TVE: Learning Meta-attribution for Transferable Vision Explainer

Guanchu Wang [1] **Yu-Neng Chuang** [1] **Fan Yang** [2] **Mengnan Du** [3] **Chia-Yuan Chang** [4]
Shaochen Zhong [1] **Zirui Liu** [1] **Zhaozhuo Xu** [5] **Kaixiong Zhou** [6] **Xuanting Cai** [7] **Xia Hu** [1]

Abstract

Explainable machine learning significantly improves the transparency of deep neural networks. However, existing work is constrained to explaining the behavior of individual model predictions, and lacks the ability to transfer the explanation across various models and tasks. This limitation results in explaining various tasks being time- and resource-consuming. To address this problem, we introduce a Transferable Vision Explainer (TVE) that can effectively explain various vision models in downstream tasks. Specifically, the transferability of TVE is realized through a pre-training process on large-scale datasets towards learning the meta-attribution. This meta-attribution leverages the versatility of generic backbone encoders to comprehensively encode the attribution knowledge for the input instance, which enables TVE to seamlessly transfer to explain various downstream tasks, without the need for training on task-specific data. Empirical studies involve explaining three different architectures of vision models across three diverse downstream datasets. The experimental results indicate TVE is effective in explaining these tasks without the need for additional training on downstream data. The source code is available at https://github.com/guanchuwang/TVE.

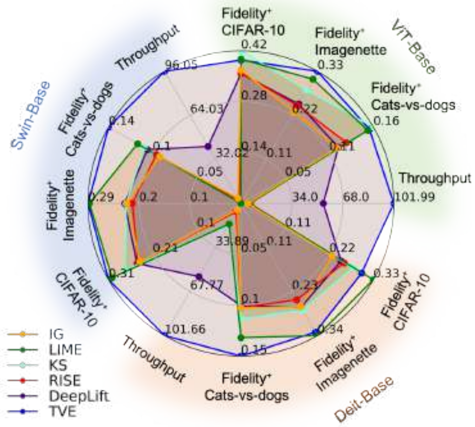


Figure 1. Performance of TVE in explaining ViT-B, Swin-B, and Deit-B on the Cats-vs-dogs, Imagenette, and CIFAR-10 datasets. *Fidelity$^+$ score* refers to the area under Fidelity$^+$-sparsity curve.

1. Introduction

Explainable machine learning (ML) contributes to enhancing the transparency of deep neural networks (DNNs) for human comprehension (Du et al., 2019). It significantly facilitates the deployment of DNNs to high-stake scenarios where model explanations are required, such as loan approvals (Steel & Angwin, 2010), healthcare (Chang et al., 2023), and targeted advertisement (Lin et al., 2021). In these fields, explainable DNN decisions are particularly important, given the practical needs of stakeholders and regulatory requirements, such as the General Data Protection Regulation (GDPR) (Goodman & Flaxman, 2017).

To overcome the black-box nature of DNNs, existing work of explainable ML can be categorized into two groups. The first group of work focuses on constructing local explanation based on perturbation of the target black-box model, like LIME (Ribeiro et al., 2016), GradCAM (Selvaraju et al., 2017), and Integrated Gradient (Sundararajan et al., 2017). These pieces of work rely on resource-intensive procedures like sampling or backpropagation of the target black-box model (Liu et al., 2021a), leading to undesirable trade-off between the efficiency and interpretation fidelity (Chuang et al., 2023a). Another group leverages the knowledge of explanation values to train DNN-based explainers, such as FastSHAP (Jethani et al., 2021), CORTX (Chuang et al., 2023b), and LARA (Rong et al., 2023; Wang et al., 2022). Such arts capable of efficiently generating explanations for an entire batch of instances through a single, streamlined

[1] Department of Computer Science, Rice University [2] Wake Forest University [3] New Jersey Institute of Technology [4] Texas A&M University [5] Stevens Institute of Technology [6] North Carolina State University [7] Meta Platforms, Inc. Correspondence to: Xia Hu <xia.hu@rice.edu>, Guanchu Wang <gw22@rice.edu>.

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

feed-forward operation of the DNN-based explainer. However, they are constrained to explaining individual black box models, and often lack the ability to transfer the explainer across various models or tasks. These constraints lead to a time and resource-intensive process in practical scenarios, as they require the development and training of separate explainers for each specific task.

To address the lack of transferability in explainers, we introduce a Transferable Vision Explainer (TVE). The primary goal of TVE is to achieve transferability through a pre-training process on large-scale image datasets, such that it can seamlessly explain various downstream tasks, as long as such tasks are within the scope of pre-training data distribution. The construction of such transferable explainers introduces two non-trivial challenges: **CH1.** Without task-specific exposure during the pre-training, how to ensure the universal effectiveness of explainer for various downstream tasks? **CH2.** How to adapt the explainer to a specific task without fine-tuning on the task-specific data?

Our work effectively tackles these challenges. To address CH1, we introduce a novel concept, named *meta-attribution*, as a foundation for explaining various downstream tasks. Specifically, the meta-attribution versatilely encodes the attribution knowledge for the input instance via exhaustively attributing each dimension of instance embedding. This knowledge is reusable for explaining various downstream tasks. It guides the pre-training of TVE on large-scale image datasets, ensuring the universal effectiveness of TVE. After the pre-training, in response to CH2, we propose a *transfer rule* to adapt the meta-attribution to explaining downstream tasks, without the need for additional training on task-specific data. Figure 1 shows the comprehensive performance of TVE pre-trained on the ImageNet dataset and transferred to the Cats-vs-dogs, Imagenette, and CIFAR-10 datasets, where TVE shows competitive fidelity and efficiency compared with state-of-the-art methods. To summarize, our work makes the following contributions:

- **Attribution transfer.** We propose a framework of attribution transfer, with a meta-attribution as foundations, and a transfer rule for explaining the downstream tasks.
- **Transferable explainer.** We build a transferable explainer TVE that explains various downstream tasks without the need for training on the task-specific data.
- **Theoretical foundation.** We validate the pre-training of TVE can minimize the explanation error bound aligned with the $\mathcal{V}$-information-based explanation.
- **Competitive performance in explaining various downstream tasks.** The pre-trained TVE shows promising results in explaining three architectures of vision Transformer across three downstream datasets. Significantly, the strong transferability of TVE facilitates efficient and flexible deployment to various downstream scenarios.

2. Notations

We introduce the notations for the problem formulation.

Target model. We focus on the explanation of vision models: $\mathcal{X} \rightarrow \mathcal{Y}_t$ in this work, where $\mathcal{X} = \mathbb{N}^{\mathrm{W}\times\mathrm{W}}$ denote the spatial space of W×W pixels; $\mathbb{N}$ denote the space of a single pixel with three channels; and $\mathcal{Y}_t$ denotes the label space. Moreover, we follow most of existing work (He et al., 2022) and implementation of DNNs (Wolf et al., 2020a) to consider the target model as $f_t = H_t \circ G$, where the backbone encoder $G(\bullet): \mathbb{N}^{\mathrm{W}\times\mathrm{W}} \rightarrow \mathbb{R}^{\mathrm{D}}$ is pre-trained on large-scale datasets; and the classifier $H_t(\bullet): \mathbb{R}^{\mathrm{D}} \rightarrow \mathcal{Y}_t$ is finetuned on a specific task t. It maybe worth noting that although we follow the transfer learning setting (Chilamkurthy, 2017; Chen et al., 2020) to freeze the backbone encoder $G(\bullet)$ during the fine-tuning of f_t. Our experiment results in Section 6.3 further show that the proposed transferable explanation framework also shows effectiveness in the scenario where the target model is fully fine-tuned on downstream data.

Image Patching. We follow existing work (Lundberg & Lee, 2017) to consider the patch-wise attribution of model prediction, i.e. the importance of each patch. Specifically, we follow existing work (Chuang et al., 2023b; Jethani et al., 2021) to split each image $\boldsymbol{x}_k$ into $\mathrm{P}\times\mathrm{P}$ patches in a grid pattern, where each patch has $\mathrm{C}\times\mathrm{C}$ pixels; and $\mathrm{W}=\mathrm{CP}$. Let $\mathcal{Z}(\boldsymbol{x}_k) = \{z_{i,j} | 1 \leq i, j \leq \mathrm{P}\}$ denote the patches of an image $\boldsymbol{x}_k \in \mathcal{X}$, where a patch $z \in \mathbb{N}^{\mathrm{C}\times\mathrm{C}}$ aligns with continuous $\mathrm{C}\times\mathrm{C}$ pixels of the image. Moreover, we define $\mathcal{N}(z) \subseteq \mathcal{Z}(\boldsymbol{x}_k)$ as the neighbors of a patch z within the grid space, because a patch together with its neighbors have richer semantic content for model explanation. In this work, we follow the vision transformer (Dosovitskiy et al., 2020) to split the image patches with $\mathrm{P}=14$ for 224×224 input images from the ImageNet dataset; and we consider $\mathcal{N}(z)$ as the zero-, one-, and two-hop neighbors of the patch z.

Model Perturbation. $f_t(\mathcal{S}; \boldsymbol{x}_k, y)$ represents the output of f_t on class y, with a perturbed instance as the input. The patch subset $\mathcal{S} \subseteq \mathcal{Z}(\boldsymbol{x}_k)$ controls the perturbation. Specifically, the pixels belonging to the patches $z \in \mathcal{Z}(\boldsymbol{x}_k) \setminus \mathcal{S}$ are removed and take 0, which is approximately the average value of normalized pixels. For example, $f_t(\mathcal{N}(z); \boldsymbol{x}_k, y)$ defines the output of f_t based on the perturbed input, where the pixels not belonging to the neighbors of patch z take 0.

Feature Attribution. This work focuses on the feature attribution of target models f_t for providing explanations. The feature attribution process involves generating importance scores, denoted as $\phi_{k,y,z}$ for each patch $z \in \mathcal{Z}(\boldsymbol{x}_k)$ of the input image $\boldsymbol{x}_k \in \mathcal{X}$, to indicate its importance to the model prediction $f_t(\mathcal{Z}(\boldsymbol{x}_k); \boldsymbol{x}_k, y)$ on class y.

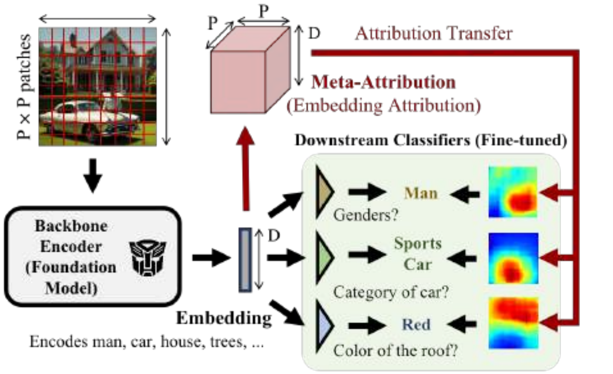


Figure 2. Illustration of attribution transfer. In this framework, the backbone can be a ViT encoder; and the downstream classifiers can be MLPs. The embedding vector comprehensively encodes the features of input image. Motivated by this, the meta-attribution comprehensively encapsulates the importance of each input patch to each element of the embedding vector. This enables it to seamlessly transfer for explaining various downstream tasks.

3. Feature Attribution can Transfer

The motivation behind attribution transfer stems from model transfer in vision tasks (Chilamkurthy, 2017; He et al., 2020; 2022). Specifically, it arises from the observation that a generic backbone encoder possesses the capability to capture essential features of input images and represent them as embedding vectors. This versatility enables the backbone to effectively adapt to a wide range of downstream tasks within the scope of pre-training data distribution. As shown in Figure 2, information of man, car, house encoded in the embedding vector enables the detection of gender, car, and building in three different downstream scenarios, respectively. Despite the demonstrated transferability of the backbone encoder, existing research has challenges in achieving 'transferable explainer' across different tasks. To bridge this gap and streamline the explanation process, we propose a *meta-attribution* that can be applied across various tasks, resulting in a significant reduction in the cost associated with generating explanations.

The *meta-attribution* is defined as a tensor that versatilely encodes the reusable attribution knowledge for explaining downstream tasks. As shown in Figure 2, we illustrate the meta-attribution as a three-dimensional tensor. A simple and effective method in this work is attributing the importance of input patches to each element of the embedding vector for the meta-attribution. As shown in Figure 2, each P × P slice of this tensor corresponds to P × P patches within the input image, encoding their importance to a specific dimension of the embedding vector. In this way, the meta-attribution inherits the adaptability of the embedding vector, making it versatile enough to adapt various explanation tasks in downstream scenarios. For instance, the meta-attribution encodes the attribution knowledge for the man and car components encoded in the embedding vector, such that it can transfer to explain the car classification and gender detection in downstream scenarios. The versatility of meta-attribution can effectively address the CH1 described in Section 1. We formalize the attribution transfer in Sections 4.

4. Meta-attribution Transfer

In this section, we begin with the explanation definition by following the $\mathcal{V}$-information theory (Xu et al., 2020; Hewitt et al., 2021; Chen et al., 2022). Then, we introduce the definition of meta-attribution in Definition 1. Finally, we propose a transfer rule to adapt the meta-attribution to explaining specific downstream tasks in Definition 2.

4.1. $\mathcal{V}$-Information-based Explanation

The importance of a patch $z \in \mathcal{Z}(\boldsymbol{x}_k)$ to downstream model $f_t(\boldsymbol{x}_k)$ is formulated into the conditional mutual information $I(\mathcal{N}(z); Y_t \mid B)$ between $\mathcal{N}(z)$ and Y_t, given the state of remaining patches $B \subseteq \mathcal{Z}(\boldsymbol{x}_k) \backslash \mathcal{N}(z)$ (Chen et al., 2022). Here, $Y_t \sim f_t(\boldsymbol{x}_k)$ denotes the variable corresponding to the model ouput. However, estimating this mutual information accurately poses a challenge due to the unknown distribution of $\mathcal{N}(z)$ and B. To address this challenge, we adopt an information-theoretic framework introduced in works by (Xu et al., 2020; Hewitt et al., 2021), known to as conditional $\mathcal{V}$-information $I_{\mathcal{V}}(\mathcal{N}(z) \to Y_t)$. In particular, it redirects the computation of mutual information to a certain predictive model within function space $\mathcal{V}$, as defined by:

$$I_{\mathcal{V}}(\mathcal{N}(z) \to Y_t \mid B) = H_{\mathcal{V}}(Y_t \mid B) - H_{\mathcal{V}}(Y_t \mid \mathcal{N}(z), B),$$

where $\mathcal{V}$-entropy $H_{\mathcal{V}}(Y_t \mid B)$ takes the lowest entropy over the function space $\mathcal{V}$, which is given by

$$H_{\mathcal{V}}(Y_t \mid B) = \inf_{f \in \mathcal{V}} \mathbb{E}_{y \sim \mathcal{Y}_t}[-\log f(B; \boldsymbol{x}_k, y)]. \quad (1)$$

Note that the $\mathcal{V}$-information explanation should align with a pre-trained target model $f_t \in \mathcal{V}$ and a specific class label y. The $\mathcal{V}$-entropy should take its value at f_t and y, instead of the infimum expectation value, for the explanation. Therefore, we relax the $\mathcal{V}$-entropy terms $H_{\mathcal{V}}(Y_t \mid B)$ and $H_{\mathcal{V}}(Y_t \mid \mathcal{N}(z), B)$ into $-\log f_t(B; \boldsymbol{x}_k, y)$ and $-\log f_t(\mathcal{N}(z) \cup B; \boldsymbol{x}_k, y)$, respectively (Ethayarajh et al., 2022), for aligning the explanation with the target model $f_t \in \mathcal{V}$ and class label y. In this way, the attribution of patch z aligned with class y is defined as follows:

$$\phi_{k,y,z} = \mathbb{E}_{B \subseteq \mathcal{Z}(\boldsymbol{x}_k) \backslash \mathcal{N}(z)} [-\log f_t(B; \boldsymbol{x}_k, y) + \log f_t(\mathcal{N}(z) \cup B; \boldsymbol{x}_k, y)]. \quad (2)$$

It is impossible to enumerate the state of B over $B \subseteq \mathcal{Z}(\boldsymbol{x}_k) \setminus \mathcal{N}(z)$ in Equation (2). We follow existing work (Mitchell et al., 2022) to approximate it into two antithetical states to simplify the computation (Mitchell et al., 2022). These cases involve considering the state of B to be entirely remaining patches $\mathcal{Z}(\boldsymbol{x}_k) \setminus \mathcal{N}(z)$ or empty set $\varnothing$, narrowing down the enumeration of $B \subseteq \mathcal{Z}(\boldsymbol{x}_k) \setminus \mathcal{N}(z)$ to $B \sim \{\mathcal{Z}(\boldsymbol{x}_k) \setminus \mathcal{N}(z), \varnothing\}$ in Equation (2). Based on our numerical studies in Appendix B, the approximate attribution shows positive correlation with the exact value, which indicates the approximation does not affect the quality of attribution. To summarize, we approximate the attribution value of patch z aligned with class y as follows:

$$\phi_{k,y,z} \approx \mathbb{E}_{B\sim\{\mathcal{Z}(\boldsymbol{x}_k)\setminus\mathcal{N}(z),\varnothing\}} \left[-\log f_t(B;\boldsymbol{x}_k,y)+\log f_t(\mathcal{N}(z)\cup B;\boldsymbol{x}_k,y)\right], \quad (3)$$

$$\sim \log f_t(\mathcal{N}(z);\boldsymbol{x}_k,y)-\log f_t(\mathcal{Z}(\boldsymbol{x}_k)\setminus\mathcal{N}(z);\boldsymbol{x}_k,y), \quad (4)$$

where the terms $f_t(\mathcal{Z}(\boldsymbol{x}_k);\boldsymbol{x}_k,y)$ and $f_t(\varnothing;\boldsymbol{x}_k,y)$ in Equation (3) are constant given $\boldsymbol{x}_k$ and y, thus being omitted in Equation (4). Intuitively, the explanation of patch z depends on the gap of logit values, where $\mathcal{N}(z)$ and background patches $\mathcal{Z}(\boldsymbol{x}_k) \setminus \mathcal{N}(z)$ are taken as the input.

4.2. Definition of Meta-attribution

We introduce the concept of meta-attribution, formally defined in Definition 1. Note that Equation (4) relies on the downstream target model f_t, which is task-related. The purpose of meta-attribution is to disentangle the task-specific aspect of the attribution from Equation (4). This disentanglement renders the meta-attribution to be task-independent, as a foundation for explaining various tasks.

Definition 1 (**Meta-attribution**). *Given a backbone encoder G, the meta-attribution for a patch $z \in \mathcal{Z}(\boldsymbol{x}_k)$, $\boldsymbol{x}_k \in \mathcal{X}$, is represented by two tensors $\mathbf{g}_{k,z}$ and $\boldsymbol{h}_{k,z}$ as follows:*

$$\begin{aligned} \mathbf{g}_{k,z} &= G(\mathcal{N}(z);\boldsymbol{x}_k), \\ \boldsymbol{h}_{k,z} &= G(\mathcal{Z}(\boldsymbol{x}_k)\setminus\mathcal{N}(z);\boldsymbol{x}_k). \end{aligned} \quad (5)$$

Following Definition 1, the meta-attribution is defined as the input tensors of the logarithmic functions in Equation (4), where the task-specific model f_t is replaced into the backbone encoder G to disentangle the meta-attribution with specific tasks. This disentanglement enables the meta-attribution to transfer across various downstream tasks.

4.3. Transfer to Task-aligned Explanation

To explain the downstream tasks, we propose a transfer rule in Definition 2 to adapt the meta-attribution to explaining downstream tasks. This rule-based transfer method can effectively address the CH2 described in Section 1, without the need for additional training on task-specific data.

Definition 2 (**Attribution Transfer**). *If the task-specific function is given by $f_t = H_t \circ G$, then the explanation of $f_t(\boldsymbol{x}_k)$ on class y is generated by*

$$\phi_{k,y,z} = \log H_t(\mathbf{g}_{k,z};y) - \log H_t(\boldsymbol{h}_{k,z};y), \quad (6)$$

where $\mathbf{g}_{k,z}$ and $\boldsymbol{h}_{k,z}$ are the meta-attribution given by Equation (5); and G and H_t represent the backbone encoder and fine-tuned classifier on task t, respectively.

Following Definition 2, we can straightforwardly achieve the solution of $\phi_{k,y,z}$ to be consistent with Equation (4)[1]. This alignment to $\phi_{k,y,z}$ can effectively explain downstream task t following the definition of conditional $\mathcal{V}$-information $I_{\mathcal{V}}(\mathcal{N}(z) \to Y_t \mid B)$, as described in Section 4.1.

5. Learning Meta-attribution

In this section, we introduce the details of Transferable Vision Explainer (TVE). Specifically, TVE pre-trains a DNN-based transferable explainer $E(\bullet \mid \theta)$ on large-scale image dataset to comprehensively learn the knowledge of meta-attribution. After the pre-training, TVE can transfer to various downstream tasks for end-to-end generating task-aligned explanation. To assess its performance, we theoretically analyze the explanation error in Theorem 1.

5.1. Explainer Pre-training

TVE employs a DNN-based explainer $E(\bullet \mid \theta)$ to generate the meta-attribution tensors. Specifically, the explainer $E(\bullet \mid \theta)$ produces two tensors for the meta-attribution, denoted as $[\hat{\mathbf{g}}_k, \hat{\mathbf{h}}_k] = E(\boldsymbol{x}_k \mid \theta)$, where $\hat{\mathbf{g}}_k = [\hat{\mathbf{g}}_{k,z} \in \mathbb{R}^{\mathrm{D}} | z \in \mathcal{Z}(\boldsymbol{x}_k)]$ and $\hat{\mathbf{h}}_k = [\hat{\mathbf{h}}_{k,z} \in \mathbb{R}^{\mathrm{D}} | z \in \mathcal{Z}(\boldsymbol{x}_k)]$ represent collections of meta-attribution for an instance $\boldsymbol{x}_k$. Each pair of elements $(\hat{\mathbf{g}}_{k,z}, \hat{\mathbf{h}}_{k,z})$ contribute to predicting the meta-attribution $(\mathbf{g}_{k,z}, \mathbf{h}_{k,z})$ defined in Definition 1. Pursuant to this objective, TVE updates the parameters of explainer $E(\bullet \mid \theta)$ to minimize the following loss function:

$$\mathcal{L}_\theta(\boldsymbol{x}_k) = \mathbb{E}_{z\sim\mathcal{Z}(\boldsymbol{x}_k)}\left[||\hat{\mathbf{g}}_{k,z} - \mathbf{g}_{k,z}||_2^2 + ||\hat{\mathbf{h}}_{k,z} - \mathbf{h}_{k,z}||_2^2\right], \quad (7)$$

where $\mathbf{g}_{k,z}$ and $\mathbf{h}_{k,z}$ are defined in Definition 1.

Algorithm 1 summarizes one epoch of pre-training the transferable explainer $E(\bullet \mid \theta)$. Specifically, TVE first samples a mini-batch of image patches (lines 2); then follows Definition 1 to generate the meta-attribution (lines 3); finally updates the parameters of $E(\bullet \mid \theta)$ to minimize the loss function given by Equation (7) (line 4). The iteration ends with the convergence of $E(\bullet \mid \theta)$. Notably, the pre-training of $E(\bullet \mid \theta)$ is guided by the meta-attribution instead of specific tasks. This empowers the trained $E(\bullet \mid \theta)$ to remain impartial towards specific tasks, providing the flexibility for seamless adaptation across various downstream tasks.

Algorithm 1 One epoch of TVE pre-training

```
Input: Pre-training dataset $\mathcal{D}$.
Output: Transferable explainer $E(\bullet \mid \theta^*)$.
1: for $\boldsymbol{x}_k \sim \mathcal{D}$ do
2:     Sample patches $z \sim \mathcal{Z}(\boldsymbol{x}_k)$.
3:     Generate $\mathbf{g}_{k,z}$ and $\mathbf{h}_{k,z}$ following Definition 1.
4:     Update $E(\bullet \mid \theta)$ to minimize Equation (7).
5: end for
```

[1] We follow Definition 2 to have $\phi_{k,y,z} = \log H_t(\mathbf{g}_{k,z};y) - \log H_t(\mathbf{h}_{k,z};y) = \log f_t(\mathcal{N}(z);\boldsymbol{x}_k,y) - \log f_t(\mathcal{Z}(\boldsymbol{x}_k)\setminus\mathcal{N}(z);\boldsymbol{x}_k,y)$ that is consistent with Equation (4).

5.2. Generating Task-aligned Explanation

TVE follows Definition 2 to generate the task-aligned explanation. Specifically, to explain the inference process $(H_t \circ G)(\boldsymbol{x}_k)$ in task t, TVE first adopts the pre-trained transferable explainer to generate the meta-attribution $[\hat{\boldsymbol{g}}_k, \hat{\mathbf{h}}_k] = E(\boldsymbol{x}_k \mid \theta)$; then takes the value of $\hat{\boldsymbol{g}}_{k,z}$ and $\hat{\mathbf{h}}_{k,z}$ into Equation (6) to estimate the importance of each patch $z \in \mathcal{Z}(\boldsymbol{x}_k)$ to the inference result on class y. To summarize, TVE generates the attribution of a patch $z \in \mathcal{Z}(\boldsymbol{x}_k)$ by

$$\hat{\phi}_{k,y,z} = \log H_t(\hat{\boldsymbol{g}}_{k,z}; y) - \log H_t(\hat{\mathbf{h}}_{k,z}; y). \tag{8}$$

Let $\hat{\boldsymbol{\phi}}_{k,y} = [\hat{\phi}_{k,y,z} \mid z \in \mathcal{Z}(\boldsymbol{x}_k)]$ denote the P×P explanation heatmap for the image $\boldsymbol{x}_k$, indicating the importance of all patches in $\boldsymbol{x}_k$ to class y. TVE can efficiently generate the entire heatmap $\hat{\boldsymbol{\phi}}_{k,y}$ for the image $\boldsymbol{x}_k$ through a single feed forward pass : $\hat{\boldsymbol{\phi}}_{k,y} = \log H_t(\hat{\boldsymbol{g}}_k; y) - \log H_t(\hat{\mathbf{h}}_k; y)$, where $\hat{\boldsymbol{g}}_k$ and $\hat{\mathbf{h}}_k$ are generated by $[\hat{\boldsymbol{g}}_k, \hat{\mathbf{h}}_k] = E(\boldsymbol{x}_k \mid \theta)$.

In particular, $H_t(\bullet; y)$ in Equation (8) encodes the knowledge of downstream task t. This knowledge significantly enables the explanation to align with the task t *without the need for additional training on the task-specific data*.

5.3. Theoretical Analysis

The theoretical analysis focuses on understanding the behavior of estimation error $|\hat{\phi}_{k,y,z} - \phi_{k,y,z}|$ during the TVE pre-training, where $\phi_{k,y,z}$ takes the $\mathcal{V}$-Information-aligned explanation defined in Section 4.1. Specifically, we examine the following two distinct cases to understand how the reduction in the pre-training loss function $\mathcal{L}_\theta(\boldsymbol{x}_k)$ diminishes the estimation error $|\hat{\phi}_{k,y,z} - \phi_{k,y,z}|$.

Ideal Case. We ideally consider $\mathcal{L}_\theta(\boldsymbol{x}_k) \to 0$ in this case. According to Equation (7), we have that $\hat{\boldsymbol{g}}_{k,z} \to \boldsymbol{g}_{k,z}$ and $\hat{\mathbf{h}}_{k,z} \to \mathbf{h}_{k,z}$. Then, the relations $\frac{H_t(\hat{\mathbf{g}}_{k,z};y)}{H_t(\mathbf{g}_{k,z};y)} \to 1$ and $\frac{H_t(\mathbf{h}_{k,z};y)}{H_t(\hat{\mathbf{h}}_{k,z};y)} \to 1$ are established. In this context, we have $|\hat{\phi}_{k,y,z} - \phi_{k,y,z}| \to 0$ according to Equations (6) and (8). This indicates $\hat{\phi}_{k,y,z}$ exactly converges to $\phi_{k,y,z}$ in the ideal scenario.

Practical Case. Without loss of generality, we consider $\mathcal{L}_\theta(\boldsymbol{x}_k)$ is not reduced to zero in this case. Specifically, Equation (7) indicates the reduction of $\mathcal{L}_\theta(\boldsymbol{x}_k)$ leads to $\hat{\boldsymbol{g}}_{k,z}$ and $\hat{\mathbf{h}}_{k,z}$ gradually approach $\boldsymbol{g}_{k,z}$ and $\mathbf{h}_{k,z}$, respectively. As a result, the values of $\frac{H_t(\hat{\mathbf{g}}_{k,z};y)}{H_t(\mathbf{g}_{k,z};y)}$ and $\frac{H_t(\mathbf{h}_{k,z};y)}{H_t(\hat{\mathbf{h}}_{k,z};y)}$ gradually converge to a narrower range around 1. We formulate this trend by assuming their values to be bounded within a range of $1-\epsilon \leq \frac{H_t(\hat{\mathbf{g}}_{k,z};y)}{H_t(\mathbf{g}_{k,z};y)}, \frac{H_t(\mathbf{h}_{k,z};y)}{H_t(\hat{\mathbf{h}}_{k,z};y)} \leq 1+\epsilon$, where $0 \leq \epsilon \ll 1$. Under these assumptions, we establish the upper bound of $|\hat{\phi}_{k,y,z} - \phi_{k,y,z}|$ in Theorem 1, with a detailed proof in Appendix C. This allows us to understand the behavior of estimation error in practical cases where $\mathcal{L}_\theta(\boldsymbol{x}_k)$ is not reduced to zero.

Theorem 1 (Explanation Error Bound). *Given the classifier $H_t(\bullet;\bullet)$ of the downstream task, if the output of classifier $H_t(\hat{\boldsymbol{g}}_{k,z}; y)$ and $H_t(\hat{\boldsymbol{h}}_{k,z}; y)$ fall within the range of $1-\epsilon \leq \frac{H_t(\hat{\mathbf{g}}_{k,z};y)}{H_t(\mathbf{g}_{k,z};y)}, \frac{H_t(\boldsymbol{h}_{k,z};y)}{H_t(\hat{\boldsymbol{h}}_{k,z};y)} \leq 1+\epsilon$, then, the upper bound of explanation error is given by*

$$\mathbb{E}_{\boldsymbol{x}_k \sim \mathcal{D}_t, y \sim \mathcal{Y}_t, z \sim \mathcal{Z}(\boldsymbol{x}_k)} |\hat{\phi}_{k,y,z} - \phi_{k,y,z}| \leq \frac{2\epsilon}{1-\epsilon}, \tag{9}$$

where $\hat{\phi}_{k,y,z}$ and $\phi_{k,y,z}$ are given by Equation (8) and (6), respectively; and $\mathcal{D}_t$ denotes the downstream dataset.

Intuition of Theorem 1. The value of ϵ reduces as the pre-training loss function $\mathcal{L}_\theta(\boldsymbol{x}_k)$ decreases. This reduction in ϵ explicitly lowers the estimation error bound $\frac{2\epsilon}{1-\epsilon}$ aligned with the $\mathcal{V}$-Information-aligned explanation $\phi_{k,y,z}$ on downstream tasks. This underscores the TVE pre-training can significantly enhance the explanations for downstream tasks.

6. Experiment Results

In this section, we conduct experiments to evaluate TVE by answering the following research questions: **RQ1:** How does TVE perform compared with state-of-the-art baseline methods in terms of the fidelity? **RQ2:** How does TVE perform in explaining fully fine-tuned target model on downstream datasets? **RQ3:** How is the transferability of TVE across different downstream datasets? **RQ4:** Do both pre-training and attribution transfer in TVE contribute to explaining downstream tasks?

6.1. Experiment Setup

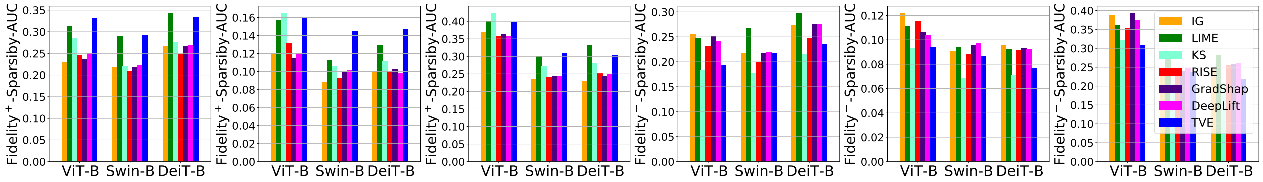
We clarify the datasets, target models, hyper-parameter settings in this section. More details about the baseline methods, evaluation metrics and implementation details are given in Appendixes F, G, and H, respectively.

Datasets. We consider the large-scale ImageNet dataset for TVE pre-training; and the Cats-vs-dogs (Elson et al., 2007), CIFAR-10 (Krizhevsky et al., 2009), and Imagenette (Howard, 2019) datasets for the downstream explaining tasks. More details are given in Appendix D.

Target Models. We comprehensively consider three architectures of vision transformers for downstream classification tasks, including the ViT-Base (Dosovitskiy et al., 2020), Swin-Base (Liu et al., 2021b), Deit-Base (Touvron et al., 2021) transformers. We consider two settings of fine-tuning target models: *classifier-tuning* and *full-fine-tuning*. More details about the target model are given in Appendix E.

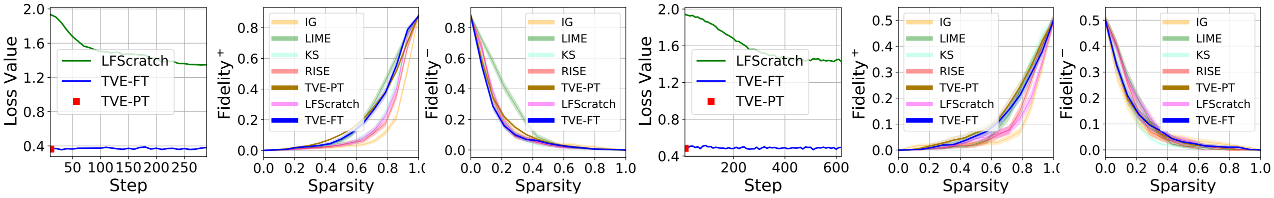
Hyper-parameter Settings. The experiment follows the pipeline of TVE pre-training, explanation generation and evaluation on multiple downstream datasets. Specifically, TVE adopts the Mask-AutoEncoder (He et al., 2022) as the backbone, followed by multiple Feed-Forward (FFN) layers[2] to generate the meta-attribution. More details about the

[2] A Mask-AutoEncoder consists of a ViT encoder followed by a ViT decoder; and an FFN layer consists of Linear layers, Layer-norm, and activation function, which are widely used in the Transformer structure. More details about the architecture are given in Appendix H.



(a) Imagenette (b) Cats-vs-dogs (c) CIFAR-10 (d) Imagenette (e) Cats-vs-dogs (f) CIFAR-10

Figure 3. Fidelity$^+$-Sparsity-AUC(↑) on the `Imagenette` (a), `Cat-vs-dogs` (b), and `CIFAR-10` (c) datasets. Fidelity$^-$-Sparsity-AUC(↓) on the `Imagenette` (d), `Cat-vs-dogs` (e), and `CIFAR-10` (f) datasets.



(a) (b) (c) (d) (e) (f)

Figure 4. Fine-tuning loss versus epoch (a), Fidelity$^+$ ↑ versus Sparsity (b), and Fidelity$^-$ ↓ versus Sparsity (c) on the Imagenette dataset. Fine-tuning loss versus epoch (d), Fidelity$^+$ ↑ versus Sparsity (e), and Fidelity$^-$ ↓ versus Sparsity (f) on the cats-vs-dogs dataset.

explainer architecture and hyper-parameters of pre-training `TVE` are given in Appendix H. When deploying `TVE` to explaining downstream tasks, the explanation aligns with the prediction class given by the target model.

6.2. Evaluation of Fidelity (RQ1)

In this section, we evaluate the fidelity of `TVE` under the classifier-tuning setting. Due to the space constraints, we present 18 figures illustrating the Fidelity$^+$-sparsity curve(↑) and the Fidelity$^-$-sparsity curve(↓) for explaining the `ViT-Base`, `Swin-Base`, and `Deit-Base` models on the `Cats-vs-dogs`, `Imagenette`, and `CIFAR-10` datasets in Appendix I. To streamline our evaluation, we simplify the assessment of fidelity-sparsity curves by calculating its Area Under the Curve (AUC) over the sparsity from zero to one, which aligns with the average fidelity value. Intuitively, a higher Fidelity$^+$-sparsity-AUC(↑) indicates superior Fidelity$^+$(↑) across most sparsity levels, reflecting a more faithful explanation. Similarly, a lower Fidelity$^-$-sparsity-AUC(↓) signifies a more faithful explanation. More details about the fidelity-sparsity-AUC are given in Appendix G. On the `Cats-vs-dogs`, `Imagenette`, and `CIFAR-10` datasets, we present the Fidelity$^+$-sparsity-AUC(↑) for explanations in Figures 3 (a)-(c), respectively, as well as the Fidelity$^-$-sparsity-AUC(↓) in Figures 3 (d)-(f), respectively. We have the following observations:

- *`TVE` consistently exhibits promising performance in terms of both* Fidelity$^+$*(↑) and* Fidelity$^-$*(↓)*, outperforming the majority of baseline methods. This underscores `TVE` faithfully explains various downstream tasks within the scope of pre-training data distribution.
- *`TVE` exhibits significant strengths in both* Fidelity$^+$*(↑) and* Fidelity$^-$*(↓)*, highlighting its effectiveness in identifying both important and non-important features. In contrast, the baseline methods fail to simultaneously achieve high Fidelity$^+$ and low Fidelity$^-$. For example, consider `LIME`'s performance when explaining the `Deit-Base` model on the `CIFAR-10` dataset. While `LIME` excels in Fidelity$^+$, it falls short in Fidelity$^-$.

6.3. Explaining Fully Fine-tuned Models (RQ2)

In this section, we evaluate the fidelity of `TVE` under the full-fine-tuning setting to demonstrate its generalization ability. Notably, the `ViT-Base` classification model including both the backbone and classifier are fine-tuned on downstream data, which are not available to `TVE` pre-training. The explanation considers three methods: learning from scratch (`LFScratch`), `TVE` pre-training (`TVE`-PT), and `TVE` fine-tuning (`TVE`-FT). To adapt to the fully fine-tuned target model, `LFScratch` trains the explainer on the downstream dataset for one epoch; `TVE`-PT simply transfers the pre-trained explainer to explaining the down-stream tasks; `TVE`-FT follows Algorithm 1 to fine-tune the explainer using the fine-tuned backbone encoder on the downstream dataset for one epoch. Here, we consider the `Imagenette` and `Cat-vs-dogs` datasets for the downstream tasks. Further details about fine-tuning the target models and explainers are given in Appendixes E and H, respectively. The loss value of `LFScratch` and `TVE`-FT versus the fine-tuning steps are shown in Figures 4 (a) and (d). The fidelity-sparsity curves of all methods are given in Figures 4 (b), (c), (e), and (f). We have the following observations:

- *`TVE` pre-training provides a good initial explainer for adaption to fully fine-tuned encoders.* According to Figures 4 (a,d), the `TVE` pre-trained explainer shows lower training loss than learning from scratch in the early epochs. This indicates the pre-training provides a good initial explainer for explaining downstream tasks.

Table 1. Explanation Fidelity$^+$-Sparsity-AUC(↑) and Fidelity$^-$-Sparsity-AUC(↓) for `Deit-Base`, `Swin-Base`, and `Deit-Base` target models on the `Cat-vs-dogs`, `Imagenette`, and `CIFAR-10` datasets.

	Datasets	Cats-vs-dogs		Imagenette		CIFAR-10	
Target model	Method	Fidelity$^+$(↑)	Fidelity$^-$(↓)	Fidelity$^+$(↑)	Fidelity$^-$(↓)	Fidelity$^+$(↑)	Fidelity$^-$(↓)
ViT-Base	`ViTShapley`	0.11±0.09	0.13±0.10	0.25±0.13	0.25±0.14	0.36±0.17	0.36±0.17
	`TVE`-H_g	0.14±0.11	0.10±0.08	0.29±0.14	**0.18**±0.10	0.39±0.18	0.34±0.17
	`TVE`	**0.16**±0.13	**0.09**±0.07	**0.33**±0.16	0.19±0.12	**0.40**±0.18	**0.31**±0.16
Swin-Base	`ViTShapley`	0.09±0.05	0.11±0.07	0.24±0.07	0.24±0.09	0.25±0.11	0.28±0.14
	`TVE`-H_g	**0.14**±0.09	0.10±0.07	**0.29**±0.08	0.24±0.07	0.26±0.12	0.27±0.13
	`TVE`	**0.14**±0.10	**0.09**±0.05	**0.29**±0.10	**0.22**±0.06	**0.31**±0.14	**0.24**±0.12
DeiT-Base	`ViTShapley`	0.12±0.08	0.1±0.07	0.22±0.09	0.29±0.11	0.28±0.13	0.24±0.13
	`TVE`-H_g	0.13±0.08	0.09±0.06	**0.33**±0.10	0.25±0.08	**0.32**±0.14	0.24±0.13
	`TVE`	**0.15**±0.10	**0.08**±0.06	**0.33**±0.10	**0.24**±0.08	0.30±0.13	**0.22**±0.12

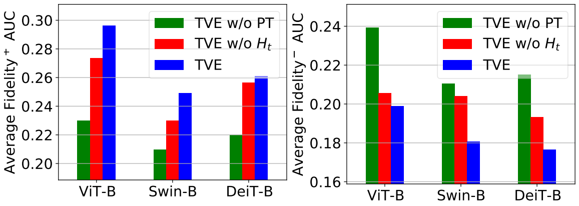


Figure 5. Fidelity of ablation studies.

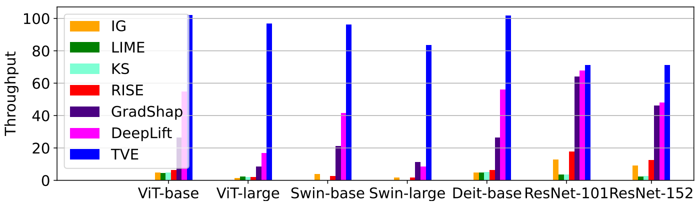


Figure 6. Throughput of explaining different architectures.

- *`TVE`-PT can effectively explain the fully fine-tuned target model, even without fine-tuning the explainer on downstream datasets.* According to Figures 4 (b,c,e,f), `TVE`-PT shows competitive fidelity when comparing with `TVE`-FT and other baseline methods, and a significant improvement over `LFScratch`. This indicates the strong generalization ability of `TVE`, acquired through pre-training on the large-scale `ImageNet` dataset.

- *The pre-training of transferable explainer and fine-tuning of backbone encoder can be executed independently and parallelly.* Specifically, `TVE` pre-trains the transferable explainer based on open-sourced pre-trained backbone encoders and large-scale `ImageNet` dataset; meanwhile, the encoder can be fine-tuned in parallel on downstream datasets. This can significantly improve the efficiency and flexibility of deploying `TVE` to practical scenarios.

6.4. Evaluation of Transferability (RQ3)

We evaluate the transferability of `TVE` compared with `ViT-Shapley` (Covert et al., 2022), a state-of-the-art DNN-based explainer for vision models. Specifically, `ViT-Shapley` pre-trains the explainer on the large-scale `ImageNet` dataset, and deploys it to the `Cat-vs-dogs`, `Imagenette`, and `CIFAR-10` datasets to generate the explanations. Different from `ViT-Shapley`, `TVE` transfers the explainer to downstream datasets via taking the task-specific classifier H_t into Equation (8). Moreover, we also consider a `TVE`-H_g method to study whether the pre-training of `TVE` contributes to explaining downstream tasks. Different from `TVE`, `TVE`-H_g takes a general classifier (pre-trained on the `ImageNet` dataset) into Equation (8) to generate the explanation. We follow Section 6.2 to adopt the fidelity-sparsity AUC to evaluate the average fidelity. Table 1 illustrates the fidelity for explaining the `ViT-Base`, `Swin-Base`, and `Deit-Base` models on the `Cat-vs-dogs`, `Imagenette`, and `CIFAR-10` datasets. We have the following insights:

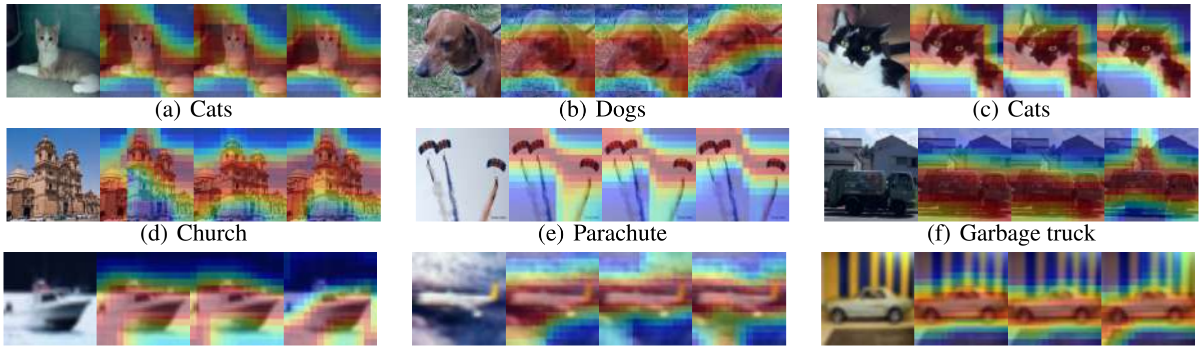
- *`TVE` has stronger transferability than `ViT-Shapley`.* Both `TVE` and `ViT-Shapley` are pre-trained on the large-scale `ImageNet` dataset, and transferred to the downstream datasets without additional training. Table 1 shows `TVE` has higher Fidelity$^+$(↑) and lower Fidelity$^-$(↓) than `ViT-Shapley`.

- *The pre-training of `TVE` significantly contributes to explaining downstream tasks.* `TVE`-H_g adopts the generally pre-trained explainer and classifier to explain downstream tasks, and achieves a reasonable fidelity on most of the datasets. This indicates the pre-training of `TVE` captures the transferable features across various datasets for explaining downstream tasks.

- *It is more faithful to explain downstream tasks based on the task-specific classifiers.* `TVE` outperforms `TVE`-H_g on most architectures and datasets, which indicates the attribution transfer had better take the classifier aligned with the downstream task for H_t in Definition 2.

6.5. Ablation Studies (RQ4)

We ablatedly study the contribution of the key steps in `TVE` to explaining downstream tasks, including the pre-training of transferable explainer and attribution transfer aligned to each task. For our evaluation, we consider three methods: `TVE` w/o Pre-training (PT), `TVE` w/o H_t, and `TVE`. Specifically, for `TVE` w/o PT, the explainer is randomly initialized without pre-training, and attribution transfer follows Definition 2. For `TVE` w/o H_t, the transferable explainer is pre-trained following Al-



(a) Cats (b) Dogs (c) Cats

(d) Church (e) Parachute (f) Garbage truck

(g) Ship (h) Airplane (i) Automobile

Figure 7. Visualization of explanation on the `Cats-vs-dogs` (a)-(c), `Imagenette` (d)-(f), and `CIFAR-10` (g)-(i) datasets. From the left to the right, each heatmap explains the inference of the `Swin-Base`, `Deit-Base`, and `ViT-Base` models, respectively.

gorithm 1, and the explanation for each task is generated by $\hat{\phi}_k = \log H_g(\hat{\boldsymbol{g}}_k; y) - \log H_g(\hat{\mathbf{h}}_k; y)$, where H_g takes a general classifier pre-trained on the `ImageNet` datasets, instead of being fine-tuned corresponding to the task. Figure 5 illustrates the results of Fidelity$^+$-Sparsity-AUC(↑) and Fidelity$^-$-Sparsity-AUC(↓) for each method, where the fidelity score represents the averaged value on the `Cats-vs-dogs`, `Imagenette`, and `CIFAR-10` datasets. Other configurations remain consistent with Appendix H. Overall, we have the following observations:

- *`TVE` pre-training significantly contributes to explaining the downstream tasks.* This can be verified by the fidelity degradation observed from `TVE` w/o PT in Figure 5.
- *The classifier H_t for attribution transfer should align with the explaining task t.* It is observed in Figure 5 that `TVE` outperforms `TVE` w/o H_t. This indicates the task-aligned H_t is better than general classifiers for the attribution transfer to a specific task t.

6.6. Evaluation of Latency

In this section, we evaluate the latency of `TVE` compared with baseline methods. Specifically, we adopt the metric Throughput $= \frac{N_{\text{test}}}{T}$(↑) to evaluate the explanation latency, where N_{test} takes the number of testing instances and T signifies the total time consumed during the explanation process. Details about our computational infrastructure are given in Appendix N. Figure 6 shows the throughput of different methods explaining the `ViT-Base/Large`, `Swin-Base/Large`, `Deit-Base`, and `ResNet-101/152` models on the `ImageNet` dataset. Overall, we observe:

- *`TVE` is more efficient than state-of-the-art baseline methods,* by generating explanations through a single feed-forward pass of the explainer. In contrast, the baseline methods rely on intensive samplings of the forward or backward passes of the target model, resulting in a considerably slower explanation process. For example, although `KernelSHAP` exhibits comparable Fidelity$^-$(↓) with `TVE`, as shown in Figure 3, its significantly lower throughput limits its practicality in real-world scenarios.
- *`TVE` exhibits the most negligible decrease in throughput as the size of the target model grows*, as seen when transitioning from `ViT-Base` to `ViT-Large`. This advantage stems from the fact that `TVE`'s latency is contingent upon the explainer's model size, rather than the target model. In contrast, the baseline methods suffer from notable performance slowdown as the size of the target model increases, due to the necessity of sampling the target model to generate explanations.

6.7. Case Studies

In this section, we visualize the explanations generated by `TVE`, demonstrating its power in helping human users understand vision models. Specifically, we randomly sample three instances from the `Cats-vs-dogs`, `Imagenette`, and `CIFAR-10` datasets, and visualize the explanations of `Swin-Base`, `Deit-Base`, and `ViT-Base` models in Figure 7, where sub-figures (a)-(c), (d)-(f), and (g)-(i) correspond to the `Cats-vs-dogs`, `Imagenette`, and `CIFAR-10` datasets, respectively. In each sub-figure, from the left-side to the right-side, the three heatmaps explain the inference of the `Swin-Base`, `Deit-Base`, and `ViT-Base` model, respectively. Notably, `TVE` generates the explanation heatmap in an end-to-end manner *without pre- or post-processing*. More case studies on the `ImageNet` dataset are shown in Appendix O. According to the case study, we observe:

- *The salient patches emphasized by `TVE`'s explanation reveal semantically meaningful patterns.* For example, as depicted in Figures 7 (d), (e), and (g), the `Swin-Base` model concentrates on the tower, canopy and bow, respectively, to identify a church, parachute, and ship.
- *`TVE` does not rely on pre-processing of the image or post-processing of the explanation heatmap.* In contrast, existing work EAC (Sun et al., 2023) requires SAM (Kirillov et al., 2023) to segment the input image before explaining, which is less flexible than `TVE`.
- *Different model architectures make predictions based on*

distinct image elements. For instance, as illustrated in Figure 7 (g), the `Swin-Base` and `Deit-Base` models primarily emphasize the ship's bow for identification. In contrast, the `ViT-Base` model takes into account the ship's keel for its prediction.

7. Conclution

In this work, we propose a framework of attribution transfer, incorporating a meta-attribution to extract the foundation knowledge and a transfer rule to utilize this knowledge for explaining various downstream tasks. Building upon this framework, we introduce `TVE`, a transferable explainer pretrained on large-scale image datasets. Notably, `TVE` shows strong transferability to effectively explain various downstream tasks without the need for training on task-specific data. Experiment results validate the promising performance of `TVE` in explaining three architectures of vision Transformer across three downstream datasets. Significantly, the strong transferability of `TVE` facilitates efficient and flexible deployment to various downstream scenarios.

Impact Statement

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

Acknowledgement

The authors thank the anonymous reviewers for their helpful comments. The work is in part supported by the grants of NSF IIS-1900990 and IIS-2310260, and the NIH Aim-Ahead program. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agencies.

References

Ancona, M., Ceolini, E., Öztireli, C., and Gross, M. Towards better understanding of gradient-based attribution methods for deep neural networks. *arXiv preprint arXiv:1711.06104*, 2017.

Chang, C.-Y., Yuan, J., Ding, S., Tan, Q., Zhang, K., Jiang, X., Hu, X., and Zou, N. Towards fair patient-trial matching via patient-criterion level fairness constraint. *arXiv preprint arXiv:2303.13790*, 2023.

Chen, H., Brahman, F., Ren, X., Ji, Y., Choi, Y., and Swayamdipta, S. Rev: information-theoretic evaluation of free-text rationales. *arXiv preprint arXiv:2210.04982*, 2022.

Chen, L., Lou, S., Zhang, K., Huang, J., and Zhang, Q. Harsanyinet: Computing accurate shapley values in a single forward propagation. *arXiv preprint arXiv:2304.01811*, 2023.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

Chilamkurthy, S. Transfer learning for computer vision tutorial. *PyTorch Tutorials*, 2017.

Chuang, Y.-N., Wang, G., Yang, F., Liu, Z., Cai, X., Du, M., and Hu, X. Efficient xai techniques: A taxonomic survey. *arXiv preprint arXiv:2302.03225*, 2023a.

Chuang, Y.-N., Wang, G., Yang, F., Zhou, Q., Tripathi, P., Cai, X., and Hu, X. Cortx: Contrastive framework for real-time explanation. *arXiv preprint arXiv:2303.02794*, 2023b.

Covert, I., Kim, C., and Lee, S.-I. Learning to estimate shapley values with vision transformers. *arXiv preprint arXiv:2206.05282*, 2022.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Du, M., Liu, N., and Hu, X. Techniques for interpretable machine learning. *Communications of the ACM*, 63(1): 68–77, 2019.

Elson, J., Douceur, J. J., Howell, J., and Saul, J. Asirra: A captcha that exploits interest-aligned manual image categorization. In *Proceedings of 14th ACM Conference on Computer and Communications Security (CCS)*. Association for Computing Machinery, Inc., October 2007.

Ethayarajh, K., Choi, Y., and Swayamdipta, S. Understanding dataset difficulty with v-usable information. pp. 5988–6008, 2022.

Goodman, B. and Flaxman, S. European union regulations on algorithmic decision-making and a "right to explanation". *AI magazine*, 38(3):50–57, 2017.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738, 2020.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009, 2022.

Hewitt, J., Ethayarajh, K., Liang, P., and Manning, C. D. Conditional probing: measuring usable information beyond a baseline. *arXiv preprint arXiv:2109.09234*, 2021.

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

Jethani, N., Sudarshan, M., Covert, I. C., Lee, S.-I., and Ranganath, R. Fastshap: Real-time shapley value estimation. In *International Conference on Learning Representations*, 2021.

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023.

Kokhlikyan, N., Miglani, V., Martin, M., Wang, E., Alsallakh, B., Reynolds, J., Melnikov, A., Kliushkina, N., Araya, C., Yan, S., et al. Captum: A unified and generic model interpretability library for pytorch. *arXiv preprint arXiv:2009.07896*, 2020.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Lin, T.-W., Sun, R.-Y., Chang, H.-L., Wang, C.-J., and Tsai, M.-F. Xrr: Explainable risk ranking for financial reports. In *Machine Learning and Knowledge Discovery in Databases. Applied Data Science Track: European Conference, ECML PKDD 2021, Bilbao, Spain, September 13–17, 2021, Proceedings, Part IV 21*, pp. 253–268. Springer, 2021.

Liu, Y., Khandagale, S., White, C., and Neiswanger, W. Synthetic benchmarks for scientific research in explainable machine learning. 2021a.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012–10022, 2021b.

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

Mitchell, R., Cooper, J., Frank, E., and Holmes, G. Sampling permutations for shapley value estimation. *The Journal of Machine Learning Research*, 23(1):2082–2127, 2022.

Petsiuk, V., Das, A., and Saenko, K. Rise: Randomized input sampling for explanation of black-box models. *arXiv preprint arXiv:1806.07421*, 2018.

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

Rong, Y., Wang, G., Feng, Q., Liu, N., Liu, Z., Kasneci, E., and Hu, X. Efficient gnn explanation via learning removal-based attribution. *arXiv preprint arXiv:2306.05760*, 2023.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pp. 618–626, 2017.

Steel, E. and Angwin, J. On the web's cutting edge, anonymity in name only. *The Wall Street Journal*, 4, 2010.

Sun, A., Ma, P., Yuan, Y., and Wang, S. Explain any concept: Segment anything meets concept-based explanation. *arXiv preprint arXiv:2305.10289*, 2023.

Sun, Y., Chen, Q., He, X., Wang, J., Feng, H., Han, J., Ding, E., Cheng, J., Li, Z., and Wang, J. Singular value fine-tuning: Few-shot segmentation requires few-parameters fine-tuning. *Advances in Neural Information Processing Systems*, 35:37484–37496, 2022.

Sundararajan, M., Taly, A., and Yan, Q. Axiomatic attribution for deep networks. In *International conference on machine learning*, pp. 3319–3328. PMLR, 2017.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., and Jégou, H. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pp. 10347–10357. PMLR, 2021.

Wang, G., Chuang, Y.-N., Du, M., Yang, F., Zhou, Q., Tripathi, P., Cai, X., and Hu, X. Accelerating shapley explanation via contributive cooperator selection. *arXiv preprint arXiv:2206.08529*, 2022.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. M. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp.

38–45, Online, October 2020a. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.emnlp-demos.6.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pp. 38–45, 2020b.

Xu, Y., Zhao, S., Song, J., Stewart, R., and Ermon, S. A theory of usable information under computational constraints. *arXiv preprint arXiv:2002.10689*, 2020.

Yang, Z., Liu, N., Hu, X. B., and Jin, F. Tutorial on deep learning interpretation: A data perspective. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pp. 5156–5159, 2022.

Appendix

A. Related Work

Explainable machine learning (ML) has made significant advancements, leading to model transparency and better human understanding of deep neural networks (DNNs) (Du et al., 2019). Specifically, existing work of explainable ML can be categorized into two groups: local explainers and DNN-based explainers (Chuang et al., 2023a).

Local Explainer. Local explainer focuses on constructing local explanation based on perturbation of the target black-box model, like KernelSHAP (Lundberg & Lee, 2017), LIME (Ribeiro et al., 2016), GradCAM (Selvaraju et al., 2017), and Integrated Gradient (Sundararajan et al., 2017). Specifically, KernelSHAP approximates the Shapleyvalue by learning an explainable surrogate (linear) model based on the DNN output of reference input for each feature; LIME generates the explanation by sampling points around the input instance and using DNN output at these points to learn a surrogate (linear) model; Integrated Gradients estimates the explanation by the integral of the gradients of DNN output with respect to the inputs, along the pathway from specified references to the inputs. These pieces of work rely on resource-intensive procedures like sampling or backpropagation of the target black-box model (Liu et al., 2021a), leading to undesirable trade-off between the efficiency and interpretation fidelity (Chuang et al., 2023a).

DNN-based Explainer. This branch of work leverages the training process to acquire proficiency in constructing a DNN-based explainer, utilizing explanation values as training labels (Chuang et al., 2023a). This innovative strategy empowers the simultaneous generation of explanations for an entire batch of instances through a single, streamlined feedforward operation of the DNN-based explainer. Exemplifying this progress are innovative approaches like FastSHAP (Jethani et al., 2021), ViT-Shapley (Covert et al., 2022), CORTX (Chuang et al., 2023b), LARA (Rong et al., 2023; Wang et al., 2022), and HarsanyiNet (Chen et al., 2023). To be concrete, FastSHAP and ViT-Shapley adopt a DNN as the explainer to learn the Shapley value, which relies on task-specific training and cannot be transferred across different tasks; and CoRTX arguments the training of DNN-based explainer through a contrastive pre-training framework, and adopt the true Shapley value to fine-tune the explainer. The DNN-based explainer have played a pivotal role in significantly streamlining the deployment of DNN explanations within real-time applications. However, they are constrained to explaining individual black box models, and they lack the ability to transfer the explanation across various models and tasks. This limitation results in the explanation of various tasks in practical scenarios becoming time- and resource-consuming due to the necessity of training different explainers for each task.

B. Approximation of Attribution

We conduct experiments to study the relationship between the approximate attribution $\mathbb{E}_{B\sim\{\mathcal{Z}(\boldsymbol{x}_k)\backslash\mathcal{N}(z),\varnothing\}}[\cdots]$ and its exact value $\mathbb{E}_{B\sim\text{Subset of }\mathcal{Z}(\boldsymbol{x}_k)\backslash\mathcal{N}(z)}[\cdots]$ on the `ImageNet` dataset, where $\cdots$ is the abbreviation of $-\log f_t(B;\boldsymbol{x}_k,y)+\log f_t(\mathcal{N}(z)\cup B;\boldsymbol{x}_k,y)$. Specifically, we collect the samples of $\mathbb{E}_{B\sim\{\mathcal{Z}(\boldsymbol{x}_k)\backslash\mathcal{N}(z),\varnothing\}}[\cdots]$ and $\mathbb{E}_{B\sim\text{Subset of }\mathcal{Z}(\boldsymbol{x}_k)\backslash\mathcal{N}(z)}[\cdots]$, where $\boldsymbol{x}_k$ take 100 instances randomly sampled from the `ImageNet` dataset; and the target models f_t take the `ViT-Base`(a, d), `Swin-Base`(b, e), and `Deit-Base`(c, f) models trained on the `ImageNet` dataset. The samples of $\mathbb{E}_{B\sim\text{Subset of }\mathcal{Z}(\boldsymbol{x}_k)\backslash\mathcal{N}(z)}[\cdots]$ versus $\mathbb{E}_{B\sim\{\mathcal{Z}(\boldsymbol{x}_k)\backslash\mathcal{N}(z),\varnothing\}}[\cdots]$ is plotted in Figure 8. It is observed that the value of $\mathbb{E}_{B\sim\{\mathcal{Z}(\boldsymbol{x}_k)\backslash\mathcal{N}(z),\varnothing\}}[\cdots]$ after the approximation shows positive linear correlation with $\mathbb{E}_{B\sim\text{Subset of }\mathcal{Z}(\boldsymbol{x}_k)\backslash\mathcal{N}(z)}[\cdots]$. This indicates the approximate value $\mathbb{E}_{B\sim\{\mathcal{Z}(\boldsymbol{x}_k)\backslash\mathcal{N}(z),\varnothing\}}[\cdots]$ can take the place of $\mathbb{E}_{B\sim\text{Subset of }\mathcal{Z}(\boldsymbol{x}_k)\backslash\mathcal{N}(z)}[\cdots]$ for the function of attribution.

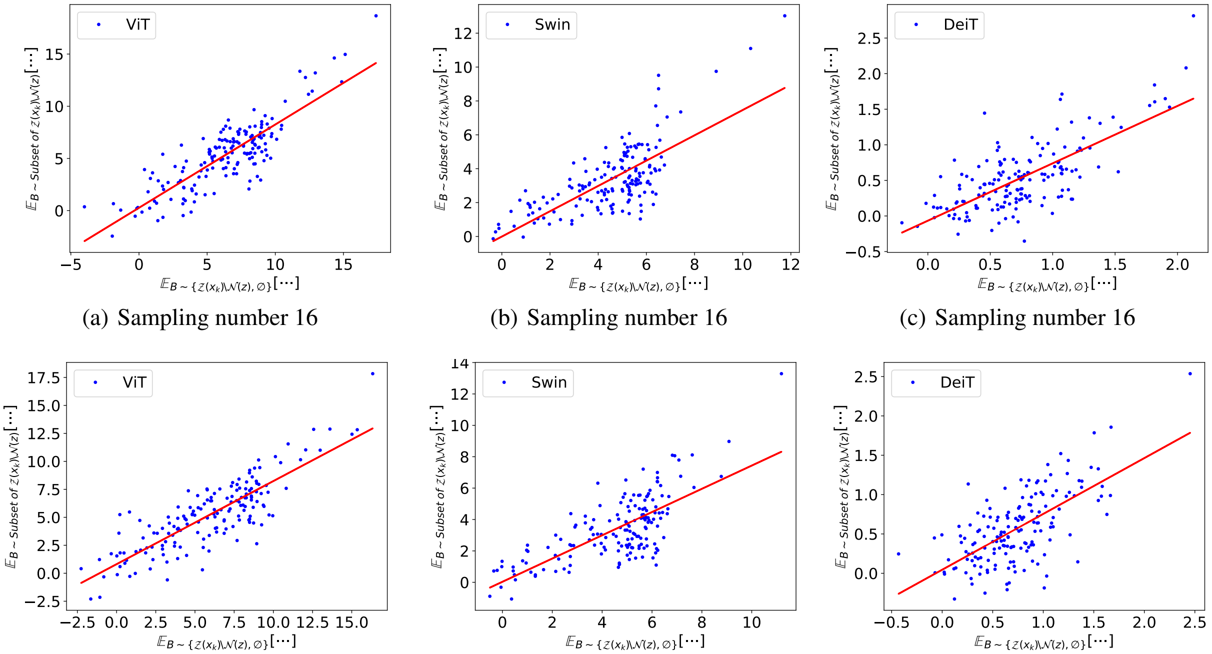
C. Proof of Theorem 1

We prove Theorem 1 in this section.

Theorem 1 (Explanation Error Bound)**.** *Given the classifier $H_t(\bullet)$ of the downstream task, if the output of classifier $H_t(\hat{\mathbf{g}}_{k,z};y)$ and $H_t(\hat{\boldsymbol{h}}_{k,z};y)$ fall within the range of $1-\epsilon\le\frac{H_t(\hat{\mathbf{g}}_{k,z};y)}{H_t(\mathbf{g}_{k,z};y)},\frac{H_t(\boldsymbol{h}_{k,z};y)}{H_t(\hat{\boldsymbol{h}}_{k,z};y)}\le 1+\epsilon$, then, the upper bound of explanation error is given by*

$$\mathbb{E}_{\boldsymbol{x}_k\sim\mathcal{D}_t,y\sim\mathcal{Y}_t,z\sim\mathcal{Z}(\boldsymbol{x}_k)}|\hat{\phi}_{k,y,z}-\phi_{k,y,z}|\le\frac{2\epsilon}{1-\epsilon},\tag{10}$$

where $\hat{\phi}_{k,y,z}$ and $\phi_{k,y,z}$ are given by Equation (8) and (6), respectively; and $\mathcal{D}_t$ denotes the downstream dataset.



(a) Sampling number 16 (b) Sampling number 16 (c) Sampling number 16

(d) Sampling number 32 (e) Sampling number 32 (f) Sampling number 32

Figure 8. $\mathbb{E}_{B\sim\text{Subset of }\mathcal{Z}(\boldsymbol{x}_k)\backslash\mathcal{N}(z),\varnothing}[\cdots]$ versus $\mathbb{E}_{B\sim\{\mathcal{Z}(\boldsymbol{x}_k)\backslash\mathcal{N}(z),\varnothing\}}[\cdots]$, where $\cdots$ is the abbreviation of $-\log f_t(B;\boldsymbol{x}_k,y)+\log f_t(\mathcal{N}(z)\cup B;\boldsymbol{x}_k,y)$; and f_t takes the trained `ViT-Base`(a, d), `Swin-Base`(b, e), and `Deit-Base`(c, f) models on the ImageNet dataset. The sampling number of $B\sim$ Subset of $\mathcal{Z}(\boldsymbol{x}_k)\setminus\mathcal{N}(z)$ is 16 and 32 for Sub-figures (a)-(c) and (d)-(f), respectively.

Proof. To achieve the explanation error bound, we first have the upper bound of $\hat{\phi}_{k,y,z}-\phi_{k,y,z}$ given by

$$\begin{aligned}
\hat{\phi}_{k,y,z}-\phi_{k,y,z} &= \log H_t(\hat{\boldsymbol{g}}_{k,z};y)-\log H_t(\boldsymbol{g}_{k,z};y)+\log H_t(\mathbf{h}_{k,z};y)-\log H_t(\hat{\mathbf{h}}_{k,z};y), && (11)\\
&= \log\frac{H_t(\hat{\boldsymbol{g}}_{k,z};y)}{H_t(\boldsymbol{g}_{k,z};y)}+\log\frac{H_t(\mathbf{h}_{k,z};y)}{H_t(\hat{\mathbf{h}}_{k,z};y)}\le\frac{H_t(\hat{\boldsymbol{g}}_{k,z};y)}{H_t(\boldsymbol{g}_{k,z};y)}-1+\frac{H_t(\mathbf{h}_{k,z};y)}{H_t(\hat{\mathbf{h}}_{k,z};y)}-1, && (12)\\
&\le\frac{H_t(\hat{\boldsymbol{g}}_{k,z};y)}{H_t(\boldsymbol{g}_{k,z};y)}-1+\frac{H_t(\mathbf{h}_{k,z};y)}{H_t(\hat{\mathbf{h}}_{k,z};y)}-1\le\epsilon+\epsilon, && (13)
\end{aligned}$$

Then, we have the lower bound of $\hat{\phi}_{k,y,z}-\phi_{k,y,z}$ as follows,

$$\begin{aligned}
\hat{\phi}_{k,y,z}-\phi_{k,y,z} &= -\log\frac{H_t(\boldsymbol{g}_{k,z};y)}{H_t(\hat{\boldsymbol{g}}_{k,z};y)}-\log\frac{H_t(\hat{\mathbf{h}}_{k,z};y)}{H_t(\mathbf{h}_{k,z};y)}\ge 1-\frac{H_t(\boldsymbol{g}_{k,z};y)}{H_t(\hat{\boldsymbol{g}}_{k,z};y)}+1-\frac{H_t(\hat{\mathbf{h}}_{k,z};y)}{H_t(\mathbf{h}_{k,z};y)}, && (14)\\
&= 2-\left(\frac{H_t(\boldsymbol{g}_{k,z};y)}{H_t(\hat{\boldsymbol{g}}_{k,z};y)}+\frac{H_t(\hat{\mathbf{h}}_{k,z};y)}{H_t(\mathbf{h}_{k,z};y)}\right)\ge 2-\frac{1}{1-\epsilon}-\frac{1}{1-\epsilon}=\frac{-2\epsilon}{1-\epsilon} && (15)
\end{aligned}$$

Combining Equations (10) and (15), we achieve the upper bound of estimation error given by

$$|\hat{\phi}_{k,y,z}-\phi_{k,y,z}|\le\max\left\{2\epsilon,\frac{2\epsilon}{1-\epsilon}\right\}=\frac{2\epsilon}{1-\epsilon}. \quad (16)$$

□

D. Details about the Datasets

We consider the large-scale ImageNet dataset (Deng et al., 2009) for `TVE` pre-training; and the `Cats-vs-dogs` (Elson et al., 2007), `CIFAR-10` (Krizhevsky et al., 2009), and `Imagenette` (Howard, 2019) datasets for the downstream task of explanation. **ImageNet** (Deng et al., 2009): A large scale image dataset which has over one million color images covering 1000 categories, where each image has 224×224 pixels. **Cats-vs-dogs** (Elson et al., 2007): A dataset of cats and dogs images. It has 25000 training instances and 12500 testing instances. **CIFAR-10** (Krizhevsky et al., 2009): An image dataset with 60,000 color images in 10 different classes, where each image has 32×32 pixels. **Imagenette** (Howard, 2019): A benchmark dataset of explainable machine learning for vision models. It contains 10 classes of the images from the Imagenet.

E. Details about Target Models for Downstream Classification.

E.1. Setup of Fine-tuning the Target Models

For downstream classification tasks, we comprehensively consider three architectures of vision transformers as the backbone encoders, including the ViT-Base/Large (Dosovitskiy et al., 2020), Swin-Base/Large (Liu et al., 2021b), Deit-Base (Touvron et al., 2021) transformers. The classification models (to be explained) consist of one of the backbone encoders with ImageNet pre-trained weights and a linear classifier. For the task-specific fine-tuning of target models, we consider two mechanisms: *classifier-tuning* and *full-fine-tuning*. Specifically, the classifier-tuning follows the transfer learning setting (Chilamkurthy, 2017; He et al., 2020; Chen et al., 2020) to freeze the parameters of backbone encoder during the fine-tuning; and the full-fine-tuning updates all parameters during the finetuning. Note that the classifier-tuning can not only be more efficient but also prevent the over-fitting problem on downstream data due to fewer trainable parameters (Sun et al., 2022). We consider the classifier-tuning for most of our experiments including Sections 6.2, 6.4, 6.5, and 6.7; and consider the full-fine-tuning in Section 6.3; while these two mechanisms yield the same result for Section 6.6. The hyper-parameters of task-specific fine-tuning are given in Appendix E.2.

E.2. Hyper-parameter Setting of Fine-tuning the Target Models on Downstream Tasks

The downstream classification models consist of the backbones of ViT-Base/Large, Swin-Base/Large, Deit-Base transformers, and a linear classifier. The hyper-parameters of fine-tuning the classification models on the Cats-vs-dogs, CIFAR-10, and Imagenette datasets are given in Table 2. After the fine-tuning, the classification accuracy on each downstream dataset is given in Table 3.

Table 2. Hyper-parameters of fine-tuning the target model on downstream datasets.

Datasets	Cats-vs-dogs	CIFAR-10	Imagenette
Target backbone	ViT-Base, Swin-Base, and Deit-Base		
Classifier	Linear classifier		
Fine-tuning mechanism	classifier-tuning and full-fine-tuning		
Optimizer	ADAM		
Learning rate	2×10^{-4}		
Mini-batch size	256		
Scheduler	Linear		
Warm-up-ratio	0.05		
Weight-decay	0.05		
Epoch	5		

Table 3. Accuracy of the target model on downstream datasets.

Model Architecture	ViT-Base		Swin-Base		Deit-Base	
Tunable parameters	θ_H	θ_H, θ_G	θ_H	θ_H, θ_G	θ_H	θ_H, θ_G
Cats-vs-dogs	99.6%	99.5%	99.6%	99.7%	99.4%	98.1%
Imagenette	99.3%	99.3%	99.8%	99.7%	99.8%	99.4%
CIFAR-10	92.2%	98.9%	97.0%	98.6%	94.2%	98.1%

F. Details about the Baseline Methods

We consider seven baseline methods for comparison, which include general explanation methods: `LIME` (Ribeiro et al., 2016), `IG` (Sundararajan et al., 2017), `RISE` (Petsiuk et al., 2018), and `DeepLift` (Ancona et al., 2017); Shapley explanation methods: `KernelSHAP` (`KS`) (Lundberg & Lee, 2017), and `GradShap` (Lundberg & Lee, 2017); and DNN-based explainer: `ViT-Shapley` (Covert et al., 2022) in our experiment.

ViT-Shapley: This work adopts vision transformers as the explainer to learn the Shapley value. This work requires task-specific data to train the explainer. **RISE:** RISE randomly perturbs the input, and average all the masks weighted by the perturbed DNN output for the final saliency map. The sampling number takes the default value 50. **IG:** Integrated Gradients estimates the explanation by the integral of the gradients of DNN output with respect to the inputs, along the pathway from specified references to the inputs. **DeepLift:** DeepLift generates the explanation by decomposing DNN output on a specific input by backpropagating the contributions of all neurons in the network to every feature of the input. **KernelSHAP:** KernelSHAP approximates the Shapley value by learning an explainable surrogate (linear) model based on the DNN output of reference input for each feature. The sampling number takes the default value 25 for each instance according to the `captum.ai` (Kokhlikyan et al., 2020). **GradShap:** GradShap estimates the importance features by computing the expectations of gradients by randomly sampling from the distribution of references. **LIME (Ribeiro et al., 2016):** LIME generates the explanation by sampling points around the input instance and using DNN output at these points to learn a surrogate (linear) model. The sampling number takes the default value 25 according to the `captum.ai`. For implementation, we take the IG, DeepLift, and GradShap algorithms on the `captum.ai`, where the `multiply_by_inputs` factor takes false to achieve the local attribution for each instance.

G. Evaluation Metrics

Fidelity-sparsity Curve: We consider the fidelity to evaluate the explanation following existing work (Yang et al., 2022; Chuang et al., 2023b). Specifically, the fidelity evaluates the explanation via *removing the important or trivial patches* from the input instance and *collecting the prediction difference of the target model* f_t. These two perspectives of evluation are formalized into $\mathrm{Fidelity}^+$ and $\mathrm{Fidelity}^-$, respectively. Specifically, provided a subset of patches $\mathcal{S}^* \subseteq \mathcal{Z}(\boldsymbol{x}_k)$ that are important to the target model f_t by an explanation method, the $\mathrm{Fidelity}^+$ and $\mathrm{Fidelity}^-$ evaluates the explanation following

$$\uparrow \mathrm{Fidelity}^+ = \frac{1}{|\mathcal{D}_{\text{task}}|} \sum_{\boldsymbol{x} \in \mathcal{D}_{\text{task}}} f_t(\mathcal{Z}(\boldsymbol{x}_k); \boldsymbol{x}_k, y) - f_t(\mathcal{Z}(\boldsymbol{x}_k) \setminus \mathcal{S}^*; \boldsymbol{x}_k, y),$$

$$\downarrow \mathrm{Fidelity}^- = \frac{1}{|\mathcal{D}_{\text{task}}|} \sum_{\boldsymbol{x} \in \mathcal{D}_{\text{task}}} f_t(\mathcal{Z}(\boldsymbol{x}_k); \boldsymbol{x}_k, y) - f_t(\mathcal{S}^*; \boldsymbol{x}_k, y).$$

Higher $\mathrm{Fidelity}^+$ indicates a better explanation for prediction y, since the truly important patches of image $\boldsymbol{x}_k$ have been removed, leading to a significant difference of model prediction. Moreover, lower $\mathrm{Fidelity}^-$ implies a better explanation for prediction y, since the truly important patches have been preserved in $\mathcal{S}^*$ to keep the prediction similar to the original one. The fidelity should be compared at the same level of sparsity $|\mathcal{S}^*|/|\mathcal{U}|$. Consequently, we consider the evaluation of fidelity versus the sparsity in most cases.

Fidelity-sparsity-AUC Metric To streamline our evaluation, we simplify the assessment of fidelity-sparsity curves by calculating its Area Under the Curve (AUC) over the sparsity from zero to one, which aligns with the average fidelity value. In the last paragraph, we have shown that higher $\mathrm{Fidelity}^+$ and lower $\mathrm{Fidelity}^-$ at the same level of sparsity indicate more faithful explanation. To streamline the evaluation, the assessment of fidelity-sparsity curves can be simplified into its Area Under the Curve (AUC) over the sparsity from zero to one, as shown in Figures 9 (a) and (b). The Fidelity-sparsity-AUC aligns with the average fidelity value. Specifically, a higher $\mathrm{Fidelity}^+$-sparsity-AUC (↑) indicates better $\mathrm{Fidelity}^+$ performance across most sparsity levels, reflecting a more faithful explanation. Similarly, a lower $\mathrm{Fidelity}^-$-sparsity-AUC signifies a more faithful explanation For the given example in Figures 9 (a) and (b), explanation A is more faithful than B.

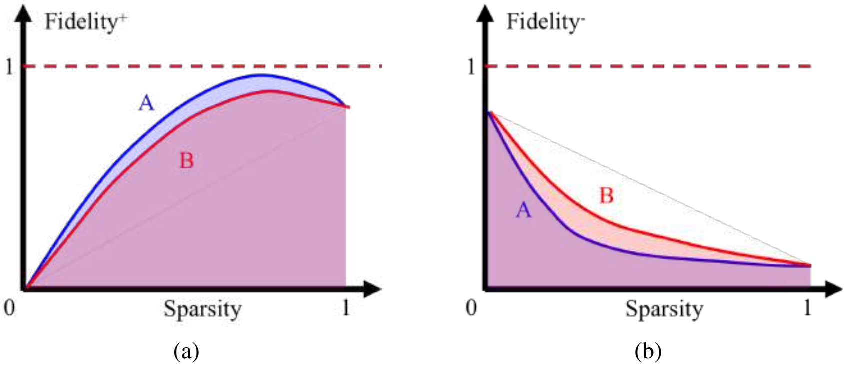


Figure 9. Illustration of Fidelity^{+}-sparsity-AUC (a) and Fidelity^{-}-sparsity-AUC (b)

H. Implementation Details about `TVE`

Architecture of Generic Explainer. The architecture of the transferable explainer is shown in Figure 10 (a). Specifically, the explainer takes the `Mask-AutoEncoder-Base` (He et al., 2022) for the backbone. As shown in Figure 11, the `Mask-AutoEncoder-Base` architecture is a pipeline of 12-layer ViT encoder and 8-layer ViT decoder, where the input and output shape are $[\mathrm{BS}, 3, 224, 224]$ and $[\mathrm{BS}, \mathrm{P}, \mathrm{P}, 768]$, respectively. More details about the `Mask-AutoEncoder-Base` can be referred to its source code[3].

Since the output shape of the `Mask-AutoEncoder-Base` is $[\mathrm{BS}, \mathrm{P} \times \mathrm{P}, 768]$ is not matched with that of the meta-attribution $[\mathrm{BS}, \mathrm{P} \times \mathrm{P}, \mathrm{D}]$, where BS denotes the mini-batch size. We adopt $n\times$ FFN-layers as explainer heads to map the output tensor of the `Mask-AutoEncoder-Base` into meta-attribution, where we found $n = 17$ enables the expalainer to have strong generalization ability to explain various downstream tasks. The structure of an explainer head is given in Figure 10 (b). The first explainer head does not have the skip connection due to the mismatch of tensor shapes. The last explainer head does not have the GELU activation.

Backbone Encoder. We comprehensively consider three backbone encoders for during the pre-training of transferable explainer, including the `ViT-Base/Large`, `Swin-Base/Large`, `Deit-Base` transformers. Their pre-trained weights are loaded from the HuggingFace library (Wolf et al., 2020b). The hyper-parameter setting of `TVE` pre-training is given in Table 4.

Table 4. Hyper-parameters of `TVE` pre-training on the ImageNet dataset.

Target Encoder	`ViT-Base`	`Swin-Base`	`DeiT-Base`
Explainer Architecture		Figure 10	
Pixel # per image W × W		224×224	
Patch # per image P × P		14×14	
Pixel # per patch C × C		16×16	
Shape of $\boldsymbol{g}_k$ and $\mathbf{h}_k$	$14 \times 14 \times 768$	$14 \times 14 \times 1024$	$14 \times 14 \times 768$
Optimizer		ADAM	
Learning rate		1×10^{-3}	
Mini-batch size		64 per GPU × 4 GPUs	
Scheduler		CosineAnnealingLR	
Warm-up-ratio		0.05	
Weight-decay		0.05	
Training steps		2×10^{5}	
Neighbor patches		0-, 1-, 2-hop neighbor patches	

[3] https://github.com/huggingface/transformers/blob/main/src/transformers/models/vit_mae/modeling_vit_mae.py

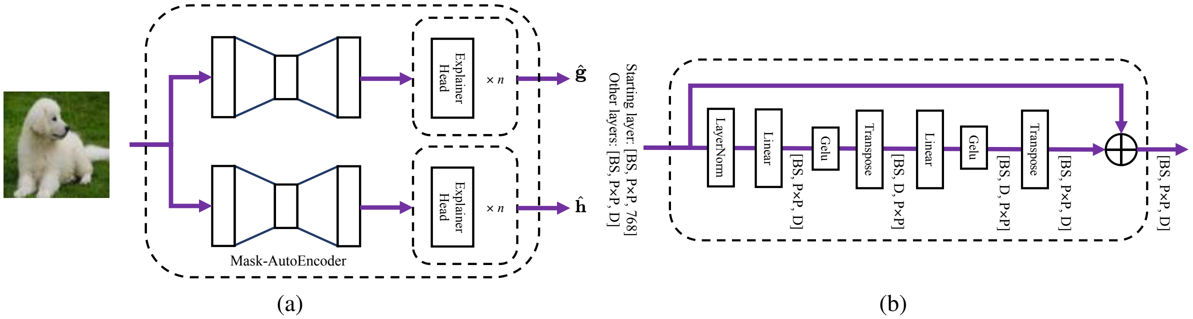


Figure 10. Transferable explainer architecture. (a) Explainer architecture. (b) FFN-layers for the explainer head.

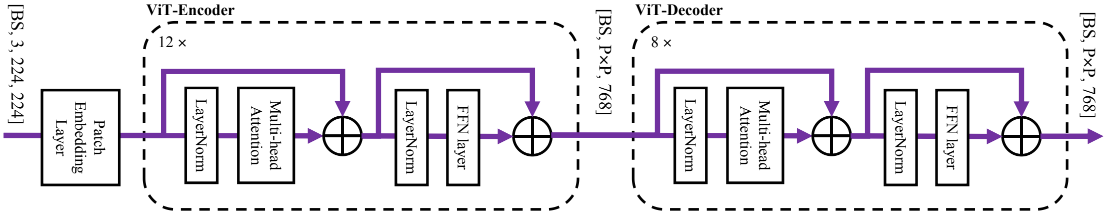
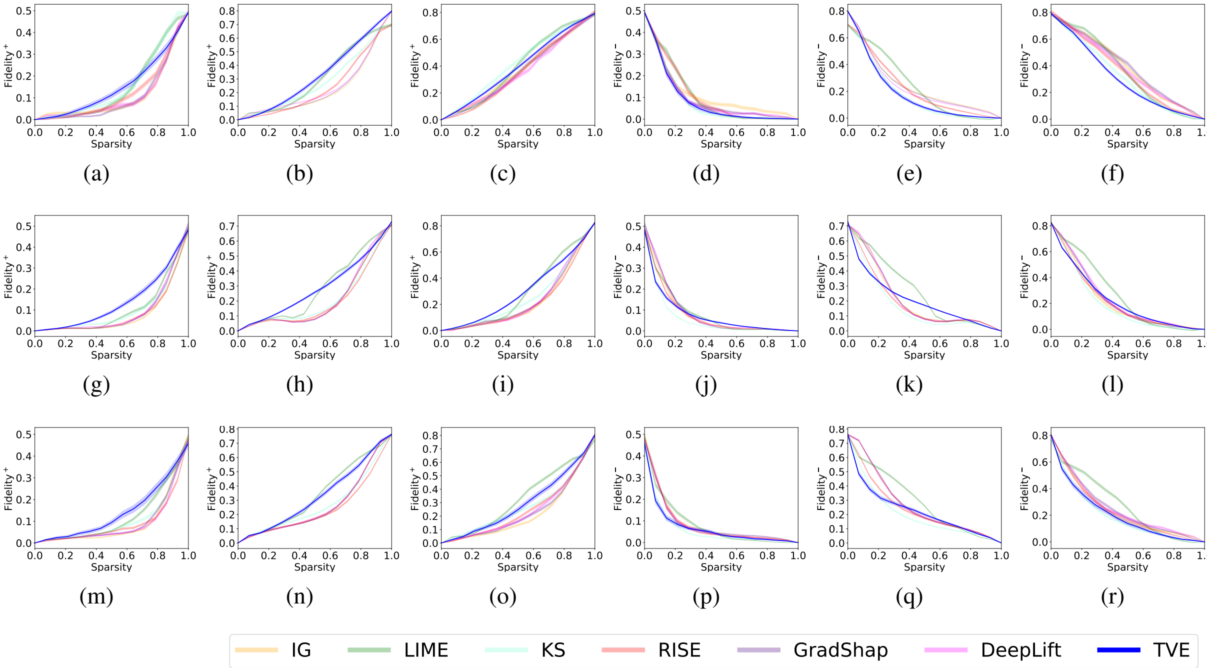


Figure 11. Structure of Mask-Autoencoder.



(a) (b) (c) (d) (e) (f)

(g) (h) (i) (j) (k) (l)

(m) (n) (o) (p) (q) (r)

Figure 12. Fidelity$^+$-sparsity curve for explaining ViT-Base on Cats-vs-dogs (a), Imagenette (a), and CIFAR-10 (c). Fidelity$^-$-sparsity curve of ViT-Base on Cats-vs-dogs (d), Imagenette (e), and CIFAR-10 (f). Fidelity$^+$-sparsity curve of Swin-Base on Cats-vs-dogs (g), Imagenette (h), and CIFAR-10 (i). Fidelity$^-$-sparsity curve of Swin-Base on Cats-vs-dogs (j), Imagenette (k), and CIFAR-10 (l). Fidelity$^+$-sparsity curve of Deit-Base on Cats-vs-dogs (m), Imagenette (n), and CIFAR-10 (o). Fidelity$^-$-sparsity curve of Deit-Base on Cats-vs-dogs (p), Imagenette (q), and CIFAR-10 (r).

I. Fidelity-Sparsity Curve of Section 6.2

We show the fidelity-sparsity curve for explaining ViT-Base, Swin-Base, and Deit-Base on the Cats-vs-dogs, Imagenette, and CIFAR-10 datasets in Figures 12 (a)-(r). It is observed that TVE consistently exhibits promising performance in terms of both Fidelity$^+$(↑) and Fidelity$^-$(↓), surpassing the majority of baseline methods. This indicates TVE's ability to faithfully explain various downstream tasks.

J. Fidelity in Explaining ResNet-50

We conducted experiments on the ImageNet dataset, with ResNet-50 as the explained model. TVE is pre-trained using the ViT-Base as the backbone encoder, following the hyper-parameter settings outlined in Appendix H. The fidelity-sparse-AUC results are presented in Table 5. It is observed that TVE outperforms baseline methods in explaining the ResNet-50, highlighting the general effectiveness of TVE.

Table 5. Fidelity-sparsity AUC of TVE of explaining the ResNet-50 on the ImageNet dataset.

Method	Fidelity$^+$-Sparsity-AUC(↑)	Fidelity$^-$-Sparsity-AUC(↑)	Throughput
LIME	0.35	0.29	4.7 image/s
RISE	0.35	0.34	6.2 image/s
TVE	0.35	0.21	101.7 image/s

K. Fidelity in Explaining ViT-Large

We conducted experiments deploying the TVE pre-trained explainer to both ViT-Base and ViT-Large models on the ImageNet dataset. The fidelity and throughput results are presented in Table 6. Notably, TVE demonstrates a negligible decrease in throughput, without degradation in fidelity, as the size of the target model increases from ViT-Base to ViT-Large.

Table 6. Fidelity-sparsity AUC of TVE of explaining the ViT-Large on the ImageNet dataset.

Target Model	Fidelity$^+$-Sparsity-AUC(↑)	Fidelity$^-$-Sparsity-AUC(↓)	Throughput(↑)
ViT-Base	0.40	0.27	101 image/s
ViT-Large	0.41	0.28	96 image/s

L. Fidelity on CIFAR-100 Dataset

We conduct experiments of explaining the ViT-Base model on the CIFAR-100 dataset, following the hyper-parameter settings in Appendix H. The fidelity-sparse-AUC results are presented in Table 7. It is observed that TVE outperforms baseline methods on the CIFAR-100 dataset, highlighting the general effectiveness of TVE.

Table 7. Fidelity-sparsity AUC of TVE on the CIFAR-100 dataset.

Method	Fidelity$^+$-Sparsity-AUC(↑)	Fidelity$^-$-Sparsity-AUC(↓)
LIME	0.355	0.337
KS	0.364	0.321
IG	0.352	0.372
RISE	0.354	0.348
TVE	0.386	0.311

M. Fidelity on MURA Dataset

We conduct experiments of explaining the ViT-Base model on the MURA dataset, following the hyper-parameter settings in Appendix H. The MURA is a dataset of musculoskeletal radiographs containing 40561 images from 14863 studies, which is out of the distribution of the ImageNet dataset. The fidelity-sparse-AUC results are presented in Table 8. TVE demonstrates comparable fidelity to baseline methods, achieving significantly faster results while maintaining competitiveness. This indicates that even for the downstream dataset not within the pre-training distribution, like the MURA dataset, TVE can effectively perform after being fully trained on the downstream dataset.

Table 8. Fidelity-sparsity AUC of TVE on the MURA dataset.

Method	Fidelity^{+}-Sparsity-AUC(↑)	Fidelity^{-}-Sparsity-AUC(↓)	Throughput(↑)
LIME	0.176	0.095	4.5 image/s
IG	0.154	0.122	4.7 image/s
RISE	0.142	0.1231	6.3 image/s
GradSHAP	0.160	0.126	26.4 image/s
DeepLift	0.146	0.086	54.9 image/s
TVE	0.174	0.099	101.9 image/s

N. Computational Infrastructure

The computational infrastructure information is given in Table 9.

Table 9. Computing infrastructure for the experiments.

Device Attribute	Value
Computing Infrastructure	GPU
GPU Model	NVIDIA-A5000
GPU Memory	24564MB
GPU Number	8
CUDA Version	12.1
CPU Memory	512GB

O. More Case Studies

We give more explanation heatmaps of `ViT-Base` on the `ImageNet` dataset in Figure 13, which are generated by `TVE`.

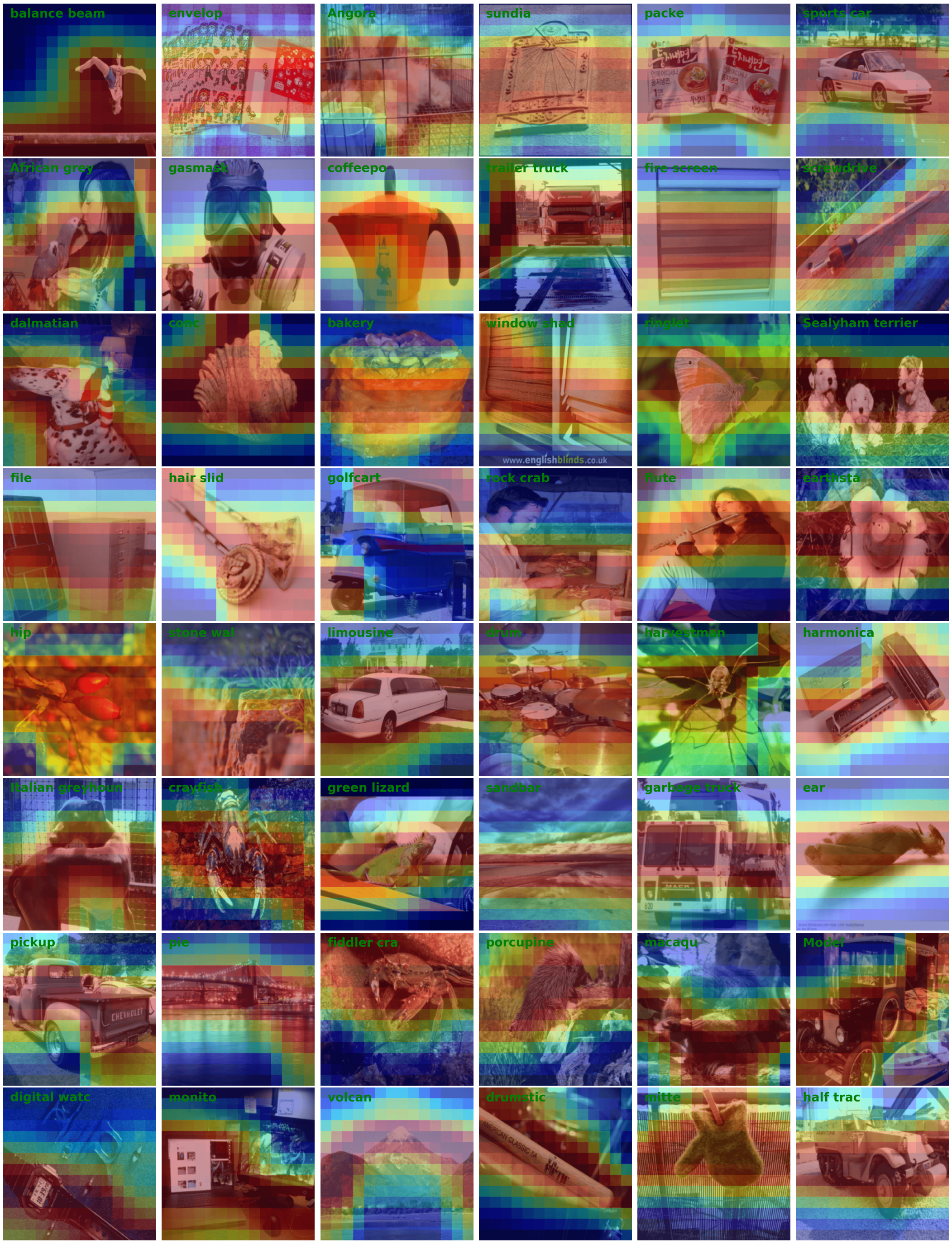


Figure 13. Explanation heatmaps of `ViT-Base` on the `ImageNet` dataset.