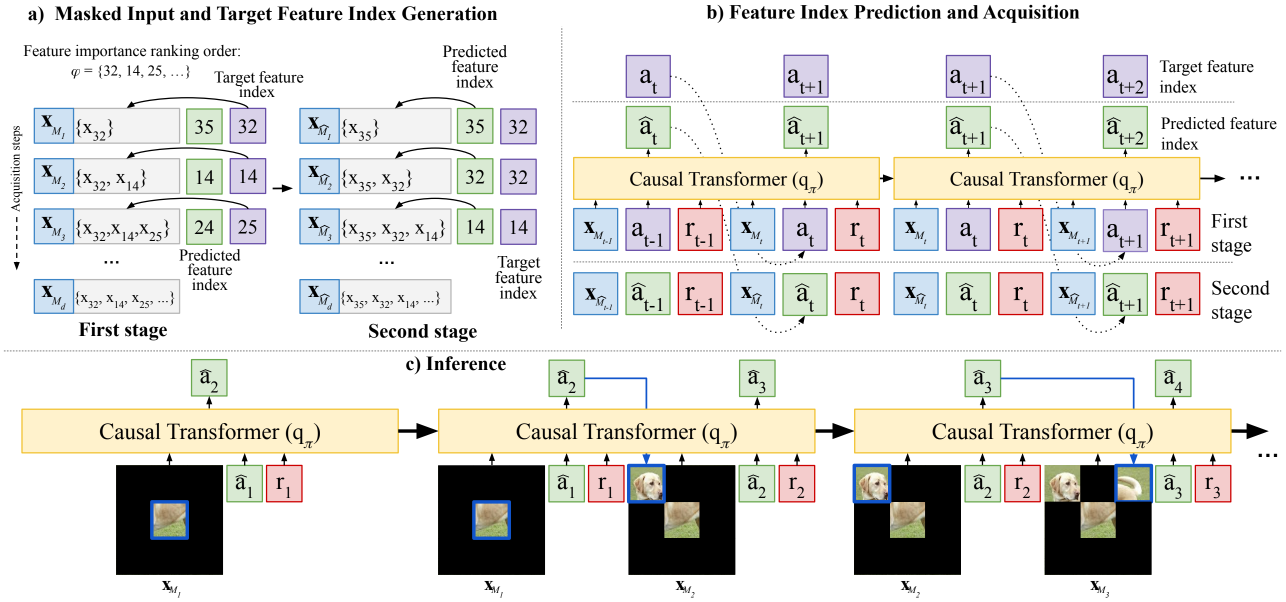
a) Masked Input and Target Feature Index Generation
Feature importance ranking order:
φ = {32, 14, 25, ...}
Target feature index
Acquisition steps
x_{M_1} {x_32} 35 32
x_{M_2} {x_32, x_14} 14 14
x_{M_3} {x_32, x_14, x_25} 24 25
...
Predicted feature index
x_{M_d} {x_32, x_14, x_25, ...}
First stage
Predicted feature index
x_{M̂_1} {x_35} 35 32
x_{M̂_2} {x_35, x_32} 32 32
x_{M̂_3} {x_35, x_32, x_14} 14 14
...
Target feature index
x_{M̂_d} {x_35, x_32, x_14, ...}
Second stage
b) Feature Index Prediction and Acquisition
a_t a_t+1 a_t+1 a_t+2
Target feature index
â_t â_t+1 â_t+1 â_t+2
Predicted feature index
Causal Transformer (q_π)
Causal Transformer (q_π)
...
x_{M_t-1} a_t-1 r_t-1 x_{M_t} a_t r_t x_{M_t} a_t r_t x_{M_t+1} a_t+1 r_t+1
First stage
x_{M̂_t-1} â_t-1 r_t-1 x_{M̂_t} â_t r_t x_{M̂_t} â_t r_t x_{M̂_t+1} â_t+1 r_t+1
Second stage
c) Inference
â_2 â_2 â_3 â_3 â_4
Causal Transformer (q_π)
Causal Transformer (q_π)
Causal Transformer (q_π)
â_1 r_1 â_1 r_1 â_2 r_2 â_2 r_2 â_3 r_3
...
x_{M_1} x_{M_1} x_{M_2} x_{M_2} x_{M_3}

Figure 1. **Explainability-driven active feature acquisition framework.** a) Our training strategy consists of two stages. This figure shows how the masked inputs and target feature indices are generated during the first and second stages for a given feature importance ranking φ. In the first stage, features are selected based on their ranking order φ. In the second stage, features are acquired by the policy network (q_π). During the first stage, the next feature in the ranking is the target feature index. However, in the second stage, the target feature is the feature index having the highest ranking order among the ones that are not acquired; because of this, the target feature remains the same until it is acquired. b) This part of the figure shows how the policy network q_π, based on the decision transformer (Chen et al., 2021), processes the masked inputs during training for both stages. Sequential data with a context length ℓ, set to 2 in this case, is fed into q_π. At each time step, q_π receives three tokens: the masked input ($\mathbf{x}_{M_t}$), action ($a_t^{(i)}$) and reward ($\mathbf{r_t}$). The action token represents the index of the last acquired feature, and the reward is the output of the predictor network. To ensure causality, future tokens are masked while q_π predicts the next feature to acquire at any time step. c) This figure illustrates the inference stage for image inputs in the causal transformer model, where predicted features (or patches) are progressively acquired in a series of sequential acquisition steps.

Nan & Saligrama, 2017; Ekanayake & Zois, 2024) explore fixed feature acquisition orders, limiting the number of potential feature subsets. These methods aim to identify easily classifiable instances, enabling the acquisition of a minimal set of features in such cases, thereby reducing overall acquisition costs.

With regards to the local explanation literature (Ribeiro et al., 2016; Petsiuk et al., 2018; Jethani et al., 2021; Lundberg & Lee, 2017), various methods focus on quantifying the contribution of individual features to model predictions for each instance. Among these methods, SHAP (Lundberg & Lee, 2017), based on game-theoretic Shapley values (Shapley, 1953), is particularly popular. SHAP calculations are computationally intensive, leading to the development of several approximations (Lundberg & Lee, 2017; Ancona et al., 2019; Jethani et al., 2022; Covert et al., 2023a). FastSHAP (Jethani et al., 2022), for instance, provides an efficient approximation using a deep explainer model. Alternatively, INVASE (Yoon et al., 2019) and L2X (Chen et al., 2018) represent approaches that learn to identify relevant instance-specific feature subsets, aiming either to explain model predictions or to highlight the most informative features for each input. Additionally, global feature importance methods aim to identify the most relevant static features across a dataset. For example, the Concrete Autoencoder (CAE) (Balın et al., 2019) selects features via an autoencoder, while SAGE (Covert et al., 2020) extends Shapley values to quantify global feature importance. For a detailed overview, we refer readers to recent surveys (Samek et al., 2021; Bolón-Canedo et al., 2022; Mesinovic et al., 2023).

3. Problem description

Let $\mathbf{x} \in \mathbb{R}^d$ represent a d-dimensional input feature vector[1], and $y \in \{1, 2, ..., C\}$ denote the associated target label, where C is the number of classes. Additionally, let $M \subseteq [d] \equiv \{1, ..., d\}$ be the subset of indices indicating the available features, and $\mathbf{x}_M$ be the masked input vector with these available features. Each j-th feature has an associated cost c_j, and each input $\mathbf{x}$ is subject to a budget constraint k. The objective is to find a predictor f_θ, parameterized with θ, and a policy network q_π, parameterized with π, such that

[1] Each feature can have different dimension size but ease of exposition, in here we have assumed each feature is one dimensional.

the following constraint objective is minimized:

$$\min_{\theta,\pi} \mathbb{E}_{\mathbf{x}yk} \mathbb{E}_{M\sim q_\pi} [\ell(f_\theta(\mathbf{x}_M), y)], \text{ s.t.} \sum_{j\in M} c_j \leq k. \quad (1)$$

Here, the first expectation is taken over the joint distribution of $\mathbf{x}$, y, and k. Including k in the joint distribution accounts for variability in the budget constraint across different instances, reflecting scenarios where the maximum allowable feature acquisition cost is input-dependent. The subset M is generated sequentially by the policy network q_π, which determines the next missing feature to acquire, i.e., $\arg\max q_\pi(\mathbf{x}_M) \in [d]\backslash M$. Subsequently, the predictor f_θ makes probabilistic predictions for any possible subset M, i.e., $f_\theta(\mathbf{x}_M) \in [0,1]^{C\times 1}$. For brevity, let the output of f_θ be denoted as $\hat{\mathbf{y}}$, i.e., $\hat{\mathbf{y}} = f_\theta(\mathbf{x}_M)$ and the output of q_π be denoted as $\hat{\mathbf{q}}$, i.e., $\hat{\mathbf{q}} = \sigma(q_\pi(\mathbf{x}_M))$, where σ is the softmax function. While softmax is also applied to the output of f_θ to obtain probabilistic predictions, we omit it from the notation for clarity.

Oracle policy network. To establish a theoretical benchmark for feature acquisition, we define the oracle policy network q^*. The oracle policy is assumed to have perfect knowledge of the true importance ranking of features for each instance $\mathbf{x}$. At each acquisition step, this policy selects the most important unobserved feature based on this ranking. The oracle policy network can be formally defined as follows: Let $M^* \subseteq [d]$ be the optimal subset of feature indices selected by the oracle policy, obtained by exhaustively evaluating all feasible subsets that satisfy the budget constraint specified in Equation 1. Once M^* is identified, the oracle constructs an internal ordering

$$M^* = \{j_1, j_2, \ldots, j_{|M^*|}\}, \quad \text{where} \quad M^* \in [d],$$

where features are sorted greedily in descending order of their importance to the prediction task. The oracle then acquires features sequentially according to this predefined order, selecting one feature at each step until all features in M^* are obtained. Notably, the internal ordering does not affect the final outcome, as all features in M^* are eventually acquired. Hence, this approach differs from a sequential greedy policy, which selects features one at a time based on marginal gains at each step. In this formulation, the oracle policy q^* is not a learnable entity but a theoretical construct. It assumes perfect knowledge of the true importance ranking of features for each instance, which is typically unavailable in real-world scenarios. Essentially, this serves as the ideal baseline and sets an upper bound on the performance achievable by any practical policy network (i.e., q_π). By comparing q_π against q^*, we can evaluate the effectiveness of the learned approach in approximating the optimal feature acquisition strategy. This oracle definition aligns with that of (Valancius et al., 2024). The main distinction is that a hard budget constraint is enforced here, whereas the oracle in (Valancius et al., 2024) incorporates feature costs into a weighted objective function.

Typically, methods in the literature (Yin et al., 2020; Covert et al., 2023b) assume that features have identical costs and that there is a fixed global budget k for all inputs. Given the available training samples $\{(\mathbf{x}^i, y^i)\}_{i=1}^N$, these methods aim to identify input-specific important features to acquire them sequentially in order of the most informative feature to the least one. To achieve this, they train q_π through exploration using reinforcement learning (RL) (Yin et al., 2020) or information-theoretic (Covert et al., 2023b) formulations, while simultaneously training the predictor network f_θ. In this work, we took a different approach by assuming access to feature importance rankings for each training sample. Consequently, instead of treating it as a feature exploration problem, we addressed it as a feature prediction problem (Figure 1).

4. Methodology and experimental settings

Feature importance ranking. In our method, we assumed access to the feature rankings φ^i for each training sample $\mathbf{x}^i$, sorted by their importance. While determining the importance of features for each input is challenging, we found that local explanation methods can effectively achieve this goal. We assumed that a model with reasonable task performance would naturally prioritize the most important instance-specific features, which can be identified using explanation methods. We empirically validated our assumption by demonstrating that when the policy network effectively selects features sequentially based on their ranking order during inference for each instance, the predictor achieves superior average performance for a given budget of k available features, outperforming the current state-of-the-art methods.

To obtain the ranking order of the features for each training instance, first, we trained a classifier using $\{(\mathbf{x}^i, y^i)\}_{i=1}^N$ with the standard cross-entropy loss minimization. Then, we run an explanation method (SHAP (Lundberg & Lee, 2017), or LIME (Ribeiro et al., 2016) etc.) to get the feature importance ranking order φ^i, where $\varphi^i(1)$ is the feature index with the highest importance and $\varphi^i(d)$ is the feature index with the least importance for the input $\mathbf{x}^i$. So our training set is $\{(\mathbf{x}^i, y^i, \varphi^i)\}_{i=1}^N$.

Policy network - Decision transformer. By approaching the problem as a conditional sequence modeling task, like the "decision transformer" (Chen et al., 2021), we trained q_π, which is a causal transformer model, with the objective of next action/token prediction. We fed q_π with sequential data and a sequence length (i.e., context length) of ℓ. At each timestep, there are three tokens including the input, the action and the reward as in Chen et al.

(2021). During training, at timestep t, the input is $\mathbf{x}^i_{M_t}$, which is the i'th sample with t many available features and $M_t = \{\varphi^i(1), ..., \varphi^i(t)\}$[2]. Whereas, the action a^i_t is the most recently acquired feature index, i.e., $a^i_t = \varphi^i(t)$ and the reward $\mathbf{r}^i_t$ is the output of the predictor with the current input, i.e., $\mathbf{r}^i_t = \hat{\mathbf{y}}^i_t = f_\theta(\mathbf{x}^i_{M_t})$. The rewards in RL-based methods (Kachuee et al., 2019; Li & Oliva, 2021) are typically functions of the predictor output; we followed a similar idea, but instead of defining a specific function, we directly fed our policy network with the predictor output. So, for a given sequence from the timestep t to $t+\ell-1$, the output of our q_π for the input i is: $\hat{\mathbf{q}}^i_t = q_\pi(\mathbf{x}^i_{M_t}, a^i_t, \mathbf{r}^i_t)$ and $\hat{\mathbf{q}}^i_{t+\ell-1} = q_\pi(\mathbf{x}^i_{M_{t:t+\ell-1}}, a^i_{t:t+\ell-1}, \mathbf{r}^i_{t:t+\ell-1})$, where $t : t+\ell-1$ indicates all the tokens from the time step t to $t+\ell-1$. We used a mini version of GPT[3] architecture (Radford, 2018) as a transformer model. Please refer to the decision transformer paper (Chen et al., 2021) for more details.

Training strategy. To train q_π, we minimized the standard cross-entropy loss by considering the index of the next feature that is not acquired with the highest importance (i.e., $\varphi^i(t+1)$) as the true label with the minibatch setting. At each iteration, the loss function is:

$$\mathcal{L}_q = -\frac{1}{N_b}\sum_{i=1}^{N_b}\sum_{t=t_i}^{t_i+\ell-1}\log(\hat{\mathbf{q}}^i_{t,\varphi^i(t+1)}), \tag{2}$$

where N_b is the batch size, $\hat{\mathbf{q}}^i_{t,\varphi^i(t+1)}$ is the $\varphi^i(t+1)$'th element of $\hat{\mathbf{q}}^i_t$, and t_i is randomly sampled integer determining the initial time step of sequence fed to the model for the i'th sample. Simultaneously, we trained the predictor f_θ also by minimizing the standard cross-entropy loss:

$$\mathcal{L}_f = -\frac{1}{N_b}\sum_{i=1}^{N}\sum_{t=t_i}^{t_i+\ell-1}\log(\hat{\mathbf{y}}^i_{t,y}). \tag{3}$$

During the first stage of training, both f_θ and q_π are fed by the input with the features that are acquired based on the ranking order provided by the local explanation method. However during inference, because q_π is not 100% accurate, the feature subset $\hat{M}_t$, generated by q_π, may not always contain the top t features with the highest ranking order. To train both models to handle this new subset of features not encountered in the first stage, we introduced a second stage of training. At the beginning of each iteration of the second stage, we first generated empirical/predicted feature acquisition $\hat{\varphi}^i$ order for each $\mathbf{x}^i$, where $\hat{\varphi}^i(t+1) = \arg\max \hat{\mathbf{q}}^i_t$ and $\hat{M}_t = \{\hat{\varphi}^i(1), \hat{\varphi}^i(2), ..., \hat{\varphi}^i(t)\}$. Then, we minimized the same losses as in the first stage with the same strategy. In $\mathcal{L}_q$, the index of the feature, which is not acquired yet and having the highest order among the features that are not acquired, is taken as the true label. For example, if $\varphi^i(1) \notin \{\hat{\varphi}^i(1), ..., \hat{\varphi}^i(t)\}$ then $\varphi^i(1)$ is taken as the true label; but if $\varphi^i(1)$ is acquired and $\varphi^i(2)$ is not acquired then $\varphi^i(2)$ is taken as the true label, i.e., $\varphi^i(1) \in \{\hat{\varphi}^i(1), ..., \hat{\varphi}^i(t)\}$ and $\varphi^i(2) \notin \{\hat{\varphi}^i(1), ..., \hat{\varphi}^i(t)\}$. By this second stage, we trained the predictor f_θ to make its prediction with the subset of features $\hat{M}_t$ acquired by q_π. Also, the policy network q_π is trained to predict the feature with the highest ranking order among the features that are not acquired using the input with the imperfect subset of features $\hat{M}_t$. This second stage helps both networks to perform better during inference, where the imperfect subset of features $\hat{M}_t$ can only be used. Note that both the predictor and policy networks are dependent on each other. However, during training, we prevented the gradient flow from one network to another. Therefore, each network has its own independent loss function; because of the dependency, we trained them simultaneously. At $t = 0$, there is no feature acquired yet, i.e., $M_0 = \emptyset$; so for all i, the outputs of q_π are the same at $t = 0$. Consequently, at $t = 0$, the same feature must be selected for acquisition across all inputs. In our approach, we initialized each input with the first feature that, on average, holds the highest importance ranking based on the training set. Detailed outlines of both training strategies are provided in the Appendix.

Implementation details. During training, we fixed the number of epochs to 200 and 16 for the first and second stage, respectively. We used Adam optimizer (Kingma & Ba, 2014) and a cosine scheduler (Loshchilov & Hutter, 2017). Before starting training, we pre-trained the predictor network, as done in (Covert et al., 2023b; Gadgil et al., 2024). We also employed a different augmentation strategy proposed in (Hoffer et al., 2020). In addition, as with other methods in the literature (Kachuee et al., 2019; Covert et al., 2023b; Gadgil et al., 2024), we shared the backbone between f_θ and q_π. We used this backbone in q_π to get the embedding of the input token. The embedding of action was extracted using a learnable embedding dictionary. For the reward embedding, we applied a linear layer followed by a non-linear activation to transform the output of f_θ into the embedding dimension. In q_π, we set context length ℓ to 4, number of heads and layers 4 and 3, respectively. We would like to clarify that we did not conduct an extensive parameter search in our experiments. The context length parameter ℓ was selected based on validation performance on the CIFAR-10 dataset, while the remaining parameters were chosen heuristically. These values were then held fixed across all experiments. For ℓ selection, we evaluated the model's performance on the CIFAR-10 validation set, averaging accuracies over the first 20 features. The resulting mean accuracies were 78.41%, 78.76%, 79.12%, and

[2] Each sample i has its own specific M_t, but we do not specify through superscript i if it is clear from the context.

[3] https://github.com/karpathy/minGPT

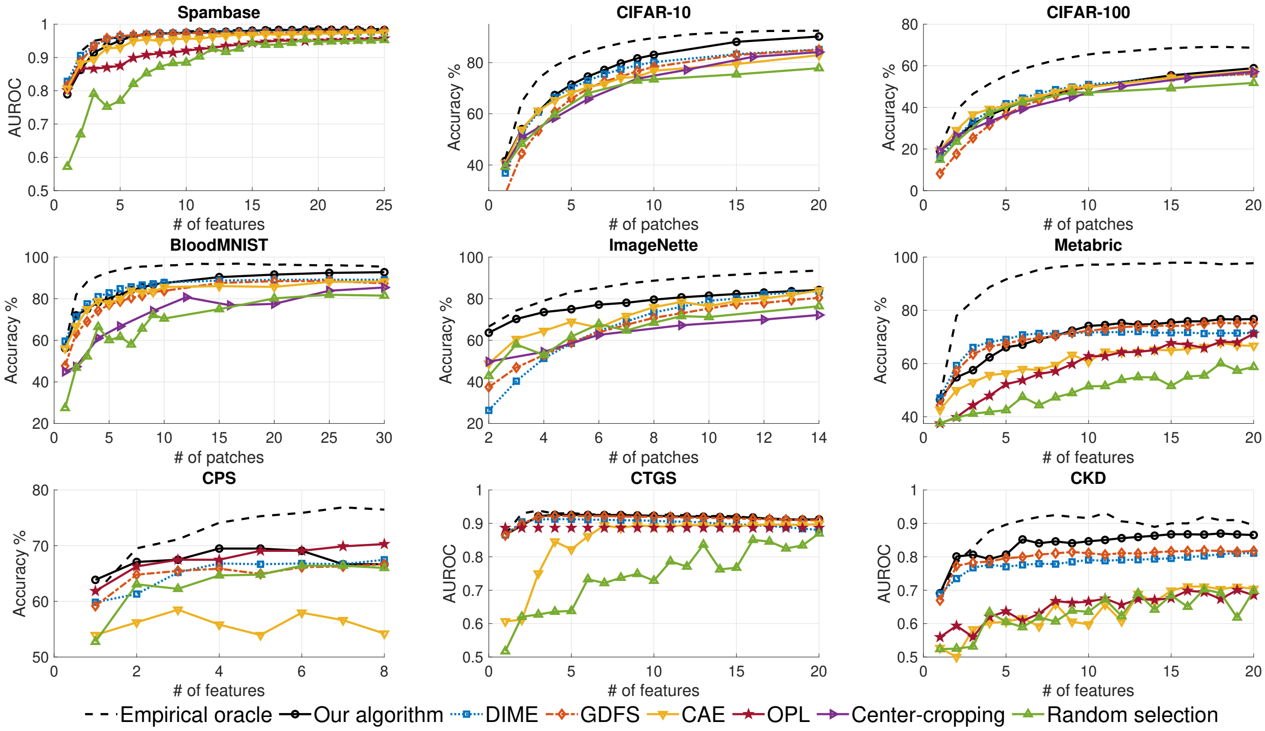

Figure 2. **Model performance.** We compared the average classification performance of our AFA method with other well-known methods across datasets with varying numbers of features. The results were averaged over three independent runs for the image datasets and nine independent runs for the tabular datasets. These datasets include five tabular datasets, four of which are medical: Spambase, Metabric, CPS, CTGS, and CKD, as well as four image datasets: CIFAR-10, CIFAR-100, BloodMNIST, and ImageNette. For improved readability, plots including standard deviations are provided in the Appendix.

78.12% for $\ell = 1$, $\ell = 2$, $\ell = 4$, and $\ell = 8$, respectively. To maintain a consistent effective batch size at each iteration (see Equations 2 and 3), we adjusted the batch size N_b inversely with ℓ. Based on these results, we selected $\ell = 4$ for all subsequent experiments. In the output of q_π, we subtracted a large constant from the logits of already acquired feature indices before applying the softmax layer to prevent re-acquisition.

Datasets. We conducted experiments on nine datasets (Table 1): five tabular datasets (Spambase, Metabric, CPS, CTGS, and CKD) and four image datasets (CIFAR-10, CIFAR-100, BloodMNIST, and ImageNette). For image datasets, we partitioned each image into non-overlapping patches. For detailed descriptions of the datasets, we refer readers to the Appendix (A.1).

Table 1. **Summary of datasets used in our experiments.** For each dataset, we list the number of features (d), classes (C), samples, and image size, along with the utilized patch size when applicable.

Dataset	d	C	# Samples	Image size Patch size
Spambase	57	2	4,601	-
CIFAR-10	64	10	60,000	32×32 4×4
CIFAR-100	64	100	60,000	32×32 4×4
BloodMNIST	196	8	17,092	28×28 2×2
ImageNette	196	10	13,395	224×224 16×16
Metabric	489	6	1,898	-
CPS	8	3	418	-
CTGS	23	2	2,139	-
CKD	50	2	1,659	-

Model architectures. To test the robustness of our method across different architectures, we varied predictor architectures. We employed ResNet50 (He et al., 2016b) for ImageNette, ResNet18 (He et al., 2016b) for the CIFAR-10 and CIFAR-100 datasets, and a custom CNN for the BloodMNIST dataset. The custom CNN has four convolution layers with output channels $16, 32, 64$, and 64, each followed by a ReLU activation and a max pooling layer. The convolution layers are followed by flattening and linear layers for classification. For the Spambase dataset, we used a multilayer perception (MLP) consisting of 2 hidden layers with 128 neurons, each followed by a ReLU and a dropout layer. On the medical tabular datasets, we utilized the same MLP architecture with 1024 hidden layer neurons on Metabric,

Table 2. **Stage-wise classification performance.** The table presents our model's performance after the first and second training stages, averaged over the first 20 features, on the Spambase, CIFAR-10, CIFAR-100, BloodMNIST, ImageNette, Metabric, CPS, CTGS and CKD datasets. Mean and standard deviation values were calculated across three independent runs for the image datasets and nine independent runs for the tabular datasets. Additionally, we included results from an extended first-stage training (250 epochs) to provide a more rigorous comparison with second-stage performance, illustrating the relative benefits of our two-stage approach versus simply extending training duration. For the Spambase, CTGS and CKD datasets, we reported the area under the receiver operating characteristic curve values, while for the remaining datasets, we provided accuracy metrics. For the CPS dataset, the average was calculated over its 8 features, as the dataset contains only 8 features.

	Spam	**CIFAR10**	**CIFAR100**	**BloodMNIST**	**ImageNette**	**Metabric**	**CPS**	**CTGS**	**CKD**
# of classes:	2	10	100	8	10	6	3	2	2
First-stage (250)	$0.952_{\pm.001}$	$75.96_{\pm0.16}\%$	$45.91_{\pm0.36}\%$	$79.83_{\pm0.19}\%$	$73.95_{\pm0.25}\%$	$62.52_{\pm1.27}\%$	$67.23_{\pm0.48}\%$	$0.916_{\pm.0002}$	$0.822_{\pm.01}$
First-stage	$0.951_{\pm.0002}$	$75.76_{\pm0.19}\%$	$46.05_{\pm0.25}\%$	$79.25_{\pm0.15}\%$	$73.76_{\pm0.42}\%$	$62.48_{\pm1.39}\%$	$67.21_{\pm0.15}\%$	$0.916_{\pm.0004}$	$0.825_{\pm.008}$
Second-stage	$0.955_{\pm.0001}$	$78.44_{\pm0.15}\%$	$46.99_{\pm0.15}\%$	$83.87_{\pm1.05}\%$	$78.96_{\pm0.12}\%$	$69.83_{\pm0.41}\%$	$67.45_{\pm0.13}\%$	$0.916_{\pm.0001}$	$0.836_{\pm.07}$

512 on CKD, 512 on CTGS and 128 on CPS.

Feature importance ranking calculations. We employed various local explanation techniques tailored to the characteristics of image and tabular datasets. For image datasets, we employed FastSHAP (Jethani et al., 2022) to generate instance-specific feature ranking orders φ^i for each input $\mathbf{x}^i$, owing to its computational efficiency and ability to handle dynamic feature importance changes during training with data augmentations. For tabular datasets, we relied on tree-based models, specifically CatBoost (Prokhorenkova et al., 2018), to derive feature ranking orders due to their strong performance on tabular data (Grinsztajn et al., 2022). Instance-specific rankings were computed using TreeSHAP (Lundberg et al., 2020), optimized for tree-based models, via the SHAP package[4]. These rankings were used in our main experiments. Additionally, we explored alternative ranking methods, including INVASE (Yoon et al., 2019), and LIME (Ribeiro et al., 2016), as well as two SHAP-based techniques, KernelSHAP (Lundberg & Lee, 2017) and IME (sampling) (Štrumbelj & Kononenko, 2010), for tabular datasets.

Comparison with the state-of-the-art methods. We evaluated our method against several existing approaches including DIME (discriminative mutual information estimation), GDFS (greedy dynamic feature selection), CAE (concrete autoencoder), OPL (RL-based method) and two baseline methods: center-cropping and random selection. DIME (Gadgil et al., 2024) prioritizes features by estimating their mutual information with the response variable in a discriminative framework. GDFS (Covert et al., 2023b) uses a greedy strategy to select features based on their conditional mutual information, employing amortized optimization to approximate the greedy policy. CAE (Balın et al., 2019) is an unsupervised, end-to-end differentiable method that employs a concrete selector layer for feature selection, gradually discretizing the selection process by lowering the temperature parameter. OPL (Kachuee et al., 2019) is a RL-based method that employs deep Q-learning, using prediction uncertainty as the reward signal during training. Due to time constraints, we evaluated OPL only on the tabular datasets. The baseline methods, center-cropping and random selection, provide simpler comparisons: center-cropping selects patches from the center of the input, while random selection picks patches arbitrarily.

Table 3. **Model performance using various feature ranking approaches.** Comparison of classification performance across five tabular datases using feature rankings derived from INVASE, a learnable instance-wise feature selection method, and various local explanation methods: TreeSHAP (T-SHAP), LIME, KernelSHAP (K-SHAP), and IME (sampling). The performance metrics are the area under the receiver operating characteristic curve for the binary-classification datasets and accuracy for the multi-class datasets. Mean and standard deviation values were calculated across nine independent runs. Results on the Metabric dataset using INVASE are omitted due to the high computational cost associated with this ranking method.

	Spam	**Metabric**	**CPS**	**CTGS**	**CKD**
# of classes:	2	6	3	2	2
T-SHAP	$0.96_{\pm0.001}$	$69.8_{\pm0.41}\%$	$67.5_{\pm0.13}\%$	$0.92_{\pm0.001}$	$0.84_{\pm0.07}$
LIME	$0.95_{\pm0.002}$	$69.2_{\pm0.18}\%$	$67.1_{\pm0.36}\%$	$0.91_{\pm0.001}$	$0.82_{\pm0.09}$
K-SHAP	$0.96_{\pm0.002}$	$69.6_{\pm0.33}\%$	$67.3_{\pm0.56}\%$	$0.92_{\pm0.001}$	$0.83_{\pm0.005}$
IME	$0.95_{\pm0.001}$	$69.8_{\pm0.10}\%$	$67.1_{\pm0.61}\%$	$0.92_{\pm0.001}$	$0.83_{\pm0.1}$
INVASE	$0.93_{\pm0.002}$	-	$68.4_{\pm0.23}\%$	$0.91_{\pm0.003}$	$0.83_{\pm0.09}$

We also evaluated an empirical oracle to approximate the optimal feature acquisition strategy. Given a predictor f_θ, the ideal feature subset M^*, as determined by the oracle policy q^*, could theoretically be identified through an exhaustive combinatorial search for any given budget k. However, performing such an optimization for every instance across multiple values of k is computationally impractical. Instead, we precomputed the optimal feature acquisition order for each instance using feature importance rankings derived from explanation methods. At inference, all features were acquired in this predetermined order, ensuring

[4] https://pypi.org/project/shap/

Table 4. **Alignment between model's feature acquisition order and the feature importance rankings.** This table presents the percentage overlap between the top T feature indices, ranked by their importance and those acquired by our method for T = 10, 15, 20, 25, and 30 on all datasets except CPS, as it only contains 8 features. Mean and standard deviation values were calculated across three independent runs for the image datasets and nine independent runs for the tabular datasets.

	Spam	**CIFAR-10**	**CIFAR-100**	**BloodMNIST**	**ImageNette**	**Metabric**	**CKD**	**CTGS**
# of features (d):	57	64	64	196	196	489	50	23
Top 10 features	$77.26_{\pm 1.06}\%$	$36.22_{\pm 0.27}\%$	$47.29_{\pm 2.25}\%$	$40.75_{\pm 2.38}\%$	$11.11_{\pm 0.11}\%$	$59.04_{\pm 1.01}\%$	$66.57_{\pm 1.44}\%$	$79.9_{\pm 0.4}\%$
Top 15 features	$82.15_{\pm 0.62}\%$	$45.83_{\pm 0.23}\%$	$57.13_{\pm 2.10}\%$	$47.94_{\pm 2.12}\%$	$16.30_{\pm 0.11}\%$	$61.5_{\pm 1.05}\%$	$69.6_{\pm 0.46}\%$	$91.1_{\pm 0.2}\%$
Top 20 features	$87.31_{\pm 0.55}\%$	$52.43_{\pm 0.24}\%$	$63.85_{\pm 1.65}\%$	$52.59_{\pm 1.85}\%$	$20.74_{\pm 0.07}\%$	$62.38_{\pm 0.60}\%$	$71.28_{\pm 0.44}\%$	$95.7_{\pm 0.2}\%$
Top 25 features	$87.64_{\pm 0.29}\%$	$57.70_{\pm 0.25}\%$	$68.10_{\pm 1.14}\%$	$55.60_{\pm 1.68}\%$	$25.06_{\pm 0.06}\%$	$62.59_{\pm 0.38}\%$	$74.21_{\pm 0.21}\%$	N/A
Top 30 features	$88.15_{\pm 0.17}\%$	$62.53_{\pm 0.28}\%$	$70.83_{\pm 0.83}\%$	$57.82_{\pm 1.46}\%$	$29.07_{\pm 0.05}\%$	$63.05_{\pm 0.84}\%$	$76.84_{\pm 0.51}\%$	N/A

Table 5. **Nearest neighbor-based feature acquisition baseline.** Performance of a baseline inspired by AACO, where features are acquired based on the importance ranking of the nearest training sample. Predictor networks from our method's second stage were reused without retraining.

	Spam	**Metabric**	**CPS**	**CTGS**	**CKD**
# of classes:	2	6	3	2	2
Our method	$0.96_{\pm 0.001}$	$69.8_{\pm 0.41}\%$	$67.5_{\pm 0.13}\%$	$0.92_{\pm 0.001}$	$0.84_{\pm 0.07}$
NN	$0.95_{\pm 0.005}$	$68.1_{\pm 0.75}\%$	$67.2_{\pm 0.22}\%$	$0.91_{\pm 0.009}$	$0.83_{\pm 0.003}$

that selections followed their instance-specific importance rankings. Additionally, to standardize initialization, each instance began with the feature that exhibited the highest average ranking across the training set.

5. Results and discussion

Figure 2 demonstrates that our method shows superior, or comparable performance on all the datasets. For example, on the ImageNette dataset, with the few number of patches, our method performs well, achieving 63.64% and 74.95% average accuracy with two and five available patches among 196 patches, respectively. Additionally, our model achieved an average AUROC score of 0.8465 on the CKD dataset with 10 features. Furthermore, the superior performance of the empirical oracle across all datasets highlights the relative potential of our approach. To assess the robustness of our method to random weight initialization and the potential variability of local explanation techniques, results were averaged over three independent runs for the image datasets and nine independent runs for the tabular datasets. Specifically, for the tabular datasets, we trained three initial models with different random seeds, producing three distinct feature ranking orders. For each ranking order, our method was trained three times to capture the variability of random network initialization. Due to computational constraints, we only varied the random initialization of the networks on the image datasets. To enhance readability, plots with standard deviation bars are included in the Appendix (Figure A1). The reported standard deviations (in Figure A1, Table 2, and Table 3) confirm that our method is robust to both sources of variability. We found that initializing inputs with three features, rather than just one, improves training stability. Based on this, we fixed first three feature acquisition order and obtained the results shown in Figure 2. The second and third features were also selected based on their average importance rankings. Note that fixing the acquisition order for all d features is equivalent to using static global feature selection methods like CAE, which is suboptimal, as our empirical results demonstrate. Therefore, initializing with more than one feature can negatively impact the achievable upper bound on performance. However, we found that fixing the acquisition order for a few initial features helps stabilize training. Additionally, since our method relies on the feature ranking order, having a better ranking can lead to improved performance. Our approach can work with any ranking order, including those provided by humans, but we have shown that local model explanation algorithms are effective in providing this order.

The average performance after both stages is shown for all the datasets in Table 2, highlighting the benefit of the second stage. The second stage provides significant improvement on most datasets, except for Spambase, CPS and CTGS that are relatively simpler compared to others, at least in terms of number of classes and features. Specifically, the second stage provides mean classification accuracy increase from 0.94% (on CIFAR-100) to 7.35% (on Metabric). Table 2 also includes results from an extended first-stage training (250 epochs) to clarify that the performance gains observed in the second stage are not solely attributable to additional training epochs. The results of the first-stage training with 250 epochs are nearly identical to those with 200 epochs, demonstrating the effectiveness of the second-stage training. However, on BloodMNIST, the additional epochs provide a meaningful performance increase. To further assess this, we conducted first-stage training with 300 epochs, yielding results of $79.73_{\pm 0.19}\%$, further reinforcing the effectiveness of the second-stage training.

We also evaluated the robustness and effectiveness of our method across different feature ranking approaches, including INVASE (Yoon et al., 2019) and LIME (Ribeiro et al., 2016), as well as two SHAP-based techniques for estimating feature importance: KernelSHAP (Lundberg & Lee, 2017) and IME (sampling) (Štrumbelj & Kononenko, 2010). These results (Table 3) indicate that while our method is robust to different ranking orders, its performance is also dependent on the quality of the ranking order generated by the explainability methods. These results were also averaged across nine independent runs, as in Table 2. To further verify the second point and test the dependency of the ranking orders' quality on the pre-trained model capacity, we conducted another ablation experiment on the CIFAR-10 dataset. Specifically, we used ResNet-10, a smaller model compared to ResNet-18, as the pre-trained model for determining the feature ranking order, while retaining ResNet-18 as the classification network. We observed that the performance of our method decreased from 78.44% to 78.22% on the test set, and from 79.12% to 78.42% on the validation set. These results confirm that the pre-trained model's capacity impacts the quality of feature ranking order and, consequently, the performance of our method. In addition, we evaluated the effectiveness of the decision transformer as the policy network by comparing our method's performance with different architectures. When the decision transformer was replaced with a ResNet block, the model's accuracy decreased from 78.44% to 76.83% on the CIFAR-10 dataset and from 46.99% to 46.70% on the CIFAR-100 dataset. Similarly, substituting the decision transformer with a CNN block reduced the model's accuracy from 83.87% to 78.23% on the BloodMNIST dataset. These results demonstrate the advantage of using a decision transformer as the policy network while highlighting that our method remains effective with alternative architectures. We further evaluated a nearest-neighbor (NN) based feature acquisition approach without any policy network, inspired by the AACO method (Valancius et al., 2024). For each masked test instance, we first identified its nearest neighbor from the training set, then determined the next feature to acquire based on that neighbor's feature importance ranking. Specifically, we selected the highest-ranked feature (according to the neighbor's ranking) that had not yet been acquired for the test instance. The results are presented in Table 5. For this baseline, we used the same predictor networks from our method's second stage. Additionally, nearest neighbors were identified using raw feature distances as in AACO, which may not be effective for image datasets.

In Table 4, we present the overlap ratios between our method's acquired feature order and the local explanation techniques-based feature importance rankings across different datasets. As the number of top features (T) increases from 10 to 30, the percentage overlap generally rises for all datasets. This trend indicates that our method's feature acquisition order increasingly aligns with the feature importance rankings as more features are considered. While the empirical oracle performances in Figure 2 demonstrate the practical benefits of using explanability-driven ranking orders in the AFA problem, Table 4 highlights the degree to which our method's acquisition strategy aligns with the feature importance rankings. We also provide example patch acquisition trajectories for both our method and the empirical oracle on the Imagenette dataset, illustrated for four classes in the Appendix (Figure A2).

We emphasize the flexibility of our proposed method, which can operate with any given feature ordering. In the absence of a definitive ground truth for feature importance rankings, we rely on explainability methods to generate these orderings. While such methods are valuable, they may not always yield optimal rankings in all settings (Kumar et al., 2020; Catav et al., 2021). As more accurate explanation techniques emerge, our approach can readily incorporate them to achieve further performance gains. Although our method demonstrates improved accuracy over baseline approaches, it incurs higher training time. However, this can be mitigated through optimizations such as early stopping and mixed-precision (half-precision) training. In practical domains like medicine, AFA is often performed using pretrained models customized to specific conditions. In such settings, explainability tools are commonly employed to support interpretability and foster clinical trust, which is an essential criterion for medical AI applications (Hill et al., 2025; Dai et al., 2024; Xue et al., 2024). Our framework is designed to take advantage of these precomputed feature importance rankings, removing the need to recompute explanations during training and thereby enabling efficient deployment. Finally, we acknowledge that our method assumes uniform feature acquisition costs. While this simplifies the framework, it may not reflect real-world conditions. Extending the method to incorporate non-uniform acquisition costs remains an important avenue for future research.

6. Conclusion

Our work introduces an active feature acquisition strategy by reframing it as a feature prediction task, where the model learns to acquire features based on explainability-driven feature importance rankings. Stage-wise results demonstrate that our two-stage training approach improves feature selection and classification performance on tabular and image datasets. The findings suggest that our method is robust across various models, datasets, and settings, and that it exhibits strong practical applicability in real-world scenarios, including domains such as medicine.

Impact statement

This paper presents work whose goal is to advance the field of machine learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

Acknowledgments

This project was supported by grants from the National Institute on Aging's Artificial Intelligence and Technology Collaboratories (P30-AG073105), the American Heart Association (20SFRN35460031), and the National Institutes of Health (R01-HL159620, R01-AG062109, R01-AG083735, and R01-NS142076).

References

Acevedo, A., Merino, A., Alférez, S., Ángel Molina, Boldú, L., and Rodellar, J. A dataset of microscopic peripheral blood cell images for development of automatic recognition systems. *Data in Brief*, 30:105474, 2020.

Ancona, M., Oztireli, C., and Gross, M. Explaining deep neural networks with a polynomial time algorithm for shapley value approximation. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 272–281. PMLR, 09–15 Jun 2019.

Balın, M. F., Abid, A., and Zou, J. Concrete autoencoders: Differentiable feature selection and reconstruction. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 444–453. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/balin19a.html.

Beebe-Wang, N., Qiu, W., and Lee, S.-I. Explanation-guided dynamic feature selection for medical risk prediction. In *ICML 3rd Workshop on Interpretable Machine Learning in Healthcare (IMLH)*, 2023. URL https://openreview.net/forum?id=1itfhff53V.

Bolón-Canedo, V., Alonso-Betanzos, A., Morán-Fernández, L., and Cancela, B. *Feature Selection: From the Past to the Future*, pp. 11–34. Springer International Publishing, Cham, 2022. ISBN 978-3-030-93052-3. doi: 10.1007/978-3-030-93052-3_2. URL https://doi.org/10.1007/978-3-030-93052-3_2.

Catav, A., Fu, B., Zoabi, Y., Meilik, A. L. W., Shomron, N., Ernst, J., Sankararaman, S., and Gilad-Bachrach, R. Marginal contribution feature importance - an axiomatic approach for explaining data. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 1324–1335. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/catav21a.html.

Chattopadhyay, A., Chan, K. H. R., Haeffele, B. D., Geman, D., and Vidal, R. Variational information pursuit for interpretable predictions. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=77lSWa-Tm3Z.

Chattopadhyay, A., Chan, K. H. R., and Vidal, R. Bootstrapping variational information pursuit with large language and vision models for interpretable image classification. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=9bmTbVaA2A.

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

Chen, L., Lu, K., Rajeswaran, A., Lee, K., Grover, A., Laskin, M., Abbeel, P., Srinivas, A., and Mordatch, I. Decision transformer: Reinforcement learning via sequence modeling. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 15084–15097. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/7f489f642a0ddb10272b5c31057f0663-Paper.pdf.

Covert, I., Lundberg, S. M., and Lee, S.-I. Understanding global feature contributions with additive importance measures. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 17212–17223. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/c7bf0b7c1a86d5eb3be2c722cf2cf746-Paper.pdf.

Covert, I. C., Kim, C., and Lee, S.-I. Learning to estimate shapley values with vision transformers. In *The Eleventh International Conference on Learning Representations*, 2023a. URL https://openreview.net/forum?id=5ktFNz_pJLK.

Covert, I. C., Qiu, W., Lu, M., Kim, N. Y., White, N. J., and Lee, S.-I. Learning to maximize mutual information for dynamic feature selection. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 6424–6447. PMLR, 23–29 Jul 2023b.

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., Gräf, S., Ha, G., Haffari, G., Bashashati, A., Russell, R., McKinney, S., Langerød, A., Green, A., Provenzano, E., Wishart, G., Pinder, S., Watson, P., Markowetz, F., Murphy, L., Ellis, I., Purushotham, A., Børresen-Dale, A.-L., Brenton, J. D., Tavaré, S., Caldas, C., and Aparicio, S. The genomic and transcriptomic architecture of 2, 000 breast tumours reveals novel subgroups. *Nature*, 486(7403):346–352, April 2012. ISSN 1476-4687. doi: 10.1038/nature10983. URL http://dx.doi.org/10.1038/nature10983.

Dai, L., Sheng, B., Chen, T., Wu, Q., Liu, R., Cai, C., Wu, L., Yang, D., Hamzah, H., Liu, Y., et al. A deep learning system for predicting time to progression of diabetic retinopathy. *Nature Medicine*, 30(2):584–594, 2024.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Dickson, E., Grambsch, P., Fleming, T., Fisher, L., and Langworthy, A. Cirrhosis Patient Survival Prediction. UCI Machine Learning Repository, 1989. DOI: https://doi.org/10.24432/C5R02G.

Dulac-Arnold, G., Denoyer, L., Preux, P., and Gallinari, P. Datum-wise classification: A sequential approach to sparsity. In Gunopulos, D., Hofmann, T., Malerba, D., and Vazirgiannis, M. (eds.), *Machine Learning and Knowledge Discovery in Databases*, pp. 375–390, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg. ISBN 978-3-642-23780-5.

Ekanayake, S. P. and Zois, D. Sequential acquisition of features and experts for datum–wise classification. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5225–5229, 2024. doi: 10.1109/ICASSP48485.2024.10447423.

Gadgil, S., Covert, I. C., and Lee, S.-I. Estimating conditional mutual information for dynamic feature selection. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Oju2Qu9jvn.

Grinsztajn, L., Oyallon, E., and Varoquaux, G. Why do tree-based models still outperform deep learning on typical tabular data? In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. URL https://openreview.net/forum?id=Fp7__phQszn.

Hammer, S. M., Katzenstein, D. A., Hughes, M. D., Gundacker, H., Schooley, R. T., Haubrich, R. H., Henry, W. K., Lederman, M. M., Phair, J. P., Niu, M., Hirsch, M. S., and Merigan, T. C. A trial comparing nucleoside monotherapy with combination therapy in hiv-infected adults with cd4 cell counts from 200 to 500 per cubic millimeter. aids clinical trials group study 175 study team. *The New England journal of medicine*, 335 15:1081–90, 1996. URL https://api.semanticscholar.org/CorpusID:40754467.

He, H., Daumé III, H., and Eisner, J. Cost-sensitive dynamic feature selection. In *International Conference on Machine Learning (ICML) workshop on Inferning: Interactions between Inference and Learning*, 2012a.

He, H., Eisner, J., and Daume, H. Imitation learning by coaching. In Pereira, F., Burges, C., Bottou, L., and Weinberger, K. (eds.), *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012b. URL https://proceedings.neurips.cc/paper_files/paper/2012/file/2dffbc474aa176b6dc957938c15d0c8b-Paper.pdf.

He, H., Mineiro, P., and Karampatziakis, N. Active information acquisition. *arXiv*, 2016a.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016b.

He, W. and Chen, T. Scalable online disease diagnosis via multi-model-fused actor-critic reinforcement learning. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, pp. 4695–4703. Association for Computing Machinery, 2022. ISBN 9781450393850. doi: 10.1145/3534678.3542672.

He, W., Mao, X., Ma, C., Huang, Y., Hernàndez-Lobato, J. M., and Chen, T. Bsoda: A bipartite scalable framework for online disease diagnosis. In *Proceedings of the ACM Web Conference 2022*, WWW '22, pp. 2511–2521, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450390965. doi: 10.1145/3485447.3512123. URL https://doi.org/10.1145/3485447.3512123.

Hill, E. D., Kashyap, P., Raffanello, E., Wang, Y., Moffitt, T. E., Caspi, A., Engelhard, M., and Posner, J. Prediction of mental health risk in adolescents. *Nature Medicine*, pp. 1–7, 2025.

Hoffer, E., Ben-Nun, T., Hubara, I., Giladi, N., Hoefler, T., and Soudry, D. Augment your batch: Improving generalization through instance repetition. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8126–8135, 2020. doi: 10.1109/CVPR42600.2020.00815.

Hopkins, M., Reeber, E., Forman, G., , and Suermondt, J. Spambase. UCI Machine Learning Repository, 1999. DOI: https://doi.org/10.24432/C53G6X.

Howard, J. Imagenette: A smaller subset of 10 easily classified classes from imagenet, March 2019. URL https://github.com/fastai/imagenette.

Janisch, J., Pevný, T., and Lisý, V. Classification with costly features using deep reinforcement learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):3959–3966, Jul. 2019. doi: 10.1609/aaai.v33i01.33013959. URL https://ojs.aaai.org/index.php/AAAI/article/view/4287.

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 Banerjee, A. and Fukumizu, K. (eds.), *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pp. 1459–1467. PMLR, 13–15 Apr 2021.

Jethani, N., Sudarshan, M., Covert, I. C., Lee, S.-I., and Ranganath, R. FastSHAP: Real-time shapley value estimation. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=Zq2G_VTV53T.

Kachuee, M., Goldstein, O., Kärkkäinen, K., and Sarrafzadeh, M. Opportunistic learning: Budgeted cost-sensitive learning from data streams. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=S1eOHo09KX.

Kharoua, R. E. Chronic kidney disease dataset, 2024. URL https://www.kaggle.com/dsv/8658224.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv*, 2014.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes, 2013. URL https://arxiv.org/abs/1312.6114.

Krizhevsky, A. Learning multiple layers of features from tiny images. pp. 32–33, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

Kumar, I. E., Venkatasubramanian, S., Scheidegger, C., and Friedler, S. Problems with shapley-value-based explanations as feature importance measures. In *International conference on machine learning*, pp. 5491–5500. PMLR, 2020.

Li, Y. and Oliva, J. Active feature acquisition with generative surrogate models. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 6450–6459. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/li21p.html.

Loshchilov, I. and Hutter, F. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017.

Lundberg, S. M. and Lee, S.-I. A unified approach to interpreting model predictions. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Lundberg, S. M., Erion, G., Chen, H., DeGrave, A., Prutkin, J. M., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., and Lee, S.-I. From local explanations to global understanding with explainable ai for trees. *Nature machine intelligence*, 2(1):56–67, 2020.

Mesinovic, M., Watkinson, P., and Zhu, T. Explainable ai for clinical risk prediction: a survey of concepts, methods, and modalities. *arXiv*, 2023.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540): 529–533, 2015.

Nan, F. and Saligrama, V. Adaptive classification for prediction under a budget. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/d9ff90f4000eacd3a6c9cb27f78994cf-Paper.pdf.

Pereira, B., Chin, S.-F., Rueda, O. M., Vollan, H.-K. M., Provenzano, E., Bardwell, H. A., Pugh, M., Jones, L., Russell, R., Sammut, S.-J., Tsui, D. W. Y., Liu, B., Dawson, S.-J., Abraham, J., Northen, H., Peden, J. F., Mukherjee, A., Turashvili, G., Green, A. R., McKinney, S., Oloumi, A., Shah, S., Rosenfeld, N., Murphy, L., Bentley, D. R., Ellis, I. O., Purushotham, A., Pinder, S. E., Børresen-Dale, A.-L., Earl, H. M., Pharoah, P. D., Ross, M. T., Aparicio, S., and Caldas, C. Erratum: The somatic mutation profiles of 2,433 breast cancers refine their genomic and transcriptomic landscapes. *Nat. Commun.*, 7 (1):11908, June 2016.

Petsiuk, V., Das, A., and Saenko, K. Rise: Randomized input sampling for explanation of black-box models, 2018. URL https://arxiv.org/abs/1806.07421.

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., and Gulin, A. Catboost: unbiased boosting with categorical features. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/14491b756b3a51daac41c24863285549-Paper.pdf.

Radford, A. Improving language understanding by generative pre-training. *Preprint*, 2018.

Rangrej, S. B. and Clark, J. J. A probabilistic hard attention model for sequentially observed scenes. In *32nd British Machine Vision Conference 2021, BMVC 2021, Online, November 22-25, 2021*, pp. 137. BMVA Press, 2021. URL https://www.bmvc2021-virtualconference.com/assets/papers/0251.pdf.

Ribeiro, M. T., Singh, S., and Guestrin, C. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pp. 1135–1144, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450342322. doi: 10.1145/2939672.2939778. URL https://doi.org/10.1145/2939672.2939778.

Samek, W., Montavon, G., Lapuschkin, S., Anders, C. J., and Müller, K.-R. Explaining deep neural networks and beyond: A review of methods and applications. *Proceedings of the IEEE*, 109(3):247–278, 2021. doi: 10.1109/JPROC.2021.3060483.

Shapley, L. S. *17. A Value for n-Person Games*, pp. 307–318. Princeton University Press, Princeton, 1953. ISBN 9781400881970. doi: doi:10.1515/9781400881970-018. URL https://doi.org/10.1515/9781400881970-018.

Shim, H., Hwang, S. J., and Yang, E. Joint active feature acquisition and classification with variable-size set encoding. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/e5841df2166dd424a57127423d276bbe-Paper.pdf.

Štrumbelj, E. and Kononenko, I. An efficient explanation of individual classifications using game theory. *Journal of Machine Learning Research*, 11(1):1–18, 2010. URL http://jmlr.org/papers/v11/strumbelj10a.html.

Torralba, A., Fergus, R., and Freeman, W. T. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11):1958–1970, 2008. doi: 10.1109/TPAMI.2008.128.

Trapeznikov, K. and Saligrama, V. Supervised sequential classification under budget constraints. In Carvalho, C. M. and Ravikumar, P. (eds.), *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics*, volume 31 of *Proceedings of Machine Learning Research*, pp. 581–589, Scottsdale, Arizona, USA, 29 Apr–01 May 2013. PMLR. URL https://proceedings.mlr.press/v31/trapeznikov13a.html.

Valancius, M., Lennon, M., and Oliva, J. Acquisition conditioned oracle for nongreedy active feature acquisition. In Salakhutdinov, R., Kolter, Z., Heller, K., Weller, A., Oliver, N., Scarlett, J., and Berkenkamp, F. (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 48957–48975. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/valancius24a.html.

van Hasselt, H., Guez, A., and Silver, D. Deep reinforcement learning with double q-learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 30(1), Mar. 2016. doi: 10.1609/aaai.v30i1.10295. URL https://ojs.aaai.org/index.php/AAAI/article/view/10295.

von Kleist, H., Zamanian, A., Shpitser, I., and Ahmidi, N. Evaluation of active feature acquisition methods for static feature settings, 2023. URL https://arxiv.org/abs/2312.03619.

Xue, C., Kowshik, S. S., Lteif, D., Puducheri, S., Jasodanand, V. H., Zhou, O. T., Walia, A. S., Guney, O. B., Zhang, J. D., Pham, S. T., et al. Ai-based differential diagnosis of dementia etiologies on multimodal data. *Nature Medicine*, 30(10):2977–2989, 2024.

Yang, J., Shi, R., and Ni, B. Medmnist classification decathlon: A lightweight automl benchmark for medical image analysis. In *IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, pp. 191–195, 2021.

Yang, J., Shi, R., Wei, D., Liu, Z., Zhao, L., Ke, B., Pfister, H., and Ni, B. Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification. *Scientific Data*, 10(1):41, 2023.

Yin, H., Li, Y., Pan, S., Zhang, C., and Tschiatschek, S. Reinforcement learning with efficient active feature acquisition. In *Learning Meets Combinatorial Algorithms at NeurIPS2020*, 2020.

Yoon, J., Jordon, J., and van der Schaar, M. INVASE: Instance-wise variable selection using neural networks. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=BJg_roAcK7.

Zubek, V. B. and Dietterich, T. G. Pruning improves heuristic search for cost-sensitive learning. In *Proceedings of the Nineteenth International Conference on Machine Learning*, ICML '02, pp. 19–26, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc. ISBN 1558608737.

A. Appendix

A.1. Dataset descriptions

We utilized several datasets in our experiments (Table 1), including ImageNette, CIFAR-10, CIFAR-100, BloodMNIST, and Spambase. ImageNette (Howard, 2019) is a 10-class subset of the ImageNet dataset (Deng et al., 2009). CIFAR-10 and CIFAR-100 (Krizhevsky, 2009) are subsets of the 80 million tiny images dataset (Torralba et al., 2008), containing 10 and 100 classes respectively. BloodMNIST (Acevedo et al., 2020), derived from the MedMNIST dataset (Yang et al., 2021; 2023), comprises images of individual normal cells collected from individuals without infection, hematologic or oncologic diseases, and free of any pharmacologic treatment at the time of blood collection. The patch sizes are 16×16 for ImageNette (makes total of 196 patches, $d = 196$), 4×4 for the CIFAR-10 and CIFAR-100 datasets ($d = 64$), and 2×2 for the BloodMNIST dataset ($d = 196$). Spambase (Hopkins et al., 1999) is a well-known tabular dataset for classifying spam emails, consisting of 57 features derived from textual data. Additionally, to assess the applicability of our method in real-world scenarios, such as healthcare, we conducted experiments on four medical tabular datasets. As part of the preprocessing, we removed ID columns and categorical columns that were not ranking-based or binary. Columns with more than 10% missing values were also excluded, while the remaining missing values were imputed with the mean. In the following and in Table 1, the number of features refers to the count after preprocessing. The Metabric dataset (Curtis et al., 2012; Pereira et al., 2016) contains targeted gene sequencing data from 1,898 breast cancer samples, where we utilized mRNA-level Z-scores, which contains 489 features, to predict the Pam50 gene status that is a multi-class classification task. The cirrhosis patient survival (CPS) dataset (Dickson et al., 1989) includes records from 418 patients, primarily with primary biliary cirrhosis, along with 8 clinical features, with the task of predicting patient survival states categorized as death, censored, or censored due to liver transplantation. The AIDS clinical trials group study 175 (CTGS) dataset (Hammer et al., 1996) contains 2139 records of patients diagnosed with AIDS, 23 features, with a binary classification task to predict whether a patient has died within a specified time period. Lastly, the chronic kidney disease (CKD) dataset (Kharoua, 2024) comprises 1659 patient records with 50 clinical features, and the task is to predict whether a patient is diagnosed with chronic kidney disease in a binary classification setting.

A.2. Pseudocodes

Below, we provide the pseudocode for our first and second training stages, as well as for the inference stage.

Algorithm 1 Pseudocode for the first-stage training of q_π and f_θ

```
Require: Training set $\{(\mathbf{x}^i, y^i, \varphi^i)\}_{i=1}^N$, batch size $N_b$, context length $\ell$, learning rate $\gamma$
Pre-train $f_\theta$ on $\{(\mathbf{x}^i, y^i)\}_{i=1}^N$ using random feature selection
Initialize $q_\pi$
for each epoch do
    for $j = 1$ to $\lceil N/N_b \rceil$ do
        Sample minibatch $\{(\mathbf{x}^i, y^i, \varphi^i)\}_{i=1}^{N_b}$ (recalculate $\varphi^i$ for each iteration if random augmentation is applied)
        Sample random integer $t_i$ for each $i$
        Initialize losses: $\mathcal{L}_q = 0$, $\mathcal{L}_f = 0$
        for $t_x = 0$ to $\ell - 1$ do
            Define $t'_i = t_i + t_x$
            Generate masked input: $\mathbf{x}^i_{M_{t'_i}}$, where $M_{t'_i} = \{\varphi^i(1), \ldots, \varphi^i(t'_i)\}$
            Compute $\hat{\mathbf{y}}^i_{t'_i} = \mathbf{r}^i_{t'_i} = f_\theta(\mathbf{x}^i_{M_{t'_i}})$
            Compute $\hat{\mathbf{q}}^i_{t'_i} = \sigma(q_\pi(\mathbf{x}^i_{M_{t_i:t'_i}}, a^i_{t_i:t'_i}, \mathbf{r}^i_{t_i:t'_i}) - 1\text{e}6 \times M_{t'_i})$, where $a^i_{t'_i} = \varphi^i(t'_i)$
            Update $\mathcal{L}_f \leftarrow \mathcal{L}_f - \frac{1}{N_b} \sum_{i=1}^{N_b} \log(\hat{\mathbf{y}}^i_{t'_i, y^i})$
            Update $\mathcal{L}_q \leftarrow \mathcal{L}_q - \frac{1}{N_b} \sum_{i=1}^{N_b} \log(\hat{\mathbf{q}}^i_{t'_i, \varphi^i(t'_i+1)})$
        end for
        Update parameters $\theta \leftarrow \theta - \gamma \nabla_\theta \mathcal{L}_f$, $\pi \leftarrow \pi - \gamma \nabla_\pi \mathcal{L}_q$
    end for
end for
```

Algorithm 2 Pseudocode for the second-stage training of q_π and f_θ

```
Require: Training set $\{(\mathbf{x}^i, y^i, \varphi^i)\}_{i=1}^{N}$, batch size $N_b$, context length $\ell$, learning rate $\gamma$, $f_\theta$ and $q_\pi$ from the first stage
for each epoch do
    for $j = 1$ to $\lceil N/N_b \rceil$ do
        Sample minibatch $\{(\mathbf{x}^i, y^i, \varphi^i)\}_{i=1}^{N_b}$ (recalculate $\varphi^i$ for each iteration if random augmentation is applied)
        Generate $\hat{\varphi}^i$ for each $i$
        Sample random integer $t_i$ for each $i$
        Initialize losses: $\mathcal{L}_q = 0$, $\mathcal{L}_f = 0$
        for $t_x = 0$ to $\ell - 1$ do
            Define $t'_i = t_i + t_x$
            Generate masked input: $\mathbf{x}^i_{\hat{M}_{t'_i}}$, where $\hat{M}_{t'_i} = \{\hat{\varphi}^i(1), \ldots, \hat{\varphi}^i(t'_i)\}$
            Compute $\hat{\mathbf{y}}^i_{t'_i} = \mathbf{r}^i_{t'_i} = f_\theta(\mathbf{x}^i_{\hat{M}_{t'_i}})$
            Compute $\hat{\mathbf{q}}^i_{t'_i} = \sigma(q_\pi(\mathbf{x}^i_{\hat{M}_{t_i:t'_i}}, a^i_{t_i:t'_i}, \mathbf{r}^i_{t_i:t'_i}) - 1\text{e}6 \times \hat{M}_{t'_i})$, where $a^i_{t'_i} = \hat{\varphi}^i(t'_i)$
            Update $\mathcal{L}_f \leftarrow \mathcal{L}_f - \frac{1}{N_b} \sum_{i=1}^{N_b} \log(\hat{\mathbf{y}}^i_{t'_i, y^i})$
            Determine the true label for the $q_\pi$ network (denote this true label as $y^i_{q_{t'_i}}$). The true label is the index of the feature, which is not acquired yet and having the highest SHAP value among the features that are not acquired
            $\mathcal{L}_q \leftarrow \mathcal{L}_q - \frac{1}{N_b} \sum_{i=1}^{N_b} \log(\hat{\mathbf{q}}^i_{t'_i, y^i_{q_{t'_i}}})$
        end for
        Update parameters $\theta \leftarrow \theta - \gamma \nabla_\theta \mathcal{L}_f$, $\pi \leftarrow \pi - \gamma \nabla_\pi \mathcal{L}_q$
    end for
end for
```

Algorithm 3 Pseudocode for the inference stage

```
Require: Input $\mathbf{x}$, context length $\ell$, $f_\theta$, $q_\pi$, and the indices of first three features for the initialization
Acquire the first three indices in their order for the first three steps
for $t = 3$ to $k$ do
    Compute $\mathbf{r}_t = f_\theta(\mathbf{x}_{\hat{M}_t})$
    Compute $\hat{\mathbf{q}}_t = \sigma(q_\pi(\mathbf{x}_{\hat{M}_{\max(1,t-\ell+1):t}}, a^i_{\max(1,t-\ell+1):t}, \mathbf{r}^i_{\max(1,t-\ell+1):t}) - 1\text{e}6 \times \hat{M}_t)$
    Compute $\hat{\varphi}^i(t) = \arg\max \hat{\mathbf{q}}_t$
    Update the mask $\hat{M}_t \cup \hat{\varphi}^i(t)$
end for
Predict $\hat{\mathbf{y}}_k = f_\theta(\mathbf{x}_{\hat{M}_k})$
```

A.3. Additional results

We present the same performance results shown in Figure 2, now including standard deviations in Figure A1. We also show example patch acquisition trajectories for both our method and the empirical oracle in Figure A2. As seen in Figure A1, our method exhibits relatively low standard deviation values, indicating robustness to fluctuations in explanation-based rankings and random weight initialization. Figure A2 further demonstrates that our method tends to acquire patches concentrated near the center of the image, focusing on informative regions aligned with the object's shape and structure. In contrast, the empirical oracle, leveraging perfect knowledge of the image, selects scattered but informative regions based on object-specific cues, as expected.

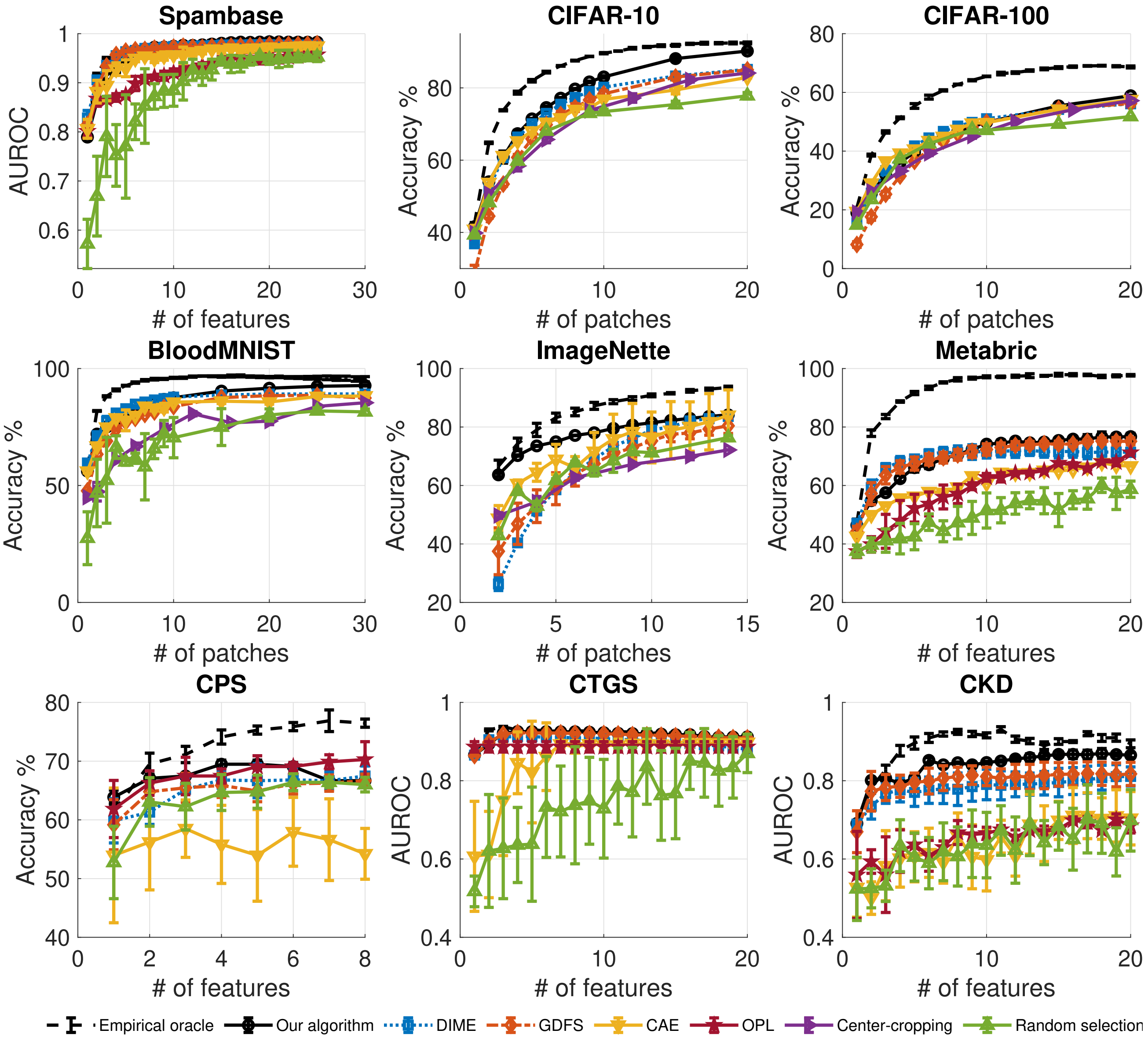

Figure A1. **Performance comparison with standard deviations.** Mean accuracy (or AUROC) and standard deviation values are shown for our method and baselines across nine datasets. Results were averaged over three independent runs for image datasets and nine independent runs for tabular datasets. Our method demonstrates consistently strong performance with relatively low variance across settings. The empirical oracle, shown in black, represents an idealized upper bound. Error bars represent standard deviations.

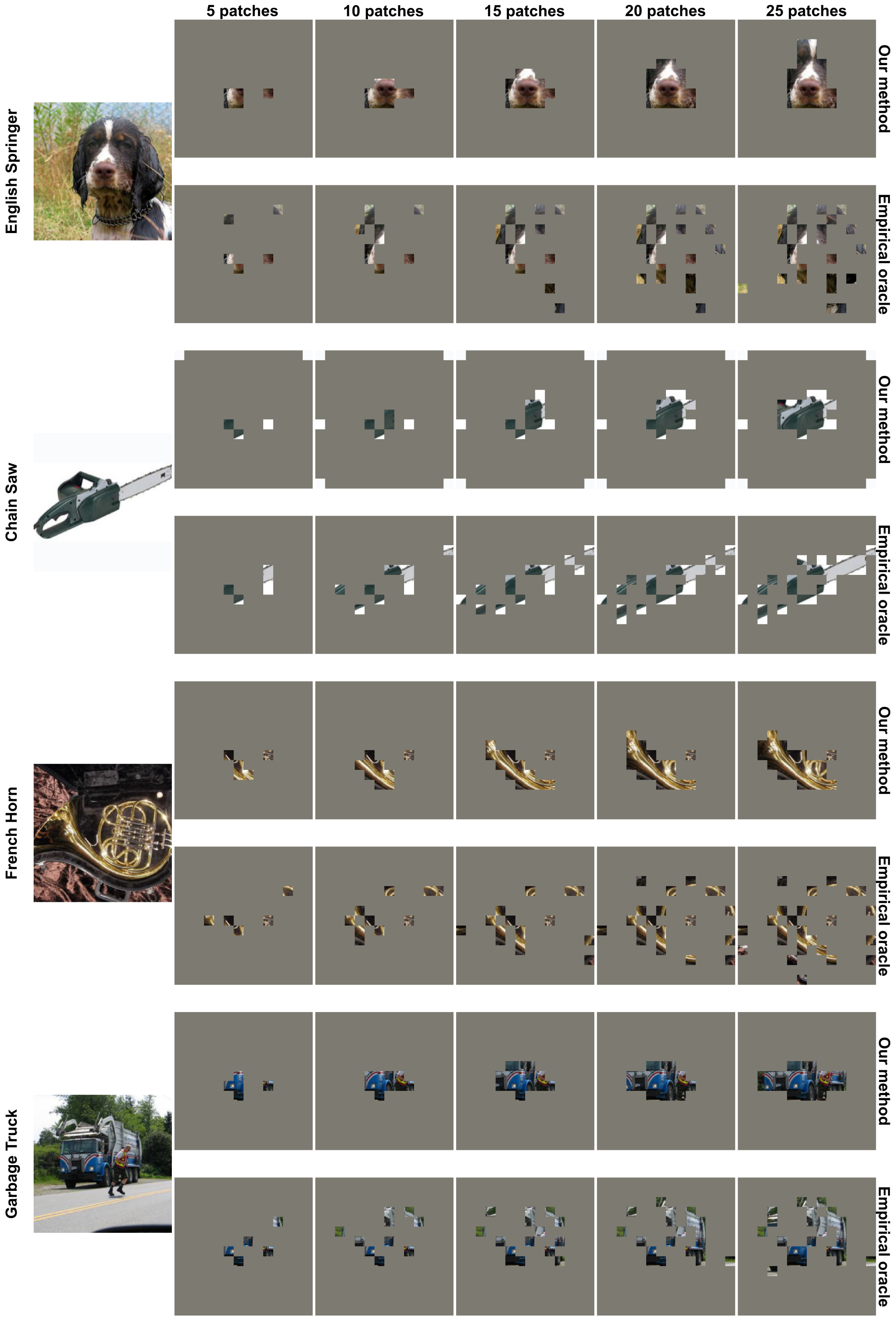

Figure A2. **Examples of feature/patch acquisition trajectories.** Illustrative examples of patch acquisition sequences for four ImageNet classes: English springer, chain saw, french horn, and garbage truck. The first column shows the original images, while the subsequent columns show the cumulative patches acquired at steps 5, 10, 15, 20, and 25. For each example, the top row corresponds to the acquisition trajectory produced by our method, and the bottom row corresponds to the empirical oracle. Our method tends to acquire patches concentrated in structurally informative regions, whereas the empirical oracle, having full image access, selects scattered but highly discriminative patches.