Fast Explanations via Policy Gradient-Optimized Explainer

Deng Pan , Nuno Moniz , Nitesh V. Chawla
Lucy Family Institute for Data & Society, University of Notre Dame
Notre Dame, IN 46556 USA
{dpan, nuno.moniz, nchawla}@nd.edu

Abstract

The challenge of delivering efficient explanations is a critical barrier that prevents the adoption of model explanations in real-world applications. Existing approaches often depend on extensive model queries for sample-level explanations or rely on expert's knowledge of specific model structures that trade general applicability for efficiency. To address these limitations, this paper introduces a novel framework Fast EXplanation (FEX) that represents attribution-based explanations via probability distributions, which are optimized by leveraging the policy gradient method. The proposed framework offers a robust, scalable solution for real-time, large-scale model explanations, bridging the gap between efficiency and applicability. We validate our framework on image and text classification tasks and the experiments demonstrate that our method reduces inference time by over 97 percent and memory usage by 70 percent compared to traditional model-agnostic approaches while maintaining high-quality explanations and broad applicability.

1 Introduction

While deep classification models demonstrate superior performance across a range of tasks, their "black-box" nature often hinders their acceptance and deployment in critical areas such as healthcare, finance, and autonomous systems [Miotto *et al.*, 2018; Ozbayoglu *et al.*, 2020; Grigorescu *et al.*, 2020]. In these high-stakes contexts, it is essential not only to achieve high predictive accuracy but also to provide clear, understandable explanations of the models' decisions to foster trust and ensure accountability.

Despite the progress in explainable AI (XAI) research [Atakishiyev *et al.*, 2021; Singh *et al.*, 2020], achieving explainability in real-world, large-scale applications remains challenging. A significant barrier is the difficulty of providing *efficient explanations* that can scale without imposing prohibitive computational costs. Current attribution-based explanation methods often require substantial computation during inference, making them impractical for time-sensitive tasks and large-scale deployment [Chuang *et al.*, 2023a; Lundberg *et al.*, 2020]. Therefore, improving the efficiency of explanations is critical for enabling their broader adoption in real-world applications.

In practice, when working with black-box models or complex architectures, it is expected to use *model-agnostic explanation methods* [Ribeiro *et al.*, 2016; Lundberg and Lee, 2017; Petsiuk *et al.*, 2018; Fong and Vedaldi, 2017]. These methods have the advantage of being applicable to a wide range of models, but they often require numerous additional forward passes or gradient computations, making them inefficient and costly for real-world applications.

In scenarios where we have full access to the model's architecture, such as CNNs or Transformers, *model-specific explanation methods* can be employed to provide rapid explanations [Selvaraju *et al.*, 2017; Chefer *et al.*, 2021; Qiang *et al.*, 2022]. These methods are tailored to specific model structures, leveraging the unique behaviors of certain architectures to achieve efficient explanations. However, in real-world settings, models are often either black-box or not easily categorized into standard architectures, which limits the application of model-specific explanations.

To address the inefficiency of model-agnostic methods and the limitation of model-specific approaches, *amortized explanation* techniques have been proposed by training a deep neural network (DNN) to approximate an explanation distribution, thereby accelerating model-agnostic explanations to a single forward pass during inference [Chuang *et al.*, 2023a; Jethani *et al.*, 2021; Chen *et al.*, 2018]. However, these methods rely heavily on approximating specific proxy explanation methods, such as SHAP, by treating their explanations as ground truth (or pseudo-label). This introduces limitations: the performance of amortized methods is inherently capped by the quality of the proxy explanations; and they also rely on the assumptions made by the proxy methods.

In this work, we propose a novel policy gradient based approach to learn a model-specific explainer that is not only capable of making fast explanation to any black-box models, but also have no reliance on pseudo-labels from existing proxy explanation methods.

Our **main contributions** are summarized as follows: 1) To the best of our knowledge, this is one of the first work that leverages reinforcement learning to directly learn an efficient explainer directly from data and the prediction model. 2) Unlike other amortized methods, our method doesn't rely on

the pseudo-labels provided by any proxy explanation method, such as SHAP. 3) A KL-divergence regularization is also introduced to enhance the generalizability of the learned explainer. 4) Comprehensive qualitative and quantitative experiments across multiple datasets demonstrate the superior quality and efficiency of our approach.

2 Related Work

In this work, we focus on explanations in the format of feature attribution [Linardatos *et al.*, 2020], i.e., finding the importance score for individual input features that influence a prediction. Therefore, we review three categories of feature attribution methods that are closely related to our approach: *model-agnostic approaches*, *model-specific approaches*, and *amortized approaches*.

Model-agnostic approaches Model-agnostic approaches are designed to be broadly applicable, making minimal or no assumptions about the to-be-explained prediction models. One common strategy involves using an explainable surrogates to approximate the local behavior of models, which is particularly useful for black-box models. For instance, LIME [Ribeiro *et al.*, 2016] fits a surrogate interpretable model (such as a linear model) to explain predictions locally by perturbing the input data and observing the changes in predictions. Similarly, SHAP [Lundberg and Lee, 2017] leverages Shapley values from game theory to ensure a unique surrogate solution with desirable properties such as local accuracy, missingness, and consistency. RISE [Petsiuk *et al.*, 2018] generates saliency maps by sampling randomized masks and evaluating their impact on the model's output.

Another category of model-agnostic techniques leverages gradient information from white-box models to provide explanations. Instead of learning surrogates, these methods exploit locally smoothed gradients to approximate the model's local behavior. The smoothing strategies vary among approaches. For instance, Integrated Gradients [Sundararajan *et al.*, 2017] computes explanations by averaging gradients of interpolated samples between a baseline input and the target input. AGI [Pan *et al.*, 2021] refines this concept by averaging gradients along multiple adversarial attack trajectories, while NeFLAG [Li *et al.*, 2023] utilizes gradients averaged over a hyperspherical neighborhood.

Although these methods are usually widely applicable, they are resource-intensive during inference due to the necessity of a large number of additional model queries.

Model-specific approaches Model-specific approaches are typically tailored to specific model architectures, enabling efficient explanations by utilizing attention weights, convolutional feature maps or custom layers. GradCAM [Selvaraju *et al.*, 2017], for instance, uses the weighted average of the convolutional feature maps to generate attributions, effectively working on CNNs. Similarly, methods like AttLRP [Chefer *et al.*, 2021] and AttCAT [Qiang *et al.*, 2022] are designed specifically for transformer-based models, relying on attention weights from various attention heads and layers to compute final explanations. DeepLIFT [Shrikumar *et al.*, 2017] provides a framework for explaining deep learning models under the condition that propagation rules can be adapted.

Amortized approaches Amortized explanation methods approximate the explanations from the resource-heavy model-agnostic methods (proxy methods) by a single forward pass. For example, FastSHAP [Jethani *et al.*, 2021], which amortizes the cost of fitting kernelSHAP by stochastically training a neural network to approximate it globally. CoRTX[Chuang *et al.*, 2023b], on the other hand, learns the explanation-oriented representation in a self-supervised manner and reduces the dependence of training on pseudo-labels from proxy methods. Overall, this type of methods achieves efficiency via an additional global surrogate function on top of the surrogates in model-agnostic methods, which introduces additional uncertainty.

In this paper, we propose a novel reinforcement learning framework that learns a distribution-based explainer that achieves the universality of model-agnostic approaches and the efficiency of model-specific approaches without relying on any proxy methods.

3 Proposed Method: Fast Explanation (FEX)

We lead with a discussion of an intractable empirical attribution, which relies on an exhaustive search over all possible feature combinations. Then, we interpret this empirical attribution as an expectation of a probability distribution and approximate this distribution via a policy gradient approach. Figure 1 illustrates the proposed model.

3.1 Empirical Attribution

For an input comprising N features, there exists 2^N distinct feature selection combinations. We represent each feature combination using a mask $\boldsymbol{m} \in \{0,1\}^N$. *For clarity, we define a mask entry of 0 to indicate that a feature is masked (removed), while a mask entry of 1 denotes that a feature is retained.*

Consider an input vector $\boldsymbol{x} = (x_1, ..., x_N)^\top$, with a *binary classification* function $f : X \to [0,1]$. For each masked version of the input, represented as $\boldsymbol{m} \odot \boldsymbol{x}$, we obtain a corresponding prediction $f(\boldsymbol{m} \odot \boldsymbol{x}) \in [0,1]$.

Intuitively, the **native contribution** of a feature x_i to $f(\boldsymbol{m} \odot \boldsymbol{x})$ can be represented by the average contribution from all features present in $\boldsymbol{m}$, i.e.,

$$c_i = \frac{f(\boldsymbol{m} \odot \boldsymbol{x})}{K_m} \tag{1}$$

where K_m denotes the number of non-zero entries in $\boldsymbol{m}$, representing the number of retained features in the masked input. This intuition for naive contribution comes from the philosophy that *all features should be viewed as equally important if there isn't any prior knowledge*.

Assume that there is a subset $M_i = \{\boldsymbol{m} | \boldsymbol{m} \in \{0,1\}^N, m_i = 1\}$, which comprises all masks that retain feature x_i. Then the sum of naive contributions of x_i over M_i is a good indicator of the importance of feature x_i.

Building on this, we define the empirical attribution as the total naive contribution from all masked inputs that retain the feature x_i. Specifically:

Definition 1. *(Empirical Attribution) Given an input vector $\boldsymbol{x} \in X$, a set of masks $M = \{0,1\}^N$, and the prediction*

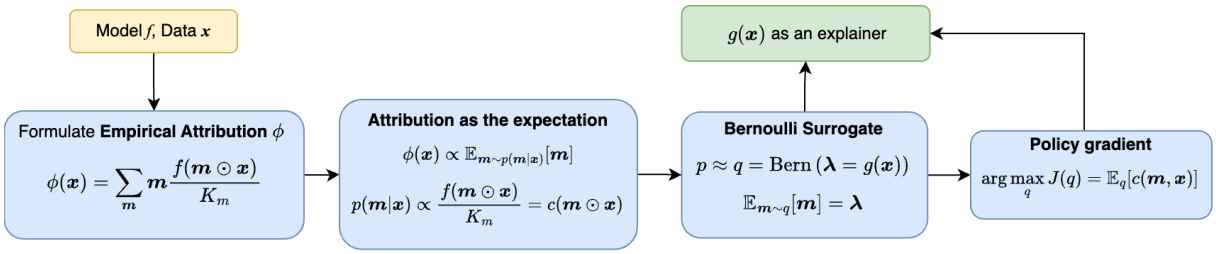

Figure 1: An illustration of the proposed method. The process begins with empirical attribution, calculated by summing over 2^N terms. To address the computational intractability of this summation, the attribution is reformulated as an expectation over a probability distribution p. Subsequently, p is approximated by a Bernoulli distribution q, enabling a closed-form solution that depends solely on the parameters of q. Finally, the parameters of q are optimized using the policy gradient method, yielding an approximation of the empirical attribution.

function $f(\boldsymbol{x}) : X \to [0, 1]$, the empirical attribution of a feature x_i is defined as:

$$\phi_i(\boldsymbol{x}) = \sum_{\boldsymbol{m} \in M_i} \frac{f(\boldsymbol{m} \odot \boldsymbol{x})}{K_m} = \sum_{\boldsymbol{m} \in M} m_i \cdot \frac{f(\boldsymbol{m} \odot \boldsymbol{x})}{K_m}, \quad (2)$$

where $M_i = \{\boldsymbol{m} \in M | m_i = 1\}$ represents the set of masks retain feature x_i, and K_m represents the number of non-zero elements in $\boldsymbol{m}$.

The cumulative empirical attribution can also be expressed in vector form when denoting $\phi(\boldsymbol{x})$ by:

$$\phi(\boldsymbol{x}) = (\phi_1(\boldsymbol{x}), \phi_2(\boldsymbol{x}), ..., \phi_N(\boldsymbol{x}))^\top \quad (3)$$

Since $\boldsymbol{m} = (m_1, m_2, ..., m_N)^T$, the vector form of the empirical attribution becomes:

$$\phi(\boldsymbol{x}) = \sum_{\boldsymbol{m}} \boldsymbol{m} \cdot \frac{f(\boldsymbol{m} \odot \boldsymbol{x})}{K_m}, \quad (4)$$

where $\phi(\boldsymbol{x}) \in \mathcal{R}_+^N$.

3.2 Attribution as an Expectation

Calculating the empirical attribution is computationally prohibitive due to its exponential complexity of $O(2^N)$. A common approach to address this challenge is to approximate it via Monte Carlo simulation, as has been similarly demonstrated in the RISE [Petsiuk *et al.*, 2018] framework. However, Monte Carlo methods remain computationally intensive, and the quality of the attribution is highly dependent on the number of simulation steps.

Note that the empirical attribution $\phi(\boldsymbol{x}) \in \mathcal{R}_+^N$ in Eq 4 can be written as a form of probability expectation with an appropriate normalization factor $A(\boldsymbol{x}) = \sum_{\boldsymbol{m}} \frac{f(\boldsymbol{m} \odot \boldsymbol{x})}{K_m}$. Specifically, we have:

$$\phi(\boldsymbol{x}) \propto \mathbb{E}_{\boldsymbol{m} \sim p(\boldsymbol{m}|\boldsymbol{x})}[\boldsymbol{m}]. \quad (5)$$

where the distribution $p(\boldsymbol{m}|\boldsymbol{x})$ is defined by

$$p(\boldsymbol{m}|\boldsymbol{x}) = \frac{f(\boldsymbol{m} \odot \boldsymbol{x})}{A(\boldsymbol{x}) \cdot K_m}. \quad (6)$$

Although it is still intractable to directly calculate the expectations, it is possible to obtain the expectation as a closed form if $p(\boldsymbol{m}|\boldsymbol{x})$ follows some specific distribution families.

3.3 Tractable Bernoulli Surrogate

For the purpose of explanation, we chose a multivariate Bernoulli distribution for its natural alignment with the disentanglement of different features. It serves as a surrogate to $p(\boldsymbol{m}|\boldsymbol{x})$:

$$q = \text{Bern}(\boldsymbol{\lambda} = g(\boldsymbol{x})), \quad (7)$$

where $\boldsymbol{\lambda} \in [0, 1]^N$ is the mean parameter of the Bernoulli distribution, parameterized by a neural network $g(\boldsymbol{x})$.

Its expectation has a closed form. Specifically, for $\boldsymbol{m}$ sampled from q, we have:

$$\mathbb{E}_{\boldsymbol{m} \sim q}[\boldsymbol{m}] = \boldsymbol{\lambda}. \quad (8)$$

This property is particularly advantageous because it allows the mean parameter $\boldsymbol{\lambda} = g(\boldsymbol{x})$ to represent the empirical attribution directly if q can approximate p, as defined in Eq 5.

However, $p(\boldsymbol{m}|\boldsymbol{x})$ is not directly computable, instead, we start by defining a score function $c(\boldsymbol{m}, \boldsymbol{x})$ as

$$c(\boldsymbol{m}, \boldsymbol{x}) = \frac{f(\boldsymbol{m} \odot \boldsymbol{x})}{K_m} \propto p(\boldsymbol{m}|\boldsymbol{x}), \quad (9)$$

Note that the expectation is primarily influenced by regions of the probability distribution with high density. Therefore, to approximate p with q, we need to optimize q such that the high density region matches p. Therefore, assume there are T masks $\boldsymbol{m}_1, ..., \boldsymbol{m}_T$ sampled from q, we aim to maximize the following objective:

$$\max_q J(q) = \mathbb{E}_q \left[\frac{1}{T} \sum_{t=1}^{T} c(\boldsymbol{m}_t, \boldsymbol{x}) \right]. \quad (10)$$

3.4 Policy Gradient

In the reinforcement learning literature, objectives with a structure similar to Eq. 10 can be effectively optimized using policy gradient methods. The policy gradient framework is characterized by four fundamental components: states (of the environment), actions (by the agent), the policy (for generating actions), and the return (of a series of actions).

Policy Gradients Adaptation

To adapt our framework to the policy gradient methodology, it is crucial to establish a clear correspondence between the key concepts in our approach and those traditionally utilized in

the policy gradient literature. This section provides a detailed mapping of these conceptual alignments.

Lets rephrase our problem as follows: Given the score function c, and original model input $\boldsymbol{x}$, we need to find a distribution q such that it maximizes Eq 10.

Input $\boldsymbol{x}$ as Static States: In policy gradient framework, a state s_t represents the current situation or configuration of the environment with which the agent interacts. In our context, since an input sample doesn't change over the masking actions, we consider these samples as static states. Formally, $s_t = \boldsymbol{x}$, where $t = 0, 1, ..., T$.

Mask $\boldsymbol{m}$ as Actions In our framework, applying masks to static input samples can be viewed as actions applied towards the states. i.e., $a_t = \boldsymbol{m}_t$.

Bernoulli Surrogate q as the Policy The policy in reinforcement learning generates actions. Similarly, in our context, the mask distribution q can be viewed as the policy that generates the masks. Specifically, $\boldsymbol{m}_t \sim q$.

Return Consequently, the weighted score function $\frac{1}{T}c(\boldsymbol{m}, \boldsymbol{x})$ performs as the reward given an action $\boldsymbol{m}$ upon state $\boldsymbol{x}$. Furthermore, if we define τ as a trajectory of a mask sequence $\boldsymbol{m}_1, ..., \boldsymbol{m}_T$, the return R can be computed by

$$R(\tau) = \sum_{\boldsymbol{m}_t \in \tau} \frac{c(\boldsymbol{m}_t, \boldsymbol{x})}{T} \tag{11}$$

Objective Function The objective function can be formally expressed as

$$J(q) = \mathbb{E}_{\tau \sim q}[R(\tau)] \tag{12}$$

where τ is a trajectory sampled from q.

Policy Gradient Formulation

Considering the above terminology connections between our framework and the policy gradient method, the gradient of the objective in Eq 12 can be expressed as:

$$\nabla J(q) = \mathbb{E}_{\tau \sim q}\left[\sum_{t=1}^{T} \nabla_q \log q(\boldsymbol{m}_t|\boldsymbol{x}) A^q(\boldsymbol{x}, \boldsymbol{m})\right]. \tag{13}$$

The advantage function $A^q(\boldsymbol{x}, \boldsymbol{m})$ is the difference between Action-Value function (Q-function) and Value function (V-function). The Q-function is defined as the expected return with the first action (mask) being $\boldsymbol{m}$

$$T \cdot Q^q(\boldsymbol{x}, \boldsymbol{m}) = c(\boldsymbol{m}, \boldsymbol{x}) + \mathbb{E}_{\tau \sim q}\left[\sum_{\boldsymbol{m_t} \in \tau, t \geq 2} c(\boldsymbol{m}_t, \boldsymbol{x})\right] \tag{14}$$

$$= c(\boldsymbol{m}, \boldsymbol{x}) + (T-1) \cdot \mathbb{E}_{\boldsymbol{m} \sim q}\left[c(\boldsymbol{m}, \boldsymbol{x})\right], \tag{15}$$

where the factor T is multiplied on both sides for conciseness. Similarly, the V-function is defined by the expected return if the first action $\boldsymbol{m}$ is sampled from q:

$$T \cdot V^q(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{m} \sim q}\left[c(\boldsymbol{m}, \boldsymbol{x})\right] + (T-1) \cdot \mathbb{E}_{\boldsymbol{m} \sim q}\left[c(\boldsymbol{m}, \boldsymbol{x})\right] \tag{16}$$

$$= T \cdot \mathbb{E}_{\boldsymbol{m} \sim q}\left[c(\boldsymbol{m}, \boldsymbol{x})\right]. \tag{17}$$

Therefore, the advantage function can be obtained by subtracting V from Q . Specifically,

$$A^q(\boldsymbol{x}, \boldsymbol{m}) = \frac{1}{T} \cdot (c(\boldsymbol{m}, \boldsymbol{x}) - V^q(\boldsymbol{x})). \tag{18}$$

Note that $V^q(\boldsymbol{x})$ can also be approximated by a neural network $v(\boldsymbol{x})$, which can be trained by minimizing the following loss function:

$$L_v(v) = \mathbb{E}_{\tau \sim q}\left[\sum_{t=0}^{T} \frac{1}{T}(c(\boldsymbol{m}_t, \boldsymbol{x}) - v(\boldsymbol{x}))^2\right]. \tag{19}$$

Proximal Policy Optimization: Policy gradient methods may suffer the issue of performance collapse when the policy changes too much during a single update. Therefore, we facilitate the clip trick used in PPO (Proximal Policy Optimization)[Schulman *et al.*, 2017] that constrains the update within each step. Consequently, the gradient in Eq 13 can be written by:

$$\nabla_q J(q) = \nabla_q L_{ppo} = \nabla_q \mathbb{E}_{\tau \sim q} \sum_{t=0}^{T} L(t), \tag{20}$$

$$L(t) = \min\left(\frac{q(\boldsymbol{m}_t|\boldsymbol{x})}{q^\ell(\boldsymbol{m}_t|\boldsymbol{x})} A^q(\boldsymbol{m}, \boldsymbol{x}), C A^q(\boldsymbol{m}, \boldsymbol{x})\right), \tag{21}$$

$$C = \text{clip}\left(\frac{q(\boldsymbol{m}_t|\boldsymbol{x})}{q^\ell(\boldsymbol{m}_t|\boldsymbol{x})}, 1-\epsilon, 1+\epsilon\right) \tag{22}$$

where q^ℓ represents the policy from the last updating step ℓ. Additionally, an **entropy regularization** term $H(q)$ is also added to balance the *exploration and the exploitation* during the reinforcement learning steps.

Combining the PPO objective, the entropy, and the MSE loss for $v(\boldsymbol{x})$, the objective function can then be written by

$$L = L_{ppo} - \lambda_{en} H(q) + \lambda_v L_v, \tag{23}$$

3.5 Generalizability

Generalizability in our framework involves two key aspects: (1) generalization over the distribution of all samples and (2) generalization across different output classes. Both are crucial for creating robust explainers that go beyond individual input-prediction pairs.

Generalization Over Sample Distribution: Generalization across samples ensures the explainer $g(\boldsymbol{x})$ consistently provides meaningful explanations over a dataset $\boldsymbol{X}$. When trained on a diverse dataset, our framework allows effective adaptation to diverse inputs.

Generalization Over Class Distribution: In multi-class classification, the prediction function $\boldsymbol{f}$ outputs a probability vector $(f_1, \ldots, f_K)^\top$ over K classes, requiring K explainers $\boldsymbol{g}_1, \ldots, \boldsymbol{g}_K$. Intuitively, when f_i dominates the predicted probabilities, the corresponding explainer $\boldsymbol{g}_i \in (0, 1)$ should also have a dominant average score. To enforce this alignment, KL-divergence is used to match the average explainer scores with the predicted class probabilities:

$$L_{\text{kl}} = \mathcal{D}_{\text{kl}}\left(\text{Softmax}\left(\frac{\sum_{i=1}^{N} \log \boldsymbol{g}_i}{N}\right), \boldsymbol{f}\right). \tag{24}$$

	FEX	FastSHAP	RISE	IG	GradSHAP	GradCAM	AttLRP
# propagation	$O(1)$	$O(1)$	$O(K)$	$O(K)$	$O(K)$	$O(1)$	$O(1)$
# backpropagation	0	0	0	$O(K)$	$O(1)$	$O(1)$	$O(1)$
Requires training	✓	✓	×	×	×	×	×
Proxy independent	✓	×	–	–	–	–	–
Model Agnostic	✓	✓	✓	✓	×	×	×
Blackbox	✓	✓	✓	×	×	×	×

Table 1: Comparison of computational costs, capabilities and limitations across different explanation methods. Here, K denotes the number of queries to the prediction model.

Algorithm 1 PPO for Fast Explanations

Input: training samples set X, prediction function $\boldsymbol{f} = (f_1, ..., f_K)^T$, initial explainer network $\boldsymbol{g} = (\boldsymbol{g}_1, ..., \boldsymbol{g}_K)^T$, initial value network $\boldsymbol{v} = (v_1, ..., v_K)^T$, and hyperparameters λ_{en}, λ_v, λ_{kl}
for $i = 1, 2, \ldots$ **do**
  Get a batch of input-output pairs $X_i \subset \{(\boldsymbol{x}, y) | \boldsymbol{x} \in X, y = \arg\max_k f_k(\boldsymbol{x})\}$
  for $\ell = 0, 1, 2, \ldots$ **do**
    Collect a set of trajectories $\mathcal{D}_\ell = \{\tau_j\}$ by running policy $q = \mathrm{Bern}(\boldsymbol{g}_y(\boldsymbol{x}))$ for all $(\boldsymbol{x}, y)$ pairs.
    Compute the Advantage $A^q(\boldsymbol{m}_t, \boldsymbol{x})$ by Eq 18.
    Obtain the PPO-Clip objective L_{ppo} by Eq 20, value network MSE loss L_v by Eq 19, KL-divergence L_{kl} by Eq 24, and entropy $H(q)$.
    Update the policy by minimizing the objective L in Eq 25
  end for
end for

Averaging the log values of the explainer scores ensures a more stable computation, as log probabilities are additive by nature. The softmax function is then applied to form a valid probability distribution. This approach guarantees consistency between the explainers and the classifier's output, facilitating robust and scalable explanations across classes. Consequently, the overall objective function for the policy gradient adaptation becomes:

$$L = L_{ppo} - \lambda_{en} H(q) + \lambda_v L_v + \lambda_{kl} L_{kl}, \tag{25}$$

3.6 Efficiency and Capabilities

Table 1 provides a detailed comparison of our Fast Explanation method (FEX) with several related explanation methods.

FEX distinguishes itself by requiring only $O(1)$ forward passes of $\boldsymbol{g}(\boldsymbol{x})$ during inference, ensuring exceptional computational efficiency. In contrast, other model-agnostic baselines, such as RISE, IG, and GradSHAP, require $O(K)$ queries to the prediction model. While methods like GradCAM and AttLRP achieve similar $O(1)$ efficiency in terms of model queries, they are inherently model-specific and therefore cannot be applied in a black-box setting.

Beyond its computational efficiency and model-agnostic nature, FEX offers an additional advantage: it does not depend on pseudo-labels generated by proxy explainers. For example, although FastSHAP achieves $O(1)$ efficiency, its reliance on pseudo-labels from SHAP introduces potential limitations, as its performance is constrained by the accuracy of the proxy explainer.

4 Experiments

We conduct experiments on both image and text classification tasks. For image classification, we use the ViT model [Dosovitskiy *et al.*, 2020] fine-tuned on the ImageNet dataset [Deng *et al.*, 2009] as the prediction model. The FEX explainer is finetuned on the full ImageNet dataset with 1.3M samples (FEX-1.3M) or a subset of 50,000 samples (FEX-50k) for one epoch . For text classification, we use the BERT model [Devlin *et al.*, 2018] fine-tuned on the SST2 dataset [Socher *et al.*, 2013] for sentiment analysis. The FEX explainer is finetuned on the Movies Reviews [Zaidan and Eisner, 2008] dataset for one epoch with batch size 256. Unless otherwise specified, in all experiments, the $g(\boldsymbol{x})$ is set to the same architecture as the predictor f, with appended MLP prediction heads, and the hyperparameters are set to $\lambda_{en} = 10^{-5}$, $\lambda_v = 0.5$ and $\lambda_{kl} = 1$.

4.1 Baselines

For the image classification task, we evaluate our proposed method against six baseline approaches, encompassing model-specific, model-agnostic, and amortized explanation techniques. The model-specific baselines include GradCAM, where we use the last hidden state as the target feature map, and AttLRP , where the default configurations from the original work are utilized. The model-agnostic baselines include GradSHAP, RISE and Integrated Gradients (IG). They require a number of queries (K) to the prediction model. In our experiments, Ks are set to 100 for all model-agnostic baselines. For the amortized methods, we include FastSHAP, where the explainer is implemented as a U-Net generating a 14×14 heatmap and is trained on 50,000 ImageNet samples (We are not able to train FastSHAP on the full ImageNet dataset because it's extremely slow).

For the text classification task, due to the discrete nature of text tokens, FastSHAP, IG and GradSHAP are not directly applicable. Hence, we only compare our method with RISE, GradCAM and AttLRP, with random attribuiton as a reference.

4.2 Metrics and Results

Figure 2 illustrates qualitative comparisons of various explanation methods for the image classification task. Our approach achieves comparable visual quality to model-specific

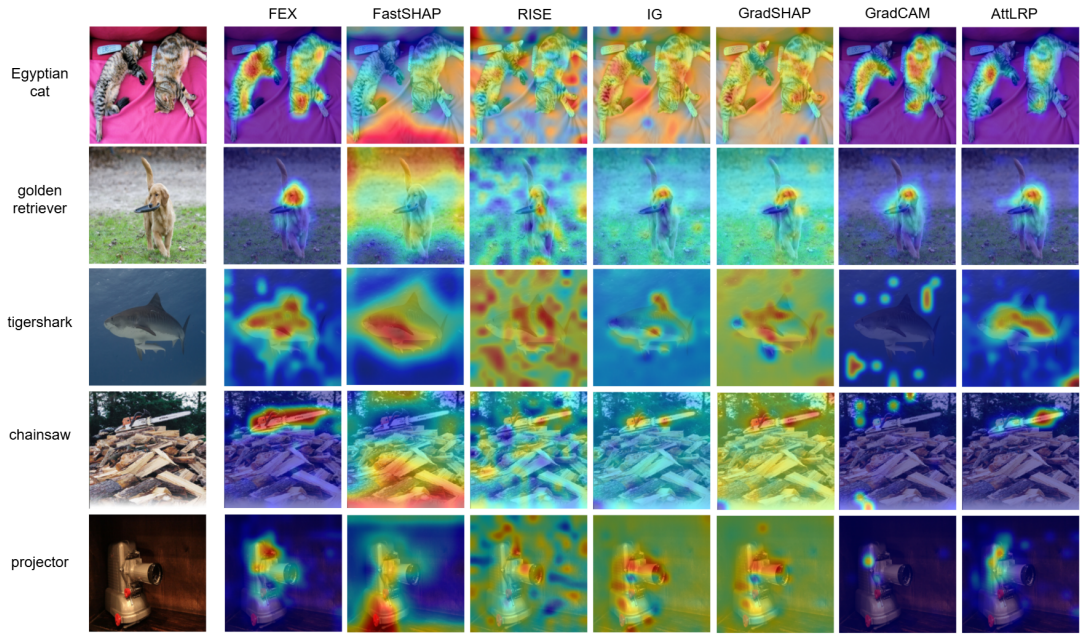

Figure 2: Qualitative examples for explaining the predictions in the image classification task.

	Ours		Amortized	Model-Agnostic			Model-Specific		Other
	FEX-50k	**FEX-1.3M**	**FastSHAP**	**RISE**	**IG**	**GradSHAP**	**GradCAM**	**AttLRP**	**Random**
Positive AUC ↓	0.3573	**0.3221**	0.4591	0.5040	0.4276	0.4599	0.5539	0.3652	0.6350
Negative AUC ↑	0.6892	**0.7296**	0.7084	0.7229	0.7216	0.7067	0.5546	0.7092	0.5790
Pixel Acc ↑	0.7862	**0.8172**	0.7674	0.5022	0.5643	0.7812	0.6786	0.8162	0.5064
mAP ↑	0.6714	**0.8939**	0.6749	0.5281	0.6135	0.6886	0.7311	0.8590	0.5050
mIoU ↑	0.4685	**0.6587**	0.4811	0.3022	0.3714	0.4958	0.4458	0.6517	0.3235

Table 2: Quantitative evaluation of explanation methods on the image classification task. Positive AUC and Negative AUC are evaluated on ImageNet dataset , while Pixel Accuracy (Pixel Acc), mean Average Precision (mAP), and mean Intersection over Union (mIoU) are reported on the image segamentation dataset.

methods such as AttLRP and GradCAM, while significantly surpassing model-agnostic baselines like IG and GradSHAP.

For quantitative evaluation, we follow the strategies outlined in [Chefer *et al.*, 2021]. For image classification, attribution performance is assessed using the area under the curve (AUC) of prediction accuracies, computed by progressively masking features ***(from 0% masked to 100% masked)*** based on their attributed importance. Positive AUC is calculated by masking the most important features first, whereas Negative AUC begins with the least important features. These evaluations are conducted on a randomly selected subset of 5,000 images from the ImageNet validation set.

To further assess the quality of explanations, we use an annotated image segmentation dataset [Guillaumin *et al.*, 2014] comprising 4,276 images across 445 categories. Segmentation labels serve as ground truth for attribution scores in the classification task. Performance is evaluated using proxy metrics: pixel accuracy, which measures the proportion of the most important patches falling within segmentation boxes; mean intersection over union (mIoU), quantifying overlap between segmentation boxes and features with above-average scores; and mean average precision (mAP), representing the area under the precision-recall curve with attribution scores as predictions.

Results for FEX and FastSHAP are averaged over three trained explainers, RISE, IG, GradSHAP, and Random are averaged over three runs, while only one run for GradCAM and AttLRP as they are deterministic. Table 2 demonstrate the superior performance of our proposed FEX method compared to other baselines.

For text classification, we adopt the ERASER benchmark [DeYoung *et al.*, 2019] and evaluate sentiment predictions on the Movie Reviews dataset [Zaidan and Eisner, 2008]. Explanations are evaluated by plotting F1 score curves as text tokens are progressively inserted based on their attributed importance. As shown in Figure 3, our method also achieves better performance on the text classification task.

In terms of efficiency, fine-tuning ViT on 1.3M ImageNet samples for a single epoch takes approximately 5 hours on a single A100 GPU, and one epoch is generally sufficient to achieve high-quality attribution scores. Notably, inference with FEX is lightweight, requiring only a single forward pass of $g(\boldsymbol{x})$. According to Table 3, our framework achieves the same level of inference efficiency as FastSHAP. It reduces inference time by over 97% and memory usage by 70% compared to traditional model-agnostic approaches (RISE, IG and

	FEX	FastSHAP	RISE	IG	GradSHAP	GradCAM	AttLRP
time (seconds)	**7.0**	11.6	260.2	311.9	313.2	14.9	106.8
memory (GB)	2.0	**1.2**	15.9	24.5	7.1	1.9	2.0
time × memory	14.0	**13.9**	4,137.2	7641.6	2,223.7	28.3	213.6

Table 3: Experiments on the inference cost for explaining 1000 image predictions of a pretrained ViT model. All experiments are conducted on the same machine with 8 CPU cores and 1 Nvidia A100 GPU.

	Trajectory Length s			Training Data Size		KL Coefficient l_{kl}			Trainable $g(x)$	
	$s=1$	$s=5$	$s=10$	FEX-50k	FEX-1.3M	$l_{kl}=0$	$l_{kl}=0.5$	$l_{kl}=1$	UNet	ViT
Positive AUC ↓	0.3383	**0.3221**	0.3222	0.3573	**0.3221**	0.3897	0.3294	**0.3221**	0.3352	**0.3221**
Negative AUC ↑	0.6977	**0.7296**	0.7282	0.6892	**0.7296**	0.6616	0.7185	**0.7296**	0.7130	**0.7296**

Table 4: Combined performance comparison across trajectory length s, training data size, KL-divergence coefficient l_{kl}, and trainable $g(x)$.

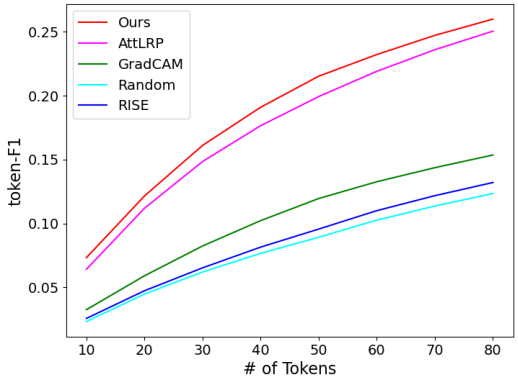

Figure 3: Quantitative evaluation results for the text classification task. The x-axis represents the number of text tokens inserted starting from the most important token, and the y-axis is the F1 score given that amount of tokens. The higher the better.

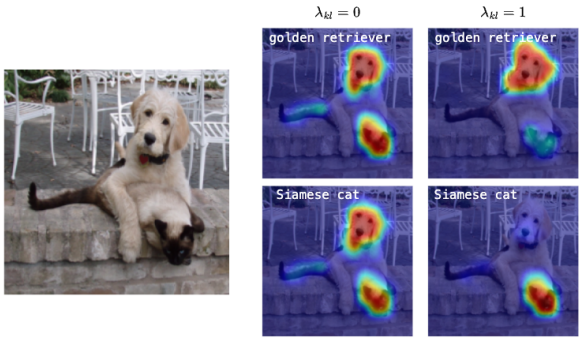

Figure 4: The top two predictions for this image are "golden retriever" and "Siamese cat". When $\lambda_{kl} = 0$, the explainer cannot differentiate these two classes. While when the KL regularization is introduced, it gains the ability to generalize over different classes.

GradSHAP).

5 Ablation Study

Effect of Trajectory Size The trajectory size sampled from the policy can impact its optimization. While longer trajectories can provide richer information for policy learning, computational constraints limit their feasibility during training. Striking a balance between trajectory size and computational efficiency is thus critical. As presented in Table 4, performance improves when the trajectory length increases from $s = 1$ to $s = 5$; however, it saturates when extending the trajectory further to $s = 10$. These results indicate that sampling excessively long trajectories is unnecessary.

Effect of Training Data Size Training data size is another crucial factor in achieving robust performance. The results in Table 4 highlight that the explainer trained on 1.3 million ImageNet samples (FEX-1.3M) significantly outperforms the one trained on only 50,000 samples (FEX-50k). This underscores the importance of using a sufficiently large dataset to enhance the explainer's generalization and reliability.

Effect of KL-Divergence Regularization The inclusion of KL-divergence regularization enhances the generalizability of the explainer across different classes. As shown in Figure 4, the absence of KL regularization results in a trained explainer that cannot effectively distinguish between classes. Additionally, the results in Table 4 indicate that introducing the KL-divergence regularization leads to improved performance.

Impact of $g(\boldsymbol{x})$ Selection $g(\boldsymbol{x})$ can be implemented as any neural network that takes $\boldsymbol{x}$ as input and outputs $\boldsymbol{\lambda} \in [0, 1]^N$, making it particularly suitable for scenarios where the prediction model f is treated as a black box. To evaluate the effect of different $g(\boldsymbol{x})$ choices, we compared UNet [Ronneberger *et al.*, 2015] and Vision Transformer (ViT) architectures. The results in Table 4 indicate no significant differences in performance, suggesting that the specific model structure is less critical as long as its capacity (e.g., parameter size) is adequate.

6 Conclusion

To address the challenge of balancing general applicability with inference speed in explainable AI (XAI), we proposed FEX framework that bridges the gap between the slow inference speed of model-agnostic methods and the limited applicability of model-specific methods. And unlike amortized approaches, which require existing model-agnostic methods as proxy explainers, our framework has no reliance on any proxy explainers. Experiments demonstrate that our method outperforms other baselines in both explanation quality and inference speed across various metrics.

Limitations

Similar to amortized methods, our framework requires training on a large and diverse dataset to achieve better quality, which may pose challenges when data privacy or data acquisition is a concern. A potential mitigation strategy is to train the explainer jointly with the predictor. This approach not only facilitates explainability for any prediction model but also ensures alignment between the explainer's domain and the predictor's application domain.

Broader Impact

Our framework enhances transparency and trust in AI, crucial for applications in sectors like healthcare. It aids debugging and bias identification, supporting ethical AI use and regulatory compliance. However, risks include potential oversimplification of explanations and exposure of proprietary model details. Addressing these challenges is key to maximizing positive impact.

References

[Atakishiyev *et al.*, 2021] Shahin Atakishiyev, Mohammad Salameh, Hengshuai Yao, and Randy Goebel. Explainable artificial intelligence for autonomous driving: A comprehensive overview and field guide for future research directions. *arXiv preprint arXiv:2112.11561*, 2021.

[Chefer *et al.*, 2021] Hila Chefer, Shir Gur, and Lior Wolf. Transformer interpretability beyond attention visualization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 782–791, 2021.

[Chen *et al.*, 2018] Jianbo Chen, Le Song, Martin Wainwright, and Michael Jordan. Learning to explain: An information-theoretic perspective on model interpretation. In *International conference on machine learning*, pages 883–892. PMLR, 2018.

[Chuang *et al.*, 2023a] Yu-Neng Chuang, Guanchu Wang, Fan Yang, Zirui Liu, Xuanting Cai, Mengnan Du, and Xia Hu. Efficient xai techniques: A taxonomic survey. *arXiv preprint arXiv:2302.03225*, 2023.

[Chuang *et al.*, 2023b] Yu-Neng Chuang, Guanchu Wang, Fan Yang, Quan Zhou, Pushkar Tripathi, Xuanting Cai, and Xia Hu. Cortx: Contrastive framework for real-time explanation. *arXiv preprint arXiv:2303.02794*, 2023.

[Deng *et al.*, 2009] 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*, pages 248–255. Ieee, 2009.

[Devlin *et al.*, 2018] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[DeYoung *et al.*, 2019] Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C Wallace. Eraser: A benchmark to evaluate rationalized nlp models. *arXiv preprint arXiv:1911.03429*, 2019.

[Dosovitskiy *et al.*, 2020] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[Fong and Vedaldi, 2017] Ruth C Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the IEEE international conference on computer vision*, pages 3429–3437, 2017.

[Grigorescu *et al.*, 2020] Sorin Grigorescu, Bogdan Trasnea, Tiberiu Cocias, and Gigel Macesanu. A survey of deep learning techniques for autonomous driving. *Journal of field robotics*, 37(3):362–386, 2020.

[Guillaumin *et al.*, 2014] Matthieu Guillaumin, Daniel Küttel, and Vittorio Ferrari. Imagenet auto-annotation with segmentation propagation. *International Journal of Computer Vision*, 110:328–348, 2014.

[Jethani *et al.*, 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*, 2021.

[Li *et al.*, 2023] Xin Li, Deng Pan, Chengyin Li, Yao Qiang, and Dongxiao Zhu. Negative flux aggregation to estimate feature attributions. *arXiv preprint arXiv:2301.06989*, 2023.

[Linardatos *et al.*, 2020] Pantelis Linardatos, Vasilis Papastefanopoulos, and Sotiris Kotsiantis. Explainable ai: A review of machine learning interpretability methods. *Entropy*, 23(1):18, 2020.

[Lundberg and Lee, 2017] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in neural information processing systems*, pages 4765–4774, 2017.

[Lundberg *et al.*, 2020] 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.

[Miotto *et al.*, 2018] Riccardo Miotto, Fei Wang, Shuang Wang, Xiaoqian Jiang, and Joel T Dudley. Deep learning for healthcare: review, opportunities and challenges. *Briefings in bioinformatics*, 19(6):1236–1246, 2018.

[Ozbayoglu *et al.*, 2020] Ahmet Murat Ozbayoglu, Mehmet Ugur Gudelek, and Omer Berat Sezer. Deep learning for financial applications: A survey. *Applied soft computing*, 93:106384, 2020.

[Pan *et al.*, 2021] Deng Pan, Xin Li, and Dongxiao Zhu. Explaining deep neural network models with adversarial gra-

dient integration. In *Thirtieth International Joint Conference on Artificial Intelligence (IJCAI)*, 2021.

[Petsiuk *et al.*, 2018] Vitali Petsiuk, Abir Das, and Kate Saenko. Rise: Randomized input sampling for explanation of black-box models. *arXiv preprint arXiv:1806.07421*, 2018.

[Qiang *et al.*, 2022] Yao Qiang, Deng Pan, Chengyin Li, Xin Li, Rhongho Jang, and Dongxiao Zhu. Attcat: Explaining transformers via attentive class activation tokens. *Advances in neural information processing systems*, 35:5052–5064, 2022.

[Ribeiro *et al.*, 2016] 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*, pages 1135–1144, 2016.

[Ronneberger *et al.*, 2015] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

[Schulman *et al.*, 2017] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[Selvaraju *et al.*, 2017] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[Shrikumar *et al.*, 2017] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International conference on machine learning*, pages 3145–3153. PMLR, 2017.

[Singh *et al.*, 2020] Amitojdeep Singh, Sourya Sengupta, and Vasudevan Lakshminarayanan. Explainable deep learning models in medical image analysis. *Journal of imaging*, 6(6):52, 2020.

[Socher *et al.*, 2013] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013.

[Sundararajan *et al.*, 2017] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328. PMLR, 2017.

[Zaidan and Eisner, 2008] Omar Zaidan and Jason Eisner. Modeling annotators: A generative approach to learning from annotator rationales. In *Proceedings of the 2008 conference on Empirical methods in natural language processing*, pages 31–40, 2008.