~~INTERPRETABILITY-DRIVEN~~ EXPLAINABILITY-DRIVEN ACTIVE FEATURE ACQUISITION IN LEARNING SYSTEMS

Anonymous authors
Paper under double-blind review

ABSTRACT

In real-world applications like medicine, machine learning models must often work with a limited number of features due to the high cost and time required to acquire all relevant data. While several static feature selection methods exist, they are suboptimal due to their inability to adapt to varying feature importance across different instances. A more flexible approach is active feature acquisition (AFA), which dynamically selects features based on their relevance for each individual case. Here, we introduce an AFA framework that leverages ~~Shapley Additive explanations~~ SHapley Additive exPlanations (SHAP) to generate instance-specific feature importance rankings. By reframing the AFA problem as a feature prediction task, we propose a policy network based on a decision transformer architecture, trained to predict the next most informative feature based on SHAP values. This method allows us to sequentially acquire features in order of their predictive significance, resulting in more efficient feature selection and acquisition. Extensive experiments across multiple datasets show that our approach achieves superior performance compared to current state-of-the-art AFA techniques, both in terms of predictive accuracy and feature acquisition efficiency. These results demonstrate the potential of ~~SHAP-based~~ explainability-driven AFA for applications where feature acquisition cost is a critical consideration~~, such as in disease diagnosis~~.

1 INTRODUCTION

In traditional machine learning settings, it is typically assumed to have all features available during inference. However, in real-world scenarios, ~~especially in medical settings,~~ acquiring these features can be expensive, time-consuming, and is often done sequentially. Therefore, it is crucial to develop methods that can make accurate predictions with a limited number of features. 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). Additionally, the chosen subset might not provide sufficient information for some cases, necessitating 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 (or dynamic) feature acquisition (AFA), which has been gaining 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. 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)~~(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 both 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 deep ~~learning model~~ learning-based local explanation methods, such as ~~Shapley Additive Explanations~~ SHapley Additive exPlanations (SHAP) (Lundberg & Lee, 2017), can be utilized to identify instance-wise feature importance rankings. With this insight, we treated the AFA problem as a feature prediction task rather than a feature exploration one. Our contributions are listed below:

- To the best of our knowledge, this is the first time in the AFA literature that the utility of local explanation methods, specifically SHAP (Lundberg & Lee, 2017), has been demonstrated for determining instance-wise feature importance rankings. We demonstrate that if we had an ideal (oracle) policy network that sequentially selects features based on their SHAP values, sorted from highest to lowest during inference, would outperform current state-of-the-art AFA techniques in terms of accuracy for any fixed number of features. Similar observations were made in the local explanation literature (Petsiuk et al., 2018; Jethani et al., 2021; 2022). They illustrate that insertion (or deletion) of the important features ranked based on their respective explanation methods improves (or degrades) model performance. However, these observations have yet to be formally compared with AFA techniques, leaving a gap in understanding how AFA methods compare to these explanation-based feature ranking approaches.
- We ~~took a different approach by training~~ trained our policy network to predict the next unacquired feature with the highest SHAP value, based on the current observation.
- We employed recently developed decision transformer (Chen et al., 2021) architecture as a policy network, and trained it using a two-stage approach. We showed that the feature importance ranking order is predictable without observing ~~them~~it. Also, our experiments demonstrate that our technique achieves better or comparable results with the state-of-the-art AFA methods on different datasets.

2 RELATED WORKS

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. These 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 (OL) 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 ~~methods~~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 the discriminative approaches such as the greedy-based methods ~~Covert et al. (2023b); Gadgil et al. (2024)~~ (Covert et al., 2023b; Gadgil et al., 2024).

In addition to AFA methods, related approaches from the budget learning literature ~~Trapeznikov & Saligrama (2013); Nan & Saligrama (2017); Ekanayake & Zois (2024)~~ (Trapeznikov & Saligrama, 2013; 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 (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)~~(Shapley, 1953), is particularly popular. However, 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. Additionally, global feature importance methods aim to identify the most relevant static features across an entire dataset. For example, the Concrete Autoencoder (CAE) (Balın et al., 2019) trains an autoencoder to select important features, while SAGE (Covert et al., 2020) extends Shapley values to quantify global feature importance through an additive importance measure. For a detailed overview, we refer readers to recent surveys (Samek et al., 2021; Bolón-Canedo et al., 2022).

Notation	**Description**
$\mathbf{x}$	Input feature vector
d	Dimensions of input feature vector
y	Target label
C	Number of classes
q_π	Policy network~~(Causal transformer)~~
f_θ	Predictor network
π	Parameters of policy network
θ	Parameters of predictor network
~~$\mathbf{r_t}$~~ $\mathbf{r}_t$	~~Index of the latest acquired feature (reward~~Logits of the predictor (action) after acquiring t features
a_t	~~Logits of the predictor (action~~Index of the latest acquired feature (reward) after acquiring t features
$\hat{\mathbf{q}}$	Output of policy network q_π
$\hat{\mathbf{y}}$	Output of predictor network f_θ
$\varphi^i(t)$	t^{th} important feature of $\mathbf{x}^{(\mathbf{i})}$ based on SHAP ranking
$\varphi^{\hat{i}}(t)$	t^{th} important feature of $\mathbf{x}^{(\mathbf{i})}$ based on policy network's predictions
M_t	Set of t most important feature indices based on SHAP ranking
$\hat{M}_t$	Set of first t feature indices acquired based on $\hat{\varphi}^i$
$\mathbf{x}_{M_t}$	Input feature vector with features from M_t unmasked
$\mathbf{x}_{\hat{M}_t}$	Input feature vector with features from $\hat{M}_t$ unmasked
ℓ	Context length of causal transformer

Table 1: Mathematical notations used in the paper.

3 PROBLEM DESCRIPTION

Let $\mathbf{x} \in \mathbb{R}^d$ represent the 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 feature j 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 the following constraint objective is minimized:

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

where the first expectation is taken over the joint distribution of $\mathbf{x}$, y, and k. The subset M is generated sequentially by the policy network q_π, which determines the next missing feature to acquire, i.e., $\arg\max q_\pi(\mathbf{x}_M) \in [d] \backslash M$. And, the predictor f_θ makes probabilistic predictions for any possible subset M, i.e., $f_\theta(\mathbf{x}_M) \in [0, 1]^{C,1}$. For brevity, let the output of q_π be denoted as $\hat{\mathbf{q}}$, i.e., $\hat{\mathbf{q}} = q_\pi(\mathbf{x}_M)$ and the output of f_θ be denoted as $\hat{\mathbf{y}}$, i.e., $\hat{\mathbf{y}} = f_\theta(\mathbf{x}_M)$.

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

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 identifying 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 paper, we approach the problem from a different perspective by assuming having access to feature importance rankings for each training sample. Consequently, instead of treating it as a feature exploration problem, we address it as a feature prediction problem (Figure 1).

4 Our Methodology

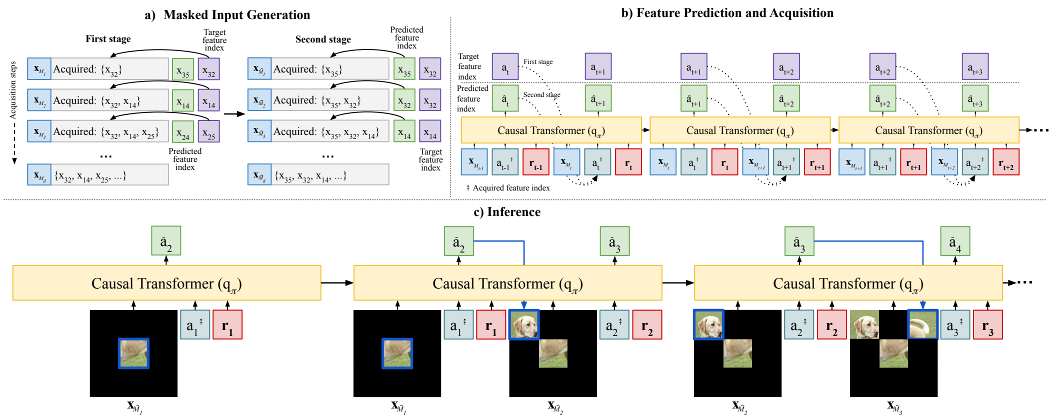

Figure 1: **Overview of our active feature acquisition framework.** a) Our training strategy consists of two stages and this figure shows how the masked inputs are generated during the first and second stages. In the first stage, features are selected based on their ranking order derived from SHAP values. 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 SHAP value 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. 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.

Feature importance ranking. In our method, we assume 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, particularly SHAP (Lundberg & Lee, 2017), can effectively achieve this goal. ~~Empirically, we observed~~ 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 observing that if the policy network perfectly acquires features in the order of highest to lowest absolute SHAP values sequentially during inference, the predictor achieves the best performance on average for a given budget of k available features, compared to the current state of the art methods (Figure 2).

To be able to get the SHAP values of the features for the each input, first, we train a classifier using $\{(\mathbf{x}^i, y^i)\}_{i=1}^N$ with the standard cross-entropy loss minimization. Then, we calculate the SHAP values

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

Table 2: **Summary of datasets used in our experiments.** For each dataset, we listed the number of features (d), number of classes, number of samples, image size, and patch size utilized. Note that Spambase is a tabular dataset; therefore, image size and patch size are not applicable.

of the features for each input $\mathbf{x}^i$ and sort them to get φ^i, where $\varphi^i(1)$ is the feature index having the highest absolute SHAP value and $\varphi^i(d)$ is the feature index having the lowest absolute SHAP value 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 the conditional sequence modeling task, similar to in the "decision transformer" framework (Chen et al., 2021), we train q_π, which is a causal transformer model, with the objective of next action/token prediction. We feed q_π with sequential data with 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 described in Chen et al. (2021). During training, at the timestep of t, the input is $\mathbf{x}^i_{M_t}$, which is the i'th sample with the 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 reinforcement learning-based methods (Kachuee et al., 2019; Li & Oliva, 2021) are typically functions of the predictor output; in our method, we follow a similar idea, but instead of defining a specific function, we directly feed our policy transformer 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 SHAP value (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 train 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 SHAP value ranking order. 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

[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

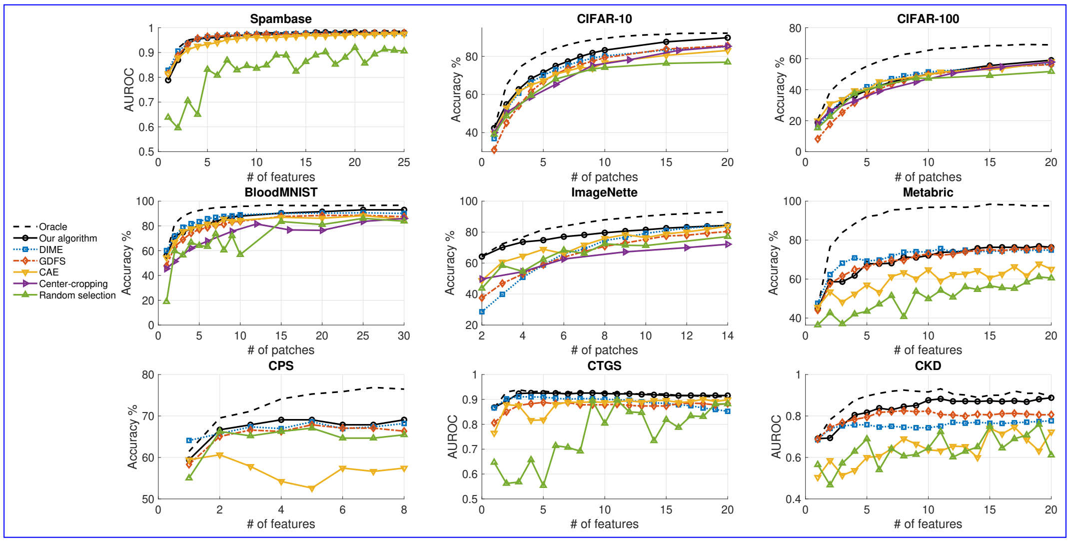

Figure 2: **Model performance.** Average classification performance of our AFA method compared with other well-known methods across varying number of features on the Spambase tabular and four image datasets: CIFAR-10, CIFAR-100, BloodMNIST and ImageNette.

highest SHAP values. To train both models to handle this new subset of features not encountered in the first stage, we introduce a second stage of training. At the beginning of each iteration of the second stage, we first generate empiric/predicted feature acquisition $\hat{\varphi}^i$ order for each $\mathbf{x}^i$, where $\hat{\varphi}^i(t+1) = \arg\max \hat{\mathbf{q}}_t^i$ and $\hat{M}_t = \{\hat{\varphi}^i(1), \hat{\varphi}^i(2), ..., \hat{\varphi}^i(t)\}$. Then, we minimize 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 SHAP value 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 train 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 SHAP value 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 prevent 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 $t = 0$, for all inputs we have to choose the same feature to be acquired. In our method, we initialized each input by the fixed first feature that has the highest SHAP value on average calculated on the training set.

Implementation details. During training, we set 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). Also, as done by 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's embedding, a simple MLP was used. In q_π, we set context length ℓ to 4, number of heads and layers 4 and 3, respectively.

	Spambase	**CIFAR-10**	**CIFAR-100**	**BloodMNIST**	**ImageNette**
# of classes:	2	10	100	8	10
First-stage	0.9512	75.68%	45.88%	79.08%	73.35%
Second-stage	0.9559	78.61%	47.06%	84.38%	79.08%

Table 3: **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, and ImageNette datasets. For the Spambase dataset, we reported the area under the receiver operating characteristic curve values, while for the remaining datasets, we provided accuracy metrics.

	CIFAR-10	**CIFAR-100**	**BloodMNIST**	**ImageNette**
# of features (d):	64	64	196	196
Top 10 features	36.54%	48.71%	42.17%	11.08%
Top 15 features	46.08%	58.30%	49.09%	15.99%
Top 20 features	52.46%	64.78%	53.58%	20.61%
Top 25 features	57.41%	68.71%	56.52%	24.92%
Top 30 features	62.00%	71.32%	58.63%	29.01%

Table 4: **Alignment between model's feature acquisition order and the SHAP-based feature importance rankings.** This table presents the percentage overlap between the top N features ranked by SHAP values and features acquired by our model for N = 10, 15, 20, 25, and 30. The datasets include CIFAR-10 and CIFAR-100 (each with 64 features), and BloodMNIST and ImageNette (each with 196 features).

5 RESULTS AND DISCUSSION

We utilized several datasets in our experiments (Table 2), including ImageNette, CIFAR-10, CIFAR-100, BloodMNIST, and Spambase. ImageNette ~~Howard (2019)~~ (Howard, 2019) is a 10-class subset of the ImageNet dataset ~~Deng et al. (2009)~~(Deng et al., 2009). CIFAR-10 and CIFAR-100 ~~Krizhevsky (2009)~~ (Krizhevsky, 2009) are subsets of the 80 Million Tiny Images dataset ~~Torralba et al. (2008)~~(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 & Suermondt, 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 2, 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 & Langworthy, 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.

	Metabric	**CPS**	**CTGS**	**CKD**
# of classes:	6	3	2	2
TreeSHAP	69.90%	67.12%	0.9167	0.8473
LIME	69.04%	66.21%	0.9125	0.8122
KernelSHAP	70.01%	66.06%	0.9146	0.8353
Sampling(IME)	69.72%	65.91%	0.9160	0.8152

Table 5: **Model performance using various feature ranking approaches.** Comparison of classification performance across four medical datasets using feature rankings derived from various local explanation methods: TreeSHAP, LIME, KernelSHAP, and Sampling (IME). The performance metrics are the area under the receiver operating characteristic curve for the binary-classification datasets and accuracy for the multi-class datasets.

To test the robustness of our method across different architectures, we also varied predictor architectures. We employed ResNet50 ~~He et al. (2016)~~ (He et al., 2016) for ImageNette, ResNet18 ~~He et al. (2016)~~ (He et al., 2016) for the CIFAR-10 and CIFAR-100 datasets, and a custom CNN for the BloodMNIST dataset. The custom CNN architecture 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 multi-layer perception (MLP) architecture 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, 512 on CKD, 512 on CTGS and 128 on CPS. For the image datasets, we employed FastSHAP (Jethani et al., 2022) to generate the feature SHAP ranking order φ^i for each instance $\mathbf{x}^i$ due to its speed. During training, we applied random augmentations that can affect feature importance. The speed of FastSHAP allows us to efficiently handle these changes in feature importance during the training process. For the ~~Spambase dataset~~tabular datasets, we did not apply any data augmentation during training. ~~We obtained the~~ Tree-based models, specifically CatBoost (Prokhorenkova et al., 2018), were used as the initial model to determine feature ranking orders, owing to their superior performance on tabular data (Grinsztajn et al., 2022). SHAP ranking orders were obtained using the SHAP package[4]~~.~~, leveraging TreeSHAP (Lundberg et al., 2020), a method specifically designed for SHAP value calculations in tree-based models

We evaluated our method against several existing approaches for feature selection: Discriminative Mutual Information Estimation (DIME), Greedy Dynamic Feature Selection (GDFS), Concrete Autoencoder (CAE), and two simple baselines~~—~~: center-cropping and random selection. DIME (Gadgil et al., 2024) takes an information-theoretic approach by prioritizing features based on their mutual information with the response variable, estimating this mutual information in a discriminative rather than a generative manner. GDFS (Covert et al., 2023b) employs a simpler, greedy strategy for selecting features based on their conditional mutual information, utilizing a learning approach grounded in amortized optimization; the policy network is shown to recover the greedy policy when trained to optimality. CAE (Balın et al., 2019) is an unsupervised, end-to-end differentiable method for global feature selection that uses a standard neural network as the decoder for reconstruction and incorporates a concrete selector layer as the encoder. The temperature in CAE is gradually decreased to slowly discretize the selections. The center-cropping and random selection methods (Covert et al., 2023b) serve as simple baselines: center-cropping selects center patches of varying sizes, while random selection chooses patches randomly.

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 64.2% and 74.8% average accuracy with two and five available patches among 196 patches, respectively. Additionally, our model achieved an AUROC score of 0.8761 on the CKD dataset. To demonstrate the relative potential of our approach, we also provided the oracle setting performances, where during inference the features are perfectly acquired based on the SHAP ranking. On the oracle setting, we also initialized the instances with the feature having the highest SHAP value on average, as in our method. Therefore, while it is theoretically possible to achieve these performance

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

levels, empirically we could not attain them with our current method. We discovered that instead of initializing the inputs with only one feature, it is more effective for stable training to begin with three features. Based on this finding, we fixed first three feature acquisition order and we obtained the results as shown in Figure 2. For the initial features, we selected the second and third features based on their average importance. 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 in 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, particularly SHAP, are effective in providing this order.

The average performance after both stages is shown for all the datasets (Table 3), highlighting the benefit of the second stage. The second stage provides significant improvement on almost all datasets, except on the Spambase that is a relatively simpler dataset compared to others, at least in terms of number of classes and features. Specifically, the second stage provides classification accuracy increase from 2.57% (on CIFAR-100) to 7.81% (on ImageNette).

~~Lastly, in Table~~ We also performed some ablation experiments to evaluate the robustness and effectiveness of our method. Firstly, we tested our method with alternative feature ranking approaches, including another explainability method, LIME (Ribeiro et al., 2016), and two different SHAP value calculation techniques: KernelSHAP (Lundberg & Lee, 2017) and IME (sampling) (Štrumbelj & Kononenko, 2010). These results (Table 5) indicate that while our method is robust to different ranking orders, its performance is also dependent on the the quality of the ranking order generated by the explainability methods. To further verify the second point and test the dependency of the SHAP 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 SHAP ranking order, while retaining ResNet-18 as the classification network. We observed that the performance of our method decreased from 78.61% 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 SHAP-based ranking order and, consequently, the performance of our method. In addition,we evaluated the effectiveness of using the decision transformer 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 78.61% to 76.83% on the CIFAR-10 dataset and from 47.06% to 46.70% on the CIFAR-100 dataset. Similarly, substituting the decision transformer with a CNN block reduced the model's accuracy from 84.38% to 78.23% on the BloodMNIST dataset. These results demonstrate the advantage of using a decision transformer as the policy network while also highlighting that our method performs reasonably well with other architectures as the policy network.

~~(Lastly, i)~~In Table 4, we present the overlap ratios between our model's acquired feature order and the SHAP-based feature importance rankings across different datasets. As the number of top features (N) increases from 10 to 30, the percentage overlap generally rises for CIFAR-10, CIFAR-100, BloodMNIST, and ImageNette. This trend indicates that our model's feature acquisition order increasingly aligns with the SHAP rankings as more features are considered. While the oracle performances in Figure 2 demonstrate the practical benefits of using SHAP values in the AFA problem, Table 4 highlights the degree to which our model's acquisition strategy predicts the SHAP-based feature importance ranking. Additionally, we would like to note that we did not perform detailed parameter search on the experiments. We selected the context length ℓ parameter by comparing the validation scores on the CIFAR-10 dataset and the other parameters were selected heuristically by hand. Subsequently, all these parameters were fixed for all the experiments. About the ℓ parameter selection, we found that our model achieved accuracies of 78.41%, 78.76%, 79.12%, and 78.12% on the CIFAR-10 validation dataset with $\ell = 1$, $\ell = 2$, $\ell = 4$, and $\ell = 8$, respectively. When we increased or decreased the context length ℓ, we correspondingly varied the batch size N_b by the same factor to maintain the same effective size at each iteration (see Equations 2 and 3). Based on these observations, we assigned $\ell = 4$. Finally, we would like to emphasize that our proposed method is flexible and can operate with any given feature order. However, due to the absence of a reference standard (or ground truth) for feature importance rankings, we relied on explainability methods to generate the feature orders. While these approaches are useful, they may not always provide good rankings in all scenarios (Kumar et al., 2020; Catav et al., 2021). As different local explanation methods that provide better rankings are developed, our approach can be readily integrated to deliver enhanced results.

~~Top 10 features 36.54% 48.71% 42.17% 11.08% Top 15 features 46.08% 58.30% 49.09% 15.99% Top 20 features 52.46% 64.78% 53.58% 20.61% Top 25 features 57.41% 68.71% 56.52% 24.92% Top 30 features 62.00% 71.32% 58.63% 29.01%~~ ~~**Alignment between model's feature acquisition order and the SHAP-based feature importance rankings.**~~ ~~This table presents the percentage overlap between the top N features ranked by SHAP values and features acquired by our model for N = 10, 15, 20, 25, and 30. The datasets include CIFAR-10 and CIFAR-100 (each with 64 features), and BloodMNIST and ImageNette (each with 196 features).~~

6 CONCLUSION

Our work ~~proposes a novel~~ introduces an explainability-based active feature acquisition strategy by reframing it as a feature prediction task, where the model learns to acquire features based on instance-specific SHAP value rankings. Stage-wise results ~~show~~ demonstrate that our two stage training approach ~~enhances~~ improves feature selection and classification performance on ~~both~~ tabular and image datasets. The findings ~~also indicate~~ further suggest that our method is robust across ~~varying~~ various models, datasets and settings. ~~Future work could apply our method to more practical datasets, such as those in medical diagnosis, to evaluate its usability in~~ Additionally, our experimental results on medical tabular datasets highlight the practical applicability of our method in real-world scenarios ~~.~~ like healthcare. Future work could explore dynamic recalculations of feature attributions during training after each feature acquisition step, replacing the current use of fixed ordering.

REFERENCES

Andrea Acevedo, Anna Merino, Santiago Alférez, Ángel Molina, Laura Boldú, and José Rodellar. A dataset of microscopic peripheral blood cell images for development of automatic recognition systems. *Data in Brief*, 30:105474, 2020.

Marco Ancona, Cengiz Oztireli, and Markus Gross. Explaining deep neural networks with a polynomial time algorithm for shapley value approximation. In Kamalika Chaudhuri and Ruslan Salakhutdinov (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.

Muhammed Fatih Balın, Abubakar Abid, and James Zou. Concrete autoencoders: Differentiable feature selection and reconstruction. In Kamalika Chaudhuri and Ruslan Salakhutdinov (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.

Verónica Bolón-Canedo, Amparo Alonso-Betanzos, Laura Morán-Fernández, and Brais Cancela. *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.

Amnon Catav, Boyang Fu, Yazeed Zoabi, Ahuva Libi Weiss Meilik, Noam Shomron, Jason Ernst, Sriram Sankararaman, and Ran Gilad-Bachrach. Marginal contribution feature importance - an axiomatic approach for explaining data. In Marina Meila and Tong Zhang (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.

Aditya Chattopadhyay, Kwan Ho Ryan Chan, Benjamin David Haeffele, Donald Geman, and Rene Vidal. Variational information pursuit for interpretable predictions. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=77lSWa-Tm3Z.

Aditya Chattopadhyay, Kwan Ho Ryan Chan, and Rene Vidal. 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.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (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.

Ian Covert, Scott M Lundberg, and Su-In Lee. Understanding global feature contributions with additive importance measures. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (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.

Ian Connick Covert, Chanwoo Kim, and Su-In Lee. 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.

Ian Connick Covert, Wei Qiu, Mingyu Lu, Na Yoon Kim, Nathan J White, and Su-In Lee. Learning to maximize mutual information for dynamic feature selection. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (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.

Christina Curtis, Sohrab P. Shah, Suet-Feung Chin, Gulisa Turashvili, Oscar M. Rueda, Mark J. Dunning, Doug Speed, Andy G. Lynch, Shamith Samarajiwa, Yinyin Yuan, Stefan Gräf, Gavin Ha, Gholamreza Haffari, Ali Bashashati, Roslin Russell, Steven McKinney, Anita Langerød, Andrew Green, Elena Provenzano, Gordon Wishart, Sarah Pinder, Peter Watson, Florian Markowetz, Leigh Murphy, Ian Ellis, Arnie Purushotham, Anne-Lise Børresen-Dale, James D. Brenton, Simon Tavaré, Carlos Caldas, and Samuel Aparicio. 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.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 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.

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

Gabriel Dulac-Arnold, Ludovic Denoyer, Philippe Preux, and Patrick Gallinari. Datum-wise classification: A sequential approach to sparsity. In Dimitrios Gunopulos, Thomas Hofmann, Donato Malerba, and Michalis Vazirgiannis (eds.), *Machine Learning and Knowledge Discovery in Databases*, pp. 375–390, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg. ISBN 978-3-642-23780-5.

Sachini Piyoni Ekanayake and Daphney–Stavroula Zois. 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.

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

Leo Grinsztajn, Edouard Oyallon, and Gael Varoquaux. 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.

Scott M. Hammer, David A. Katzenstein, Michael D. Hughes, Holly Gundacker, Robert T. Schooley, Richard H. Haubrich, W. Keith Henry, Michael M. Lederman, John P. Phair, Manette Niu, Martin S. Hirsch, and Thomas C. Merigan. 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.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Weijie He and Ting Chen. 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.

Weijie He, Xiaohao Mao, Chao Ma, Yu Huang, José Miguel Hernàndez-Lobato, and Ting Chen. 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.

Elad Hoffer, Tal Ben-Nun, Itay Hubara, Niv Giladi, Torsten Hoefler, and Daniel Soudry. 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.

Reeber Erik Forman George Hopkins, Mark and Jaap Suermondt. Spambase. UCI Machine Learning Repository, 1999. DOI: https://doi.org/10.24432/C53G6X.

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

Jaromír Janisch, Tomáš Pevný, and Viliam Lisý. 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.

Neil Jethani, Mukund Sudarshan, Yindalon Aphinyanaphongs, and Rajesh Ranganath. Have we learned to explain?: How interpretability methods can learn to encode predictions in their interpretations. In Arindam Banerjee and Kenji Fukumizu (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.

Neil Jethani, Mukund Sudarshan, Ian Connick Covert, Su-In Lee, and Rajesh Ranganath. FastSHAP: Real-time shapley value estimation. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=Zq2G_VTV53T.

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

Rabie El Kharoua. chronic kidney disease dataset , 2024. URL https://www.kaggle.com/dsv/8658224.

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

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

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

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

Yang Li and Junier Oliva. Active feature acquisition with generative surrogate models. In Marina Meila and Tong Zhang (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.

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

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

Scott M Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. From local explanations to global understanding with explainable ai for trees. *Nature machine intelligence*, 2(1):56–67, 2020.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

Feng Nan and Venkatesh Saligrama. Adaptive classification for prediction under a budget. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (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.

Bernard Pereira, Suet-Feung Chin, Oscar M Rueda, Hans-Kristian Moen Vollan, Elena Provenzano, Helen A Bardwell, Michelle Pugh, Linda Jones, Roslin Russell, Stephen-John Sammut, Dana W Y Tsui, Bin Liu, Sarah-Jane Dawson, Jean Abraham, Helen Northen, John F Peden, Abhik Mukherjee, Gulisa Turashvili, Andrew R Green, Steve McKinney, Arusha Oloumi, Sohrab Shah, Nitzan Rosenfeld, Leigh Murphy, David R Bentley, Ian O Ellis, Arnie Purushotham, Sarah E Pinder, Anne-Lise Børresen-Dale, Helena M Earl, Paul D Pharoah, Mark T Ross, Samuel Aparicio, and Carlos Caldas. Erratum: The somatic mutation profiles of 2,433 breast cancers refine their genomic and transcriptomic landscapes. *Nat. Commun.*, 7(1):11908, June 2016.

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

Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. Catboost: unbiased boosting with categorical features. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (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.

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

Samrudhdhi B. Rangrej and James J. Clark. 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.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "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.

Wojciech Samek, Grégoire Montavon, Sebastian Lapuschkin, Christopher J. Anders, and Klaus-Robert Müller. 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.

L. S. Shapley. *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`.

Hajin Shim, Sung Ju Hwang, and Eunho Yang. Joint active feature acquisition and classification with variable-size set encoding. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (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`.

Erik Štrumbelj and Igor Kononenko. 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`.

Antonio Torralba, Rob Fergus, and William T. Freeman. 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.

Kirill Trapeznikov and Venkatesh Saligrama. Supervised sequential classification under budget constraints. In Carlos M. Carvalho and Pradeep Ravikumar (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`.

Hado van Hasselt, Arthur Guez, and David Silver. 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`.

Henrik von Kleist, Alireza Zamanian, Ilya Shpitser, and Narges Ahmidi. Evaluation of active feature acquisition methods for static feature settings, 2023. URL `https://arxiv.org/abs/2312.03619`.

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

Jiancheng Yang, Rui Shi, Donglai Wei, Zequan Liu, Lin Zhao, Bilian Ke, Hanspeter Pfister, and Bingbing Ni. Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification. *Scientific Data*, 10(1):41, 2023.

Haiyan Yin, Yingzhen Li, Sinno Pan, Cheng Zhang, and Sebastian Tschiatschek. Reinforcement learning with efficient active feature acquisition. In *Learning Meets Combinatorial Algorithms at NeurIPS2020*, 2020.

Valentina Bayer Zubek and Thomas G. Dietterich. 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

Below we provide pseudocodes for our first and second training stages.

Algorithm 1 Pseudocode for 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 ℓ, learning rate γ

Pre-train f_θ on $\{(\mathbf{x}^i, y^i)\}_{i=1}^N$ using random feature selection
Initialize q_π
for each epoch **do**
  for 1 to $\lceil N/N_b \rceil$ **do**
    Sample minibatch $\{(\mathbf{x}^i, y^i, \varphi^i)\}_{i=1}^{N_b}$ (if random augmentation is applied, φ^i is recalculated for each iteration after sampling)
    Sample random integer t_i for each i
    Initialize $\mathcal{L}_q = 0$ and $\mathcal{L}_f = 0$
    for $t_x = 0$ to $\ell - 1$ **do**
      Define temporary parameter t'_i for each i, $t'_i = t_i + t_x$
      Generate masked input $\mathbf{x}^i_{M_{t'_i}}$, $M_{t'_i} = \{\varphi^i(1), \ldots, \varphi^i(t'_i)\}$
      Compute predictor output: $\hat{\mathbf{y}}^i_{t'_i} = f_\theta(\mathbf{x}^i_{M_{t'_i}})$
      Compute policy network output: $\hat{\mathbf{q}}^i_{t'_i} = q_\pi(\mathbf{x}^i_{M_{t:t'_i}}, a^i_{t:t'_i}, \mathbf{r}^i_{t:t'_i})$
      Update losses: $\mathcal{L}_f \leftarrow -\frac{1}{N_b}\sum_{i=1}^{N_b} \log(\hat{\mathbf{y}}^i_{t'_i, y^i}) + \mathcal{L}_f$
      $\mathcal{L}_q \leftarrow -\frac{1}{N_b}\sum_{i=1}^{N_b} \log(\hat{\mathbf{q}}^i_{t'_i, \varphi^i(t'_i+1)}) + \mathcal{L}_q$
    Update parameters $\theta \leftarrow \theta - \gamma \nabla_\theta \mathcal{L}_f, \quad \pi \leftarrow \pi - \gamma \nabla_\pi \mathcal{L}_q$

Algorithm 2 Pseudocode for 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 ℓ, learning rate γ, f_θ and q_π from the first stage

for each epoch **do**
  for 1 to $\lceil N/N_b \rceil$ **do**
    Sample minibatch $\{(\mathbf{x}^i, y^i, \varphi^i)\}_{i=1}^{N_b}$ (if random augmentation is applied, φ^i is recalculated for each iteration after sampling)
    Generate $\hat{\varphi}^i$ for each i
    Sample random integer t_i for each i
    Initialize $\mathcal{L}_q = 0$ and $\mathcal{L}_f = 0$
    for $t_x = 0$ to $\ell - 1$ **do**
      Define temporary parameter t'_i for each i, $t'_i = t_i + t_x$
      Generate masked input $\mathbf{x}^i_{\hat{M}_{t'_i}}$, $\hat{M}_{t'_i} = \{\hat{\varphi}^i(1), \ldots, \hat{\varphi}^i(t'_i)\}$
      Compute predictor output: $\hat{\mathbf{y}}^i_{t'_i} = f_\theta(\mathbf{x}^i_{\hat{M}_{t'_i}})$
      Compute policy network output: $\hat{\mathbf{q}}^i_{t'_i} = q_\pi(\mathbf{x}^i_{\hat{M}_{t:t'_i}}, a^i_{t:t'_i}, \mathbf{r}^i_{t:t'_i})$
      Update losses: $\mathcal{L}_f \leftarrow -\frac{1}{N_b}\sum_{i=1}^{N_b} \log(\hat{\mathbf{y}}^i_{t'_i, y^i}) + \mathcal{L}_f$
      Determine the true label for the q_π 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
      $\mathcal{L}_q \leftarrow -\frac{1}{N_b}\sum_{i=1}^{N_b} \log(\hat{\mathbf{q}}^i_{t'_i, y^i_{q_{t'_i}}}) + \mathcal{L}_q$
    Update parameters $\theta \leftarrow \theta - \gamma \nabla_\theta \mathcal{L}_f, \quad \pi \leftarrow \pi - \gamma \nabla_\pi \mathcal{L}_q$