Active Feature Acquisition Via Explainability-driven Ranking

Osman Berke Guney, Ketan Suhaas Saichandran, Karim Elzokm, Ziming Zhang, Vijaya B. Kolachalama

September 19, 2025

Motivation Background

Problem Statement

- Real-world feature acquisition is often costly, time-consuming, and sequential.
- Developing models that can make accurate predictions while minimizing feature acquisition.

Static vs. Active Feature Acquisition (AFA)

- A static global subset is suboptimal since it ignores instance variability.
- AFA can identify important features sequentially for each individual instance.

AFA Approaches

- **Rl-Based AFA:** A policy network for feature acquisition and a prediction network for prediction with the available subset of features.
- **Greedy-Based AFA:** Estimating the conditional mutual information (CMI) of unacquired features given the current available subset of features.
- **Imputation-Based AFA:** Imputing missing features from nearest neighbors and selecting the next feature based on the ensemble.

Proposed Methods

- Utilize local explanation methods (e.g., SHAP) to identify instance-wise feature importance rankings.
- Reframe AFA as a feature prediction task to select the next unacquired feature with the highest importance ranking based on current observations.
- Employ a decision transformer architecture as the policy network and train it using a two-stage approach.

Problem Formulation

- d-dimensional input feature vector $\mathbf{x} \in \mathbb{R}^d$;
- The associated target label $y \in \{1, 2, ..., C\}$;
- Subset of acquired feature indices $M \subseteq \{1, ..., d\}$;
- Masked input vector $\mathbf{x}_M$;
- Feature cost c_j for j-th feature and budget constraint k.

Objective: Finding a predictor f_θ, and a policy network q_π, such that the constraint objective is minimized:

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

Oracle Policy Network (Upper Bound)

- Assume the policy has access to the true importance ranking for each instance;
- Assume the policy has perfect knowledge of the optimal subset that satisfies the budget constraint;
- Oracle policy network q^* sequentially selects the features in the optimal subset ordered by their importance ranking.

Feature importance ranking

- Step 1: Train a classifier using $\{\mathbf{x}^i, y^i\}_{i=1}^N$;
- Step 2: Run an explanation method to get feature importance ranking order ϕ^i;
- Step 3: Use training set $\{\mathbf{x}^i, y^i, \phi^i\}_{i=1}^N$ to train policy network and predictor network.

Policy Network: Decision Transformer

- Input token: $\mathbf{x}^i_{M_t}$
- Action token: $a^i_t = \phi^i(t)$;
- Reward token: $r^i_j = \hat{\mathbf{y}}^i_t = f_\theta(\mathbf{x}^i_{M_t})$

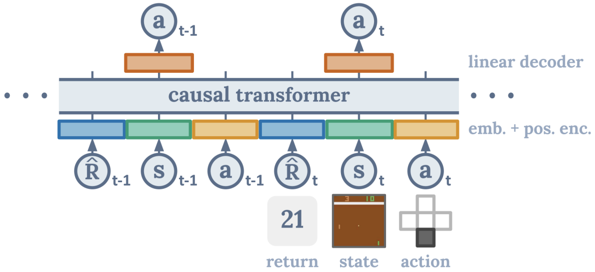


Figure: Decision Transformer architecture

Policy Network

Fed q_π with sequential data and a sequence length l. For a given sequence from the timestep t to $t+l-1$:

$$\hat{\mathbf{q}}_t^i = q_\pi(\mathbf{x}_{M_t}^i, \mathbf{a}_t^i, \mathbf{r}_t^i),$$

$$\hat{\mathbf{q}}_{t+\ell-1}^i = q_\pi(\mathbf{x}_{M_{t:t+\ell-1}}^i, \mathbf{a}_{t:t+\ell-1}^i, \mathbf{r}_{t:t+\ell-1}^i).$$

Policy network q_π and predictor network f_θ are trained simultaneously using standard cross-entropy loss:

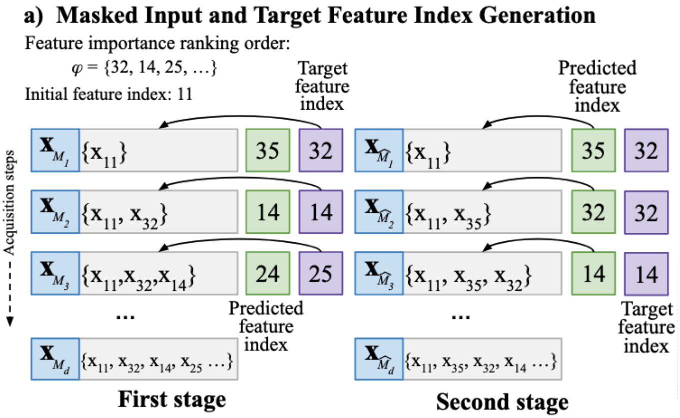
$$L_q = -\frac{1}{N_b}\sum_{i=1}^{N_b}\sum_{t=t_i}^{t_i+\ell-1}\log(\hat{\mathbf{q}}_{t,\varphi^i(t+1)}^i),$$

$$L_f = -\frac{1}{N_b}\sum_{i=1}^{N_b}\sum_{t=t_i}^{t_i+\ell-1}\log(\hat{\mathbf{y}}_{t,y}^i).$$

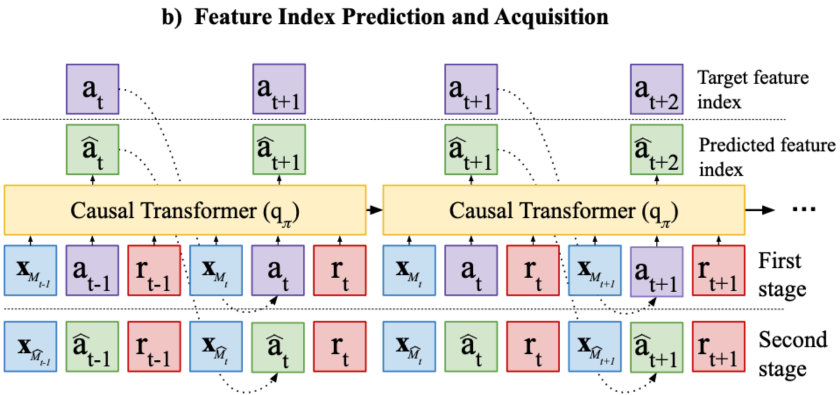
Two-Stage Training

First stage:
Initialize with the feature with the highest average importance ranking.
Train f_θ and q_π to imitate the feature importance ranking ϕ and subset M made by local explanation methods.
Second stage:
Refine models to handle imperfect feature subsets $\hat{M}_t$ based on the predicted ranking $\hat{\phi}$ from learned policy.

Two-Stage Training



Two-Stage Training



Implementation Details

- Pre-train predictor network f_θ;
- Share the backbone between f_θ and q_π;
- Use the shared backbone for input token embeddings;
- Use a learnable embedding dictionary for action token embeddings
- Use a linear layer followed by a non-linear activation for reward token embeddings.
- Subtract a large constant from q_π's logit prediction on already acquired features.

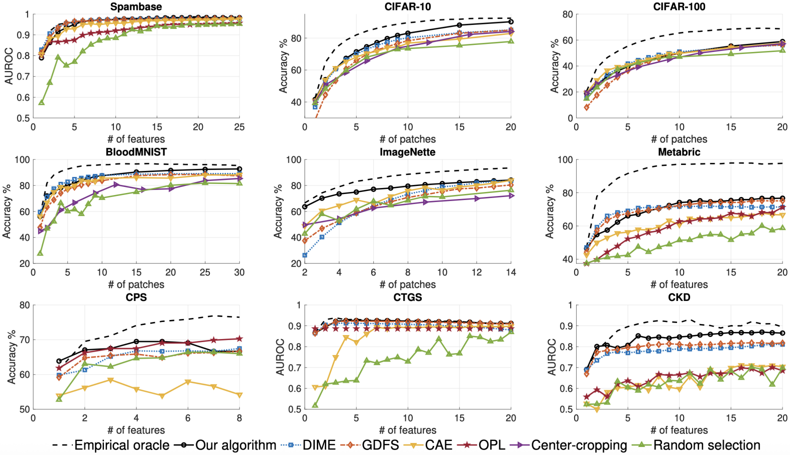
Comparison with other methods

- Global importance: CAE (concrete autoencoder)
- Greedy-based AFA: DIME (discriminative mutual information estimation), GDFS (greedy dynamic feature selection)
- RL-based AFA: OPL (opportunistic learning);
- Nearest neighbor-based: AACO;
- Baseline: center-cropping and random selection;
- Empirical oracle: optimal feature acquisition order for each instance based on importance ranking;

Datasets

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	-

Results


Spambase
AUROC
# of features
CIFAR-10
Accuracy %
# of patches
CIFAR-100
Accuracy %
# of patches
BloodMNIST
Accuracy %
# of patches
ImageNette
Accuracy %
# of patches
Metabric
Accuracy %
# of features
CPS
Accuracy %
# of features
CTGS
AUROC
# of features
CKD
AUROC
# of features
Empirical oracle
Our algorithm
DIME
GDFS
CAE
OPL
Center-cropping
Random selection

Results

	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	$0.95_{\pm 0.005}$	$68.1_{\pm 0.75}\%$	$67.2_{\pm 0.22}\%$	$0.91_{\pm 0.009}$	$0.83_{\pm 0.003}$

Stage-wise results

	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	$0.955_{\pm.0001}$	$78.44_{\pm0.15}\%$	$46.99_{\pm0.15}\%$	$83.87_{\pm1.05}\%$	$78.96_{\pm0.12}\%$	$69.83_{\pm0.41}\%$	$67.45_{\pm0.13}\%$	$0.916_{\pm.0001}$	$0.836_{\pm.07}$

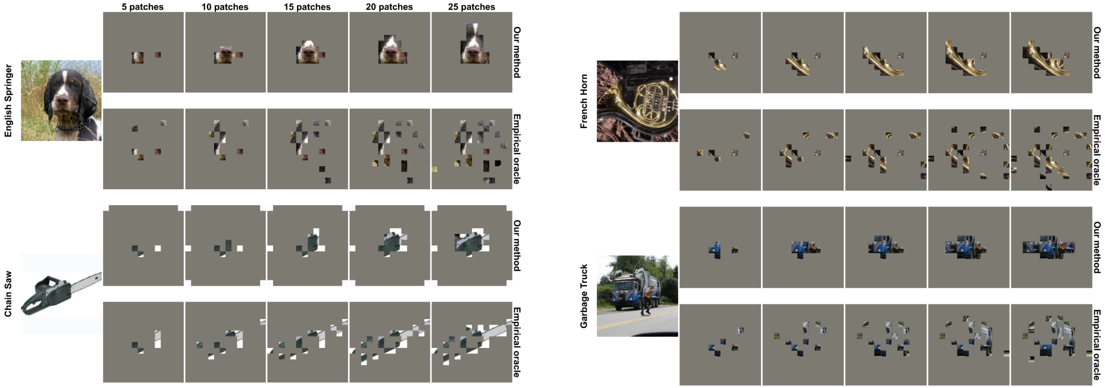
Results with different feature ranking approaches

	Spam	Metabric	CPS	CTGS	CKD
# of classes:	2	6	3	2	2
T-SHAP	$0.96_{\pm 0.001}$	$69.8_{\pm 0.41}\%$	$67.5_{\pm 0.13}\%$	$0.92_{\pm 0.001}$	$0.84_{\pm 0.07}$
LIME	$0.95_{\pm 0.002}$	$69.2_{\pm 0.18}\%$	$67.1_{\pm 0.36}\%$	$0.91_{\pm 0.001}$	$0.82_{\pm 0.09}$
K-SHAP	$0.96_{\pm 0.002}$	$69.6_{\pm 0.33}\%$	$67.3_{\pm 0.56}\%$	$0.92_{\pm 0.001}$	$0.83_{\pm 0.005}$
IME	$0.95_{\pm 0.001}$	$69.8_{\pm 0.10}\%$	$67.1_{\pm 0.61}\%$	$0.92_{\pm 0.001}$	$0.83_{\pm 0.1}$
INVASE	$0.93_{\pm 0.002}$	-	$68.4_{\pm 0.23}\%$	$0.91_{\pm 0.003}$	$0.83_{\pm 0.09}$

Alignment between model prediction and importance rankings

# of features (d):	**Spam** 57	**CIFAR-10** 64	**CIFAR-100** 64	**BloodMNIST** 196	**ImageNette** 196	**Metabric** 489	**CKD** 50	**CTGS** 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

Examples of feature acquisition trajectories



Conclusions

- The proposed method outperforms or matches SOTA AFA approaches.
- The superior performance of the empirical oracle highlights that instance-specific feature importance rankings derived from local explanation methods are effective for the AFA tasks.
- Two-stage training strategy is effective.
- The proposed method is robust across various models, datasets, and settings, showing strong applicability in real-world scenarios.

Discussions

- The flexibility of the proposed method can operate with any given feature ordering, including those provided by humans.
- The potential improvement from more accurate explanation techniques.
- Highlight the practical use in medicine, where pretrained AFA are customized for specific conditions and explainability tools are used to enhance interpretability and build trust in AI applications.

Limitations

- The proposed method is computationally expensive.
- The feature acquisition costs are simplified to be uniform.