Transferring Visual Explainability from Self-Explaining to Prediction-Only Vision Transformers via Task Arithmetic

Yuya Yoshikawa
STAIR Lab, Chiba Institute of Technology
yoshikawa@stair.center

Ryotaro Shimizu
ZOZO Research
ryotaro.shimizu@zozo.com

Takahiro Kawashima
ZOZO Research
takahiro.kawashima@zozo.com

Yuki Saito
ZOZO Research
yuki.saito@zozo.com

Abstract

In image classification scenarios where both prediction and explanation efficiency are required, self-explaining models that perform both tasks in a single inference are effective. However, for users who already have prediction-only models, training a new self-explaining model from scratch is costly in both labeling and computation. This study proposes a method to transfer the visual explanation capability of self-explaining Vision Transformer (ViT) models learned in a source domain to prediction-only, zero-shot-capable ViT models based on vision-language models (VLMs) in a target domain via task arithmetic, without target-side explanation supervision. The proposed method endows target models with explanation capability by adding an **explainability vector** induced by explanation supervision in the source domain based on the task arithmetic framework. Experiments on 10 diverse target datasets show that a single explainability vector learned on ImageNet-1k augmented with patch-level explanation supervision transfers consistently, improving explanation quality while largely preserving classification accuracy. Within-model faithfulness evaluation and comparisons with control vectors (e.g., random vectors of the same norm) show that the transferred explanations are faithful to the model's own predictions and that the effect is specific to the explainability vector. Beyond the transfer itself, we further investigate whether transfer success can be anticipated *a priori* from model-internal features alone, and find that, for a fixed source, a small set of such features carries nontrivial signal about the transfer outcome on held-out targets. Code is available at https://github.com/yuyay/explainability_vector.

1 Introduction

Visual explainability, which provides human-understandable explanations for a prediction model's outputs, is important for various applications, such as medical image diagnosis [44], autonomous driving [5], manufacturing [1], cybersecurity [6], and model debugging [4].

Although explanations provide insights to users, generating them often costs far more computation than prediction alone. When a trained model is already available, visual explanations can be generated using post-hoc explanation methods [69], such as SHAP [41] and Grad-CAM [51]. However, SHAP typically requires hundreds of model inferences per image, and Grad-CAM requires a backward computation of gradients with respect to intermediate feature maps. Such computational challenges can be addressed by using *self-explaining models*, which are designed to perform both prediction and explanation in a single inference [3, 31, 67, 68]. Although self-explaining models are very efficient

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

in the inference phase, their training can be costly, because it requires ground-truth (GT) explanation data for user-desired explanations [50, 64] or numerous model forward computations to learn these explanations [58].

The goal of this study is to efficiently equip prediction-only, zero-shot-capable Vision Transformer (ViT) classifiers based on vision-language models (VLMs), such as CLIP image classifiers, with visual explainability without paying the high training cost of building a self-explaining model in the target domain. To this end, we focus on *task arithmetic* [28], which enables model editing using *task vectors* that represent the model's capabilities learned for different tasks across different domains. Task vectors can also express *task analogies*, *"A is to B as C is to D,"* with A and B in a source domain and C and D in a target domain. If the analogy holds, the task vector for D follows from those for A, B, and C without training the model on D.

Inspired by the idea, we propose a method to transfer the visual explainability of self-explaining ViTs learned in a source domain to such prediction-only models in a target domain. Our self-explaining model extends VLM-based image classifiers such as CLIP [49] by producing patch-level attributions from patch and class-label text embeddings alongside the class logits (Figure 1). Because their classification head is built from frozen text embeddings of class names, all models finetuned from the same VLM checkpoint share one parameter space across label sets. This is the standard finetuning setting for such models [60, 28]. We train two source-domain models, with and without explanation supervision, and define the **explainability vector** as their parameter difference. Adding this vector to a target-domain prediction-only model transfers the explanation capability, instantiating the task analogy *"prediction-only is to prediction-with-explanation in the source domain as prediction-only is to prediction-with-explanation in the target domain."*

Our main contributions are as follows:

- We propose the first task-arithmetic approach to transfer visual explainability from self-explaining to prediction-only VLM-based ViTs across domains without target-side explanation supervision, by defining an explainability vector as a task vector induced by explanation supervision in the source domain (Section 4.3).
- We empirically show that a single explainability vector learned on ImageNet-1k dataset augmented with patch-level explanation supervision transfers consistently to 10 diverse target datasets, improving explanation quality while largely preserving classification accuracy, and that the transferred explanations are faithful to the model itself and specific to the explainability vector (Section 5.1).
- We additionally find that the success of such transfer is predictable to a meaningful degree. When the source is fixed and targets are held out, a small set of model-internal features computed without target-side explanation supervision yields held-out Pearson $r > 0.7$ on multiple explanation-quality metrics. This provides initial evidence that transferability is a measurable property of the source–target pair (Sections 4.4, 5.2).

2 Related Work

Self-Explaining Vision Models To improve the faithfulness and computational efficiency of explanations, various self-explaining models that are capable of performing both prediction and explanation have been proposed [26, 31]. In ViT-based models, the attention map between the class token and image patches can be visualized as the importance of each image patch [70, 48]. Since some studies have raised concerns about the validity of attention as an explanation [52, 30, 9], several studies have proposed methods for training models with constraints that explicitly encourage attention to be more aligned with human-interpretable explanations [70, 48]. As with our study, several studies have also addressed training self-explaining models to directly produce Shapley value-based explanations [58, 2]. These approaches encourage the explanations to approximate Shapley values through the design of the objective function or network architecture, without requiring precomputed GT Shapley value explanations. Similar approaches have been proposed in the research on amortized explainers [13], which learn the behavior of post-hoc explainers for a trained prediction model in advance and generate only explanations efficiently in the inference phase. Similar to ours, Covert et al. [16] proposed a method that trains a ViT to output patch-level attributions based on Shapley values from patch tokens. While this approach can efficiently train models to produce Shapley value explanations, it is difficult to generalize it to explanations other than Shapley values due to its specialization.

Post-hoc Explainers for ViTs and CLIP Post-hoc explainers designed for ViTs trace relevance through the attention and token transformations of the network, e.g., relevancy propagation [8, 9] and TokenTM [61]. Several methods further improve CLIP-specific interpretability by analyzing or modifying its internal representations: attention surgery [40], gradient-based explanations [72], layer/head decomposition [20], and adversarial finetuning [22]. These works analyze a trained model or modify it to improve its interpretability, whereas our method transfers an attribution-producing capability via task arithmetic so that the model itself produces attributions in its forward pass. Their attribution maps can also serve as the explanation supervision for the GT attributions (Section 4.2).

Visual Explainability Transfer Some studies have proposed methods for transferring visual explainability to other domains. Transferable Vision Explainer (TVE) [57] pretrains an encoder on a large-scale dataset like ImageNet to output meta-attribute vectors from images. In the inference phase, it uses a combination of this encoder and a pretrained classification head for the target task to output patch contributions. TVE leaves the target classifier untouched and produces attributions with a separate explainer network, whereas our method edits the finetuned target so that the target itself produces explanations. Li et al. [39] showed that transferring attention patterns, which are often read as explanations, from a teacher to a student ViT recovers most of the benefit of weight finetuning, although the gap widens when the pretraining and downstream data distributions differ.

3 Task Arithmetic

Our approach builds on the task arithmetic framework [28]. Task arithmetic edits neural network models by manipulating differences in their parameter spaces. The central idea is that the effect of learning a specific capability (or task) can be represented as a vector in the parameter space, and these vectors for different tasks can be algebraically combined to transfer or compose capabilities across models. The validity of task arithmetic is supported by the observations that aggregating models by adding multiple model parameters can improve model performance [29, 59, 60, 65, 66]. Beyond this validity, task arithmetic has been used to compose or remove skills such as mathematical reasoning [11], fairness [23], unlearning [46], and bias mitigation [54], but its capability for transferring visual explainability remains unexplored.

Formally, let $\boldsymbol{\theta}_{\text{base}} \in \mathbb{R}^D$ denote the parameters of a base model where D is the dimensionality of the model parameters, and let $\boldsymbol{\theta}_{\text{ft}} \in \mathbb{R}^D$ denote the parameters of the same model after finetuning on a specific task from the base parameters $\boldsymbol{\theta}_{\text{base}}$. The difference $\boldsymbol{\tau} = \boldsymbol{\theta}_{\text{ft}} - \boldsymbol{\theta}_{\text{base}}$ is called the *task vector* for that task, which is expected to contain the knowledge and capabilities necessary to perform the task. The addition operation of task vectors allows us to construct a model that is capable of performing multiple tasks simultaneously, similar to multi-task learning. Specifically, given J task vectors $\{\boldsymbol{\tau}_j\}_{j=1}^J$, where j is the task index, a new model composed of capabilities from these tasks can be constructed as

$$\boldsymbol{\theta} = \boldsymbol{\theta}_{\text{base}} + \lambda \sum_{j=1}^{J} \boldsymbol{\tau}_j, \tag{1}$$

where $\lambda \in \mathbb{R}$ is a scaling factor controlling the extent to which the capabilities are added (if $\lambda > 0$) or removed (if $\lambda < 0$). A single λ is used in (1) for simplicity. A separate coefficient λ_j per task vector is a natural extension within the same framework (cf. [65]), and our transfer rule in Section 4.3 uses separate coefficients for the classification and explainability vectors.

For a more complicated arithmetic operation that we are interested in, analogical relationships between tasks can be considered. Suppose we have models finetuned for tasks A and B in the same domain (referred to as the *source domain* for convenience), and for task C in another domain (referred to as the *target domain* for convenience). Our aim is to obtain model parameters $\boldsymbol{\theta}$ that achieve a new task D in the target domain. If the relationship between A and B is analogous to that between C and D, we can estimate the parameters for task D using the following analogy:

$$\boldsymbol{\tau}_{\mathsf{A}} - \boldsymbol{\tau}_{\mathsf{B}} \cong \boldsymbol{\tau}_{\mathsf{C}} - \boldsymbol{\tau}_{\mathsf{D}} \quad \rightarrow \quad \boldsymbol{\tau}_{\mathsf{D}} \cong \boldsymbol{\tau}_{\mathsf{C}} - \boldsymbol{\tau}_{\mathsf{A}} + \boldsymbol{\tau}_{\mathsf{B}}. \tag{2}$$

Then, we can endow the model with the capability for task D by adding the task vector $\boldsymbol{\tau}_{\mathrm{D}}$ to the base parameters $\boldsymbol{\theta}_{\text{base}}$ as follows:

$$\boldsymbol{\theta} = \boldsymbol{\theta}_{\text{base}} + \lambda \boldsymbol{\tau}_{\mathsf{D}}. \tag{3}$$

This property transfers learned capabilities from the source to the target domain even when supervision for task D is unavailable there. The next section applies it to visual explainability.

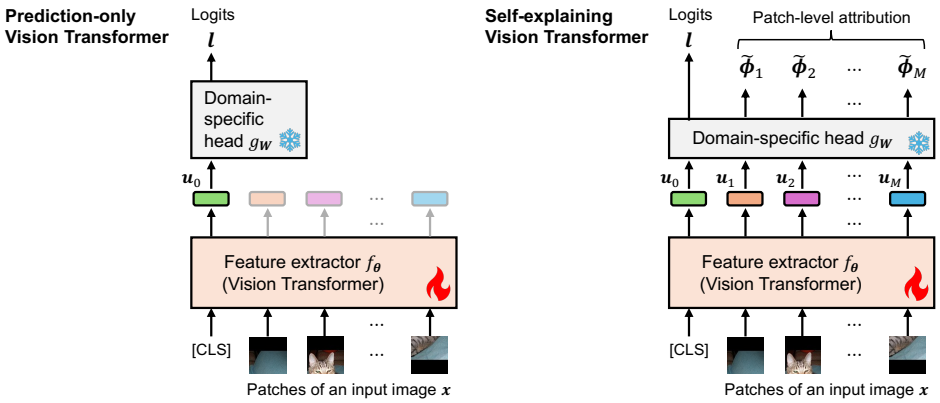


Figure 1: Model architectures of prediction-only ViT (left) and our self-explaining ViT (right). Gray boxes represent functions with fixed parameters. All the learnable parameters $\boldsymbol{\theta}$ are only in the feature extractor (filled in orange).

4 Proposed Method

4.1 Self-Explaining Vision Transformer Model

The proposed method requires a self-explaining image classifier that predicts the logits of class labels and infers patch-level attributions corresponding to the contributions of the patches for each class. As a requirement for the model to utilize task arithmetic, it is necessary that the model is represented by parameters $\boldsymbol{\theta} \in \mathbb{R}^D$ even if the source and target label sets are different. In other words, the model should be capable of zero-shot classification. To satisfy the requirement, our self-explaining model is designed as shown in Figure 1, which is an extension of "prediction-only" ViT image classifiers [49]. It consists of a feature extractor and a domain-specific head. We denote the input space as $\mathcal{X}$, and the number of patches in an input image as M. The feature extractor $f_{\boldsymbol{\theta}}$ is a neural network with the ViT architecture. It takes M patches of an input image $\boldsymbol{x} \in \mathcal{X}$ and a special token $[\texttt{CLS}]$ as input and outputs $M+1$ K-dimensional feature vectors. The domain-specific head is a linear function $g_{\boldsymbol{W}} : \mathbb{R}^K \to \mathbb{R}^C$ with weights $\boldsymbol{W} \in \mathbb{R}^{K \times C}$ where C is the number of classes. Let $\boldsymbol{u}_0 \in \mathbb{R}^K$ denote the output feature vector for $[\texttt{CLS}]$ and $\boldsymbol{u}_m \in \mathbb{R}^K$ that for the mth patch, where $m \in \{1, 2, \ldots, M\}$ is a patch index. The same head $g_{\boldsymbol{W}}$ is applied to every token. Applied to $\boldsymbol{u}_0$, it yields the class logits $\boldsymbol{l} = g_{\boldsymbol{W}}(\boldsymbol{u}_0) = [l_1, l_2, \ldots, l_C]^\top \in \mathbb{R}^C$, exactly as in the prediction-only model. Applied to $\boldsymbol{u}_m$, it yields the unnormalized patch-level attributions $\tilde{\boldsymbol{\phi}}_m = g_{\boldsymbol{W}}(\boldsymbol{u}_m) = [\tilde{\phi}_{m1}, \tilde{\phi}_{m2}, \ldots, \tilde{\phi}_{mC}]^\top \in \mathbb{R}^C$, whose cth entry is the contribution of the mth patch to the cth class. Stacking the M patch vectors as columns gives the attribution matrix $\tilde{\boldsymbol{\Phi}} = [\tilde{\boldsymbol{\phi}}_1, \tilde{\boldsymbol{\phi}}_2, \ldots, \tilde{\boldsymbol{\phi}}_M] \in \mathbb{R}^{C \times M}$, whose rows index classes and whose columns index patches. Its cth row $\tilde{\boldsymbol{\Phi}}_c \in \mathbb{R}^M$ is the patch-level explanation of the decision for the cth class. $\boldsymbol{W}$ is constructed from the class-name text embeddings produced by the text encoder of the VLM[1], i.e., $\boldsymbol{W}_{\cdot,c} \in \mathbb{R}^K$ is the text embedding associated with the cth class. In our model, $\boldsymbol{W}$ is fixed during training and is therefore not included in the model parameters. Our model reuses the VLM's pretrained parameters as the base parameters $\boldsymbol{\theta}_{\text{base}}$. Note that they do not have the capability to explain the classification results because the VLM is not optimized for explanation.

Remarks The key difference between our self-explaining and prediction-only ViTs is that the domain-specific head $g_{\boldsymbol{W}}$ also produces the patch-level attributions. This shared-head constraint does not degrade classification or explanation quality compared with a separate attribution head (Appendix A). Although the two outputs are produced in parallel, head-level circuit analysis shows that classification and explanation share most of their backbone computation, with specialization emerging only in the final layers (Appendix B).

4.2 Learning Classification Capability and Explainability

We acquire explainability on a source domain by finetuning our self-explaining model in a supervised manner. First, we construct a source dataset, a labeled image dataset with explanation supervision

[1]For example, if the class name is "cat," we use the average of text embeddings generated by the text encoder for multiple input texts, such as "a photo of a cat," "art of the cat," and so on.

$\mathcal{D}^{\mathrm{S}} = \{(\boldsymbol{x}^{(n)}, y^{(n)}, \boldsymbol{\phi}^{(n)})\}_{n=1}^{N^{\mathrm{S}}}$ where $\boldsymbol{x}^{(n)} \in \mathcal{X}$ is the nth input image, $y^{(n)} \in \{1, 2, \ldots, C^{\mathrm{S}}\}$ is the GT class label of the input $\boldsymbol{x}^{(n)}$, and $\boldsymbol{\phi}^{(n)} \in \mathbb{R}^M$ is the GT patch-level attributions given $\boldsymbol{x}^{(n)}$ and $y^{(n)}$. We can design the GT patch-level attributions flexibly according to users' purposes and applications. They are typically given by human annotators or by computing a post-hoc explanation method, such as SHAP [41, 53] and Grad-CAM [51]. Since obtaining such attributions may be highly expensive, we assume that they are only available in the source domain.

Initialized at the base parameters $\boldsymbol{\theta}_{\mathrm{base}}$, we learn the model parameters by solving the following optimization problem:

$$\boldsymbol{\theta}^{\mathrm{S}}_{\mathrm{ft}\star} = \arg\min_{\boldsymbol{\theta}} \ \alpha \mathcal{L}_{\mathrm{cls}}(\boldsymbol{\theta}; \mathcal{D}^{\mathrm{S}}) + (1-\alpha)\mathcal{L}_{\mathrm{exp}}(\boldsymbol{\theta}; \mathcal{D}^{\mathrm{S}}), \tag{4}$$

where $\mathcal{L}_{\mathrm{cls}}$ is the cross-entropy loss between the logits and the GT class label. The explanation loss $\mathcal{L}_{\mathrm{exp}}$ is defined as

$$\mathcal{L}_{\mathrm{exp}}(\boldsymbol{\theta}; \mathcal{D}^{\mathrm{S}}) = \frac{1}{M|\mathcal{D}^{\mathrm{S}}|} \sum_{n=1}^{|\mathcal{D}^{\mathrm{S}}|} \|\mathsf{Norm}(\tilde{\boldsymbol{\Phi}}^{(n)}_{y^{(n)}}) - \mathsf{Norm}(\boldsymbol{\phi}^{(n)})\|_2^2, \tag{5}$$

where $\tilde{\boldsymbol{\Phi}}^{(n)}_{y^{(n)}} \in \mathbb{R}^M$ is the $y^{(n)}$th row of $\tilde{\boldsymbol{\Phi}}$, which is the predicted patch-level attributions for the input $\boldsymbol{x}^{(n)}$ when it is classified as the $y^{(n)}$th class, and $\mathsf{Norm}(\boldsymbol{v})$ is a normalization function such that $\mathsf{Norm}(\boldsymbol{v}) = (\boldsymbol{v} - \mu_{\boldsymbol{v}})/\sigma_{\boldsymbol{v}}$ where $\mu_{\boldsymbol{v}}$ and $\sigma_{\boldsymbol{v}}$ are the mean and standard deviation of $\boldsymbol{v}$, respectively. The hyperparameter $\alpha \in [0, 1]$ balances the two losses, $\mathcal{L}_{\mathrm{cls}}$ and $\mathcal{L}_{\mathrm{exp}}$. When $\alpha = 1$, the learned parameters are identical to the finetuned parameters of the prediction-only VLM-based image classifier, $\boldsymbol{\theta}^{\mathrm{S}}_{\mathrm{ft}}$, where the subscript 'ft' stands for finetuned only for classification. The case of $\alpha = 0$ results in amortized explainers [16, 64] where the learned parameters have only the explanatory capability. What we want are the model parameters $\boldsymbol{\theta}^{\mathrm{S}}_{\mathrm{ft}\star}$ learned with $0 < \alpha < 1$ that have both the classification capability (denoted by 'ft') and the explainability (denoted by '$\star$') in the source domain.

4.3 Transferring Visual Explainability

For the target domain, we assume that a prediction-only model $\boldsymbol{\theta}^{\mathrm{T}}_{\mathrm{ft}}$, obtained by minimizing $\mathcal{L}_{\mathrm{cls}}$ on labeled target images, is available, but GT patch-level attributions are not. Therefore, the self-explaining model in the target domain, $\boldsymbol{\theta}^{\mathrm{T}}_{\mathrm{ft}\star}$, cannot be trained directly. Our goal is to obtain $\tilde{\boldsymbol{\theta}}^{\mathrm{T}}_{\mathrm{ft}\star}$, a good approximation of $\boldsymbol{\theta}^{\mathrm{T}}_{\mathrm{ft}\star}$, by transferring the explainability encoded in the source-domain model $\boldsymbol{\theta}^{\mathrm{S}}_{\mathrm{ft}\star}$ to $\boldsymbol{\theta}^{\mathrm{T}}_{\mathrm{ft}}$ without any additional training.

To this end, we utilize the task arithmetic framework introduced in Section 3. Based on the task analogy (2), the task vectors obtained from the parameters trained on the source and target domains are expected to have the following relationship:

$$\boldsymbol{\tau}^{\mathrm{S}}_{\mathrm{ft}\star} - \boldsymbol{\tau}^{\mathrm{S}}_{\mathrm{ft}} \approxeq \boldsymbol{\tau}^{\mathrm{T}}_{\mathrm{ft}\star} - \boldsymbol{\tau}^{\mathrm{T}}_{\mathrm{ft}}, \tag{6}$$

where $\boldsymbol{\tau}^{\mathrm{S}}_{\mathrm{ft}\star} = \boldsymbol{\theta}^{\mathrm{S}}_{\mathrm{ft}\star} - \boldsymbol{\theta}_{\mathrm{base}}$ and $\boldsymbol{\tau}^{\mathrm{T}}_{\mathrm{ft}\star} = \boldsymbol{\theta}^{\mathrm{T}}_{\mathrm{ft}\star} - \boldsymbol{\theta}_{\mathrm{base}}$ are the task vectors representing both the classification capability and explainability for the source and target domains, respectively. $\boldsymbol{\tau}^{\mathrm{S}}_{\mathrm{ft}} = \boldsymbol{\theta}^{\mathrm{S}}_{\mathrm{ft}} - \boldsymbol{\theta}_{\mathrm{base}}$ and $\boldsymbol{\tau}^{\mathrm{T}}_{\mathrm{ft}} = \boldsymbol{\theta}^{\mathrm{T}}_{\mathrm{ft}} - \boldsymbol{\theta}_{\mathrm{base}}$ are the task vectors representing only the classification capability for the source and target domains, respectively. The equality in (6) holds if and only if $\boldsymbol{\tau}^{\mathrm{S}}_{\star} = \boldsymbol{\tau}^{\mathrm{T}}_{\star}$, where $\boldsymbol{\tau}^{\mathrm{S}}_{\star} = \boldsymbol{\theta}^{\mathrm{S}}_{\mathrm{ft}\star} - \boldsymbol{\theta}^{\mathrm{S}}_{\mathrm{ft}}$ and $\boldsymbol{\tau}^{\mathrm{T}}_{\star} = \boldsymbol{\theta}^{\mathrm{T}}_{\mathrm{ft}\star} - \boldsymbol{\theta}^{\mathrm{T}}_{\mathrm{ft}}$ are the task vectors representing the explainability for the source and target domains, respectively. We refer to $\boldsymbol{\tau}^{\mathrm{S}}_{\star}$ and $\boldsymbol{\tau}^{\mathrm{T}}_{\star}$ as **explainability vectors** for the source and target domains, respectively.

From (6), if $\boldsymbol{\tau}^{\mathrm{S}}_{\star} = \boldsymbol{\tau}^{\mathrm{T}}_{\star}$ holds, $\boldsymbol{\tau}^{\mathrm{T}}_{\mathrm{ft}\star}$ can be expressed as

$$\boldsymbol{\tau}^{\mathrm{T}}_{\mathrm{ft}\star} = \underbrace{\boldsymbol{\theta}^{\mathrm{T}}_{\mathrm{ft}} - \boldsymbol{\theta}_{\mathrm{base}}}_{\boldsymbol{\tau}^{\mathrm{T}}_{\mathrm{ft}}} + \underbrace{\boldsymbol{\theta}^{\mathrm{S}}_{\mathrm{ft}\star} - \boldsymbol{\theta}_{\mathrm{base}}}_{\boldsymbol{\tau}^{\mathrm{S}}_{\mathrm{ft}\star}} - \underbrace{(\boldsymbol{\theta}^{\mathrm{S}}_{\mathrm{ft}} - \boldsymbol{\theta}_{\mathrm{base}})}_{\boldsymbol{\tau}^{\mathrm{S}}_{\mathrm{ft}}} = \underbrace{\boldsymbol{\theta}^{\mathrm{T}}_{\mathrm{ft}} - \boldsymbol{\theta}_{\mathrm{base}}}_{\boldsymbol{\tau}^{\mathrm{T}}_{\mathrm{ft}}} + \underbrace{\boldsymbol{\theta}^{\mathrm{S}}_{\mathrm{ft}\star} - \boldsymbol{\theta}^{\mathrm{S}}_{\mathrm{ft}}}_{\boldsymbol{\tau}^{\mathrm{S}}_{\star}}. \tag{7}$$

Then, we can obtain the model parameters $\tilde{\boldsymbol{\theta}}^{\mathrm{T}}_{\mathrm{ft}\star}$ by adding the task vector $\boldsymbol{\tau}^{\mathrm{T}}_{\mathrm{ft}\star}$ to the base parameters $\boldsymbol{\theta}_{\mathrm{base}}$. Introducing two scaling coefficients, $\lambda_1 \in \mathbb{R}$ for classification capability and $\lambda_2 \in \mathbb{R}$ for

explainability, we have

$$\tilde{\boldsymbol{\theta}}^{\mathrm{T}}_{\mathrm{ft}\star} = \boldsymbol{\theta}_{\mathrm{base}} + \lambda_1 \boldsymbol{\tau}^{\mathrm{T}}_{\mathrm{ft}} + \lambda_2 \boldsymbol{\tau}^{\mathrm{S}}_{\star}. \tag{8}$$

Since λ_1 and λ_2 are hyperparameters that do not affect model training, they can be tuned efficiently.

Remark on theoretical interpretation Appendix G interprets $\boldsymbol{\tau}^{\mathrm{S}}_{\star}$ as the optimal perturbation under a local quadratic approximation of (4) and shows that adding it to $\boldsymbol{\theta}^{\mathrm{T}}_{\mathrm{ft}}$ increases the target classification loss in proportion to $(\boldsymbol{\tau}^{\mathrm{S}}_{\star})^{\top} \boldsymbol{H}^{\mathrm{T}} \boldsymbol{\tau}^{\mathrm{S}}_{\star}$, where the Hessian $\boldsymbol{H}^{\mathrm{T}}$ needs no explanation supervision.

4.4 Predicting Transfer Success

Task arithmetic provides a simple and efficient method for transferring explainability, but it does not guarantee that the transfer will always be successful because the analogy in (6) may not hold for all source–target pairs, and its benefit varies considerably across source–target pairs as exemplified later in our experiments. This raises a natural question: can the resulting explanation-quality improvement be anticipated from quantities computable *before* obtaining any target-side explanation supervision? We formalize this question and examine whether a regression on model-internal features can recover the expected improvement of the transferred model.

Feature design We use six features grouped into two classes. The regression must not require any target-side explanation supervision, because with such supervision one could train $\boldsymbol{\theta}^{\mathrm{T}}_{\mathrm{ft}\star}$ directly. All features are therefore derived from $\boldsymbol{\theta}^{\mathrm{S}}_{\mathrm{ft}\star}, \boldsymbol{\theta}^{\mathrm{S}}_{\mathrm{ft}}, \boldsymbol{\theta}^{\mathrm{T}}_{\mathrm{ft}}, \boldsymbol{\tau}^{\mathrm{S}}_{\star}$, and target images with their class labels. *(i) Classification-circuit features* (3 features) compare the classification circuit of $\boldsymbol{\theta}^{\mathrm{T}}_{\mathrm{ft}}$ with the transferred model and the source explanation circuit, motivated by the classification/explanation head-level overlap established in Appendix B. They compare the per-head classification effect vectors of $\tilde{\boldsymbol{\theta}}^{\mathrm{T}}_{\mathrm{ft}\star}$ and $\boldsymbol{\theta}^{\mathrm{T}}_{\mathrm{ft}}$ via cosine similarity and top-k Jaccard overlap, plus the cosine between the source explanation effects of $\boldsymbol{\theta}^{\mathrm{S}}_{\mathrm{ft}\star}$ and the target classification effects of $\boldsymbol{\theta}^{\mathrm{T}}_{\mathrm{ft}}$. *(ii) $\boldsymbol{\tau}$-output-shift features* (3 features) probe how strongly $\boldsymbol{\tau}^{\mathrm{S}}_{\star}$ acts on target image content, on the hypothesis that a $\boldsymbol{\tau}$ that barely perturbs target image features cannot transfer explainability there. They summarize how the [CLS] feature of a frozen zero-shot CLIP backbone shifts on unlabeled target images under the perturbation $\boldsymbol{\theta}_{\mathrm{base}} \mapsto \boldsymbol{\theta}_{\mathrm{base}} + \boldsymbol{\tau}^{\mathrm{S}}_{\star}$, via the mean and standard deviation of the per-image ℓ_2 shift across images and the cosine of the image-mean (centroid) feature pair. Formal definitions for all six features are in Appendix C.

Regression model Given N source–target pairs available at training time, we form a feature matrix $\boldsymbol{X} \in \mathbb{R}^{N \times 6}$ (with each column standardized to zero mean and unit variance) and a target vector $\boldsymbol{y} \in \mathbb{R}^{N}$ encoding the explanation-quality improvement of $\tilde{\boldsymbol{\theta}}^{\mathrm{T}}_{\mathrm{ft}\star}$ over $\boldsymbol{\theta}^{\mathrm{T}}_{\mathrm{ft}}$ on each pair, then fit a linear regression $\hat{\boldsymbol{y}} = \boldsymbol{X}\hat{\boldsymbol{\beta}}$ with regularization strength chosen by inner leave-one-target-out cross-validation. At test time, the regression produces $\hat{y}$ for a held-out target without ever touching any target-side explanation supervision.

5 Experiments

We evaluate the proposed explainability transfer in two parts: Section 5.1 measures whether a single explainability vector learned on a large-scale source (ImageNet+X) transfers to 10 diverse target datasets, and Section 5.2 empirically tests the predictability examined in Section 4.4 with ImageNet+X as the source and 10 targets. Complementary results are deferred to the appendix. They include a backbone-scaling study on CLIP with ViT-L/14 (Appendix E) and an all-pairs transferability evaluation across eight smaller-source datasets (Appendix F).

Common setup We use CLIP with ViT-B/32 [49, 18] pretrained by OpenAI as the self-explaining model, whose parameters serve as $\boldsymbol{\theta}_{\mathrm{base}}$. The model input is a 224×224 RGB image, each patch is 32×32, and the number of patches is $M = 49$. We fix the hyperparameter $\lambda_1 = 1.0$ to focus on the effect of explainability transfer via $\boldsymbol{\tau}^{\mathrm{S}}_{\star}$ while preserving classification performance on the target datasets, and use $\alpha = 0.8$ and $\lambda_2 = 0.7$ unless otherwise specified. These values are a representative operating point on the accuracy–explanation trade-off rather than optimal values (Appendix M discusses selecting λ_2 without target-side explanation supervision). We sweep $\alpha \in \{0.1, 0.2, \ldots, 0.9\}$ and $\lambda_2 \in \{0.4, 0.5, \ldots, 1.2\}$ in the sensitivity analysis (Figure 2). The computing environments are described in Appendix D.

Table 1: Average performances and their standard deviations of ViT-B/32-based self-explaining models over the 10 target datasets when transferring explainability learned on ImageNet+X, compared with those of baselines. The last row shows the oracle performances achieved by finetuning on the target datasets with explanation supervision (target+X). **Bold and underlined values** indicate the best among the proposed method and the baselines for each metric, while **bold values** indicate that there is no statistically significant difference from the best, as determined by a paired t-test at a significance level of 0.05.

	Param.	Accuracy (↑)	E-RMSE (↓)	Rank corr. (↑)	IoU@1 (↑)	IoU@5 (↑)	IoU@10 (↑)
Zero-shot	$\boldsymbol{\theta}_{\mathrm{base}}$	0.475 ± 0.211	1.508 ± 0.069	-0.095 ± 0.059	0.012 ± 0.006	0.040 ± 0.013	0.090 ± 0.021
Finetuned on ImageNet	$\boldsymbol{\theta}^{\mathrm{S}}_{\mathrm{ft}}$	0.379 ± 0.239	1.528 ± 0.081	-0.128 ± 0.076	0.009 ± 0.006	0.031 ± 0.016	0.074 ± 0.026
Finetuned on ImageNet+X	$\boldsymbol{\theta}^{\mathrm{S}}_{\mathrm{ft}\star}$	0.353 ± 0.243	$\mathbf{1.149 \pm 0.092}$	$\mathbf{0.197 \pm 0.063}$	$\mathbf{0.106 \pm 0.059}$	$\mathbf{0.187 \pm 0.082}$	$\mathbf{0.261 \pm 0.079}$
Finetuned on target	$\boldsymbol{\theta}^{\mathrm{T}}_{\mathrm{ft}}$	$\underline{\mathbf{0.890 \pm 0.106}}$	1.562 ± 0.084	-0.145 ± 0.068	0.009 ± 0.005	0.033 ± 0.013	0.076 ± 0.021
Proposed	$\tilde{\boldsymbol{\theta}}^{\mathrm{T}}_{\mathrm{ft}\star}$	$\mathbf{0.886 \pm 0.113}$	$\underline{\mathbf{1.125 \pm 0.116}}$	$\underline{\mathbf{0.200 \pm 0.061}}$	$\underline{\mathbf{0.133 \pm 0.092}}$	$\underline{\mathbf{0.216 \pm 0.114}}$	$\underline{\mathbf{0.279 \pm 0.093}}$
Finetuned on target+X	$\boldsymbol{\theta}^{\mathrm{T}}_{\mathrm{ft}\star}$	0.909 ± 0.100	1.028 ± 0.224	0.242 ± 0.104	0.248 ± 0.217	0.291 ± 0.189	0.328 ± 0.140

Evaluation metrics For classification quality, we report top-1 accuracy. For explanation quality, we report E-RMSE between predicted and GT patch-level attributions, Kendall rank correlation [33, 32] (*Rank corr.* for short) measuring the similarity between the rankings of predicted and GT attributions, and IoU@k ($k \in \{1, 5, 10\}$), a top-k variant of the IoU in [71], measuring the overlap between the indices of the top-k predicted and GT patch-level attributions, which evaluates the correctness of highlighted patches. These GT-agreement metrics directly evaluate our objective of reproducing a user-specified explanation on the target. Whether the highlighted patches also matter for the transferred model's own predictions is evaluated separately with within-model faithfulness metrics in Section 5.1.5.

5.1 Transferring Explainability Learned on ImageNet

Setup We use ImageNet-1k [17] as the source dataset, since its scale and class diversity make it a natural candidate for capturing transferable patch-level explanations. We construct the ImageNet+X dataset, in which each sample consists of a labeled image from the original ImageNet-1k and a GT patch-level attribution. The GT attributions are generated using Kernel SHAP [41, 53] with 500 perturbations per image, applied to the CLIP image classifier finetuned on ImageNet-1k. Similar approaches for generating explanation supervision have been used in previous studies [64, 43]. As the target datasets, we use 10 image classification datasets: Cars [36], DTD [14], EuroSAT [25], GTSRB [55], RESISC45 [12], SUN397 [63], SVHN [45], MNIST [38], CIFAR100 [37], and STL10 [15]. As with ImageNet+X, we add the GT patch-level attributions to the 10 target datasets for evaluation, following the same Kernel SHAP procedure with the per-target finetuned classifier.

5.1.1 Quantitative Results

Table 1 shows the average performances of the explainability transferred by the proposed method, in comparison to the baseline models with the parameters $\boldsymbol{\theta}_{\mathrm{base}}$, $\boldsymbol{\theta}^{\mathrm{S}}_{\mathrm{ft}}$, $\boldsymbol{\theta}^{\mathrm{S}}_{\mathrm{ft}\star}$, and $\boldsymbol{\theta}^{\mathrm{T}}_{\mathrm{ft}}$. For reference, we also show the performances of the model finetuned on the target dataset with explanation supervision, i.e., the oracle $\boldsymbol{\theta}^{\mathrm{T}}_{\mathrm{ft}\star}$. The table shows that compared to the baselines, the proposed method achieves the second-best accuracy while consistently achieving the best explainability performance. The accuracy gap to the oracle $\boldsymbol{\theta}^{\mathrm{T}}_{\mathrm{ft}\star}$ is only 0.023 ($\approx 2.5\%$ relative), and the E-RMSE gap is $\approx 9\%$ relative. The proposed method thus transfers explainability acquired on ImageNet+X while largely preserving classification accuracy. The most accurate baseline, $\boldsymbol{\theta}^{\mathrm{T}}_{\mathrm{ft}}$, has the worst or second-worst explainability, while the second-best explainability, that of $\boldsymbol{\theta}^{\mathrm{S}}_{\mathrm{ft}\star}$, comes with the lowest accuracy. The proposed method combines the high target-domain accuracy of $\boldsymbol{\theta}^{\mathrm{T}}_{\mathrm{ft}}$ with the explainability vector extracted from $\boldsymbol{\theta}^{\mathrm{S}}_{\mathrm{ft}\star}$, attaining classification accuracy on par with $\boldsymbol{\theta}^{\mathrm{T}}_{\mathrm{ft}}$ and explanation quality close to the oracle $\boldsymbol{\theta}^{\mathrm{T}}_{\mathrm{ft}\star}$.

5.1.2 Impact of Scaling Coefficient

We investigate the impact of the scaling coefficient for explainability λ_2 for each target dataset. Figure 2 shows the normalized E-RMSE, IoU@10, and accuracy for each target dataset over $\lambda_2 \in \{0.4, 0.5, \ldots, 1.2\}$. The scores are normalized by dividing them by the corresponding scores of $\boldsymbol{\theta}^{\mathrm{T}}_{\mathrm{ft}}$. Although trends vary by dataset, the explainability scores in Figures 2(a) and (b) improve

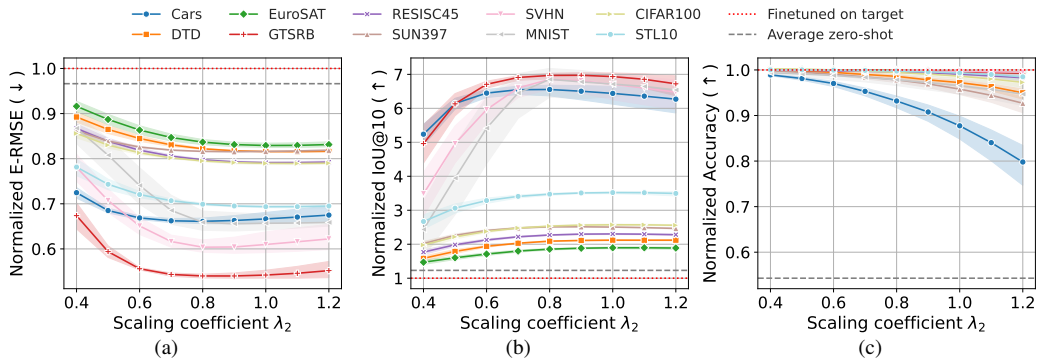


Figure 2: Normalized (a) E-RMSE, (b) IoU@10, and (c) accuracy for each target dataset over the scaling coefficient for explainability λ_2. Lines show the mean over the nine models trained with $\alpha \in \{0.1, 0.2, \ldots, 0.9\}$. Shaded areas show a 95% bootstrap interval of that mean, obtained by resampling the nine α values.

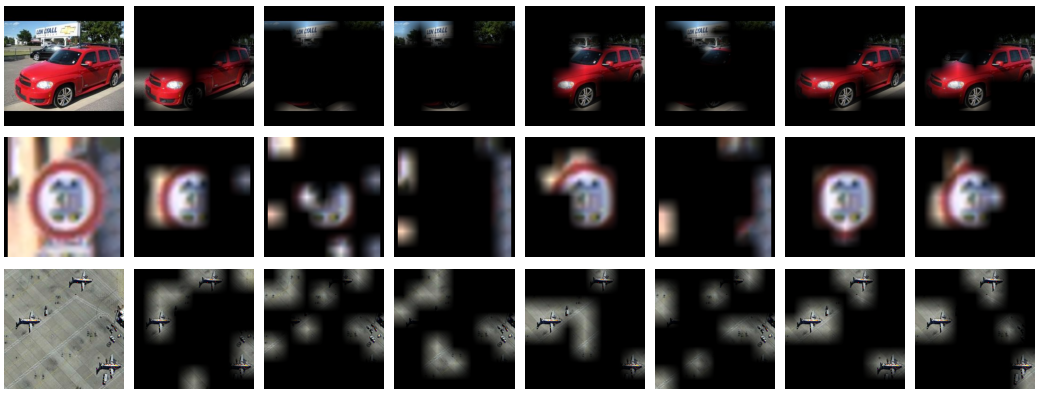


Figure 3: Visualization examples highlighting the top-10 patches in the predicted patch-level attributions. Rows show (from top to bottom) the results for Cars, GTSRB, and RESISC45 datasets.

for all 10 target datasets, with improvements plateauing around $\lambda_2 \approx 0.8$. Figure 2(c) shows that classification accuracy tends to decrease as λ_2 increases, but the degradation is kept within 5% of $\boldsymbol{\theta}_{\text{ft}}^{\text{T}}$ when $\lambda_2 \leq 1.0$, except for Cars.

The theoretical analysis in Appendix G shows that such a drop can occur independently of the gain in explanation quality. For Cars the drop is noticeable, but the explanation performance is already high at $\lambda_2 \leq 0.6$, so an appropriate λ_2 balances the two. The fixed setting $\lambda_2 = 0.7$ used uniformly across the 10 targets in Table 1 already attains substantial improvements without any target-side explanation supervision. λ_2 can also be selected without target explanation labels. Among the values whose mean accuracy on the other targets stays within a tolerance of the best, we take the one with the best mean explanation metric on those targets. With a tolerance of 0.5 points, this rule selects 0.7 ± 0.1 for 9 of 10 held-out targets (Appendix M).

5.1.3 Qualitative Evaluation

To evaluate the predicted patch-level attributions qualitatively, we visualize the top-10 patches in the attributions produced by the proposed method and the other methods in Figure 3, which corresponds to the IoU@10 evaluation. The patches highlighted by the proposed method (Figure 3(g)) are more similar to the GT ones (Figure 3(b)) than those highlighted by the three baseline methods that do not use any explanation supervision (Figures 3(c), (d), and (f)), and visually approach the target+X oracle (Figure 3(h)). The model finetuned on ImageNet+X (Figure 3(e)) also generates high-quality explanations similar to the proposed method, consistent with the quantitative evaluation. However, due to its low classification performance, it is not practical to use this single model for both

classification and explanation simultaneously. Additional randomly selected examples for each of the 10 target datasets are provided in Appendix Q.

5.1.4 Comparison with Kernel SHAP

The GT patch-level attributions for ViT-B/32 are generated using Kernel SHAP applied to $\boldsymbol{\theta}_{\mathrm{ft}}^{\mathrm{T}}$ with $P = 500$ perturbations (SHAP-500) per image, which produces high-quality explanations but is computationally expensive (computational complexity is detailed in Appendix I). Once $\tilde{\boldsymbol{\theta}}_{\mathrm{ft}\star}^{\mathrm{T}}$ is obtained, the proposed method instead produces patch-level attributions in a single forward pass. To quantify this efficiency advantage, we compute Kernel SHAP attributions on $\boldsymbol{\theta}_{\mathrm{ft}}^{\mathrm{T}}$ with varying P (using independent perturbation samples from those used for GT) and evaluate them against the GT alongside the proposed method's deterministic attributions.

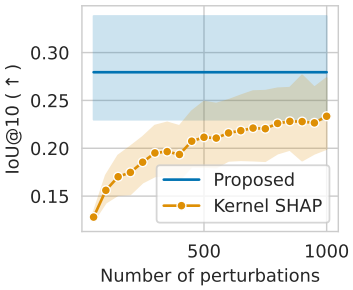


Figure 4: IoU@10 of the proposed method and Kernel SHAP applied to $\boldsymbol{\theta}_{\mathrm{ft}}^{\mathrm{T}}$ across the 10 target datasets, with varying number of perturbations P for SHAP. The shaded area represents the 95% bootstrap confidence interval. The full five-metric grid is in Figure 10 in Appendix H.

Figure 4 reports the average IoU@10 across the 10 target datasets, complementing the top-10 visualization in the previous subsection (the full five-metric grid is in Appendix H). The proposed method attains IoU@10 $= 0.279$ against the GT, whereas Kernel SHAP on $\boldsymbol{\theta}_{\mathrm{ft}}^{\mathrm{T}}$ reaches only 0.233 even at $P = 1{,}000$ perturbations. This gap reflects the sampling variance of Kernel SHAP at the GT budget. An independent SHAP-500 run on the same model attains IoU@10 $= 0.211$ against the GT, so any single SHAP-500 realization is a noisy estimator. The learned attributions may average out this noise, at the cost of a single forward pass.

5.1.5 Faithfulness, Specificity, and Comparison with External Explainers

GT agreement does not show whether the highlighted patches matter for the transferred model's own predictions, whether the effect is specific to $\boldsymbol{\tau}_{\star}^{\mathrm{S}}$, or how the method compares with other ways of explaining a finetuned classifier. We address these questions with the pipeline of Table 1 on a 10-target set in which CIFAR10 replaces SUN397. On this set, the mean accuracy is 0.914 for $\boldsymbol{\theta}_{\mathrm{ft}}^{\mathrm{T}}$ and 0.911 for the proposed method.

Within-model faithfulness We test whether the patches highlighted by the transferred model $\tilde{\boldsymbol{\theta}}_{\mathrm{ft}\star}^{\mathrm{T}}$ actually drive its own predictions. We mask the patches in order of attributed importance and measure how the predicted-class probability of $\tilde{\boldsymbol{\theta}}_{\mathrm{ft}\star}^{\mathrm{T}}$ changes on the correctly classified images among 500 class-stratified test images per target. We quantify this change with four metrics: Fidelity$+$ and Fidelity$-$ [57], and deletion and insertion scores [47]. We compare three attributions scored on the same model. They are the self-explanation of $\tilde{\boldsymbol{\theta}}_{\mathrm{ft}\star}^{\mathrm{T}}$, Kernel SHAP applied to $\tilde{\boldsymbol{\theta}}_{\mathrm{ft}\star}^{\mathrm{T}}$ itself (number of perturbations $P = 500$) as a model-specific reference, and a random attribution. The self-explanation matches or exceeds the model-specific SHAP reference on 36 of the 40 comparisons (10 targets $\times$ 4 metrics) and beats random on all 40 (Appendix J). The highlighted patches are thus the ones that move the model's own predictions.

Specificity of the effect We test whether the explanation improvement is caused by $\boldsymbol{\tau}_{\star}^{\mathrm{S}}$ itself rather than by adding any vector to the model. To this end, we replace $\boldsymbol{\tau}_{\star}^{\mathrm{S}}$ in (8) by control vectors, keep the rest of the pipeline unchanged, and compare the resulting models (Table 2). Adding an equal-norm random direction or the vector restricted to the first six transformer blocks gives almost the same results as adding no vector. In contrast, the last six blocks alone recover 84% of the Rank corr. improvement and 82% of the IoU@1 improvement of $\boldsymbol{\tau}_{\star}^{\mathrm{S}}$ over no vector. These results show that the effect depends on the direction of the vector and comes mostly from the later blocks (Appendices K

Table 2: Accuracy and explanation quality when each vector is added in place of $\boldsymbol{\tau}_{\star}^{\mathrm{S}}$ in (8), averaged over the 10 targets. Per-target results are in Appendix K.

Added vector	Acc.	Rank corr.	IoU@1
Proposed	0.911	0.204	0.132
No vector	0.914	−0.147	0.009
Equal-norm random	0.909	−0.144	0.010
Front-6 blocks	0.912	−0.147	0.009
Back-6 blocks	0.913	0.148	0.109
Shuffled supervision	0.909	0.190	0.079

and B.3). A vector retrained with explanation maps shuffled across source images recovers 96% of the Rank corr. improvement but only 57% of the IoU@1 improvement (Appendix K). This suggests that the vector carries both an attribution prior that does not depend on the content of the supervision and a component that does.

Comparison with external explainers We also compare the proposed method with two external explainers: a separate model trained with the explanation loss alone ($\alpha = 0$ in (4)) that attributes the predictions of $\boldsymbol{\theta}_{\mathrm{ft}}^{\mathrm{T}}$, and TVE [57], which in our evaluation with its public checkpoint explains a linear probe on its own frozen encoder (Appendix L). Both produce non-random explanations for the classifier they accompany. Unlike them, the proposed method produces the explanation within the finetuned target itself, in one forward pass with no extra parameters. TVE's probe is 13.5 points less accurate than $\boldsymbol{\theta}_{\mathrm{ft}}^{\mathrm{T}}$ on this set. When the deployed model must not be modified, post-hoc explainers such as Kernel SHAP remain the appropriate choice (Appendix O).

5.2 Empirical Test of Transfer-Success Prediction

We empirically test the predictability examined in Section 4.4 with ImageNet+X as the source and the 10 targets of Section 5.1, with CIFAR10 [37] in place of CIFAR100. Let $q(\cdot)$ denote a function that maps a model's parameters to a scalar explanation-quality score evaluated on $\mathcal{D}^{\mathrm{T}}$, instantiated with $q \in \{\text{E-MSE}, \text{IoU@1}, \text{IoU@5}, \text{IoU@10}, \text{Rank corr.}\}$. E-MSE is the squared E-RMSE, $\mathcal{L}_{\mathrm{exp}}$ in (5). All scores use 512 fixed random test images per target. For each metric, let $b_q = q(\boldsymbol{\theta}_{\mathrm{ft}}^{\mathrm{T}})$ denote the baseline value before explainability transfer and $a_q = q(\tilde{\boldsymbol{\theta}}_{\mathrm{ft}\star}^{\mathrm{T}})$ the value after transfer. For source dataset $\mathcal{D}^{\mathrm{S}}$ and target dataset $\mathcal{D}^{\mathrm{T}}$, we define the predictor target as the (sign-aligned) raw improvement in explanation quality: $\Delta_q(S, T) = s_q \cdot (a_q - b_q)$, where $s_q \in \{+1, -1\}$ aligns the sign so that larger Δ_q indicates better transfer ($s_q = -1$ for E-MSE and $s_q = +1$ otherwise). We obtain $N = 10$ source–target pairs (one per target) and perform leave-one-target-out (LOTO) evaluation: each fold withholds one target, fits Lasso or Ridge regression on the remaining nine pairs with inner-LOTO regularization selection, and predicts Δ_q for the held-out target. For each metric, we use the better of Lasso and Ridge and report the held-out Pearson r between predicted and actual Δ_q over the 10 targets. A one-sided 500-permutation test reruns all steps, including this choice.

Result Table 3 shows that a regression on the six candidate features strongly correlates with the actual transfer outcome on the three reported metrics. On E-MSE, Lasso attains $r = +0.92$ across the 10 held-out targets (uncorrected permutation $p = 0.006$), with a bootstrap 95% interval of $[+0.84, +0.98]$. The intervals on Rank corr. are wider, at $[+0.56, +0.92]$ for Lasso and $[+0.70, +0.94]$ for Ridge (Appendix F.2). Over all five metrics (full grid in Appendix C), the regression reaches $r \geq +0.70$ on four, namely E-MSE, Rank corr., IoU@5, and IoU@10 (uncorrected permutation $p \leq 0.022$). The remaining metric, IoU@1, is comparatively weak ($r = +0.33$ at best). We revisit this metric- and backbone-specific unevenness in Appendix O. A held-out evaluation of the same predictor on 56 cross-domain pairs is provided in Appendix F. The claim is restricted to a fixed source with held-out targets. Under leave-one-source-out evaluation on the 56 pairs, none of 40 predictor configurations meets our criteria (Appendix F.2).

Table 3: Held-out Pearson correlation (LOTO, $N = 10$) between predicted and actual Δ_q on ViT-B/32 for three representative metrics. The full five-metric ViT-B/32 grid and ViT-L/14 results are in Appendix C. Best per metric is in **bold**.

Method	E-MSE	IoU@10	Rank corr.
Lasso	$\mathbf{+0.92}$	$+0.81$	$+0.77$
Ridge	$+0.91$	$\mathbf{+0.85}$	$\mathbf{+0.82}$

6 Conclusion

We proposed a task-arithmetic method to transfer visual explainability from self-explaining to prediction-only VLM-based ViTs across domains, without target-side explanation supervision. A single explainability vector learned on ImageNet+X improved explanation quality on 10 diverse target datasets while largely preserving accuracy. The transferred explanations are faithful to the model's own predictions, and comparisons with control vectors (e.g., random vectors of the same norm) show that the effect is specific to the vector's direction, largely mediated by the later blocks, and partly dependent on the content of the source supervision. For a fixed source, transfer success on held-out targets is predictable to a meaningful degree from a small set of model-internal features alone. Limitations and broader impacts are in Appendices O and P.

Acknowledgments and Disclosure of Funding

This work was supported by JSPS KAKENHI Grant Number JP26K14968 and by a joint research fund from ZOZO Research. The authors declare no other competing interests.

References

[1] Zoe Alexander, Duen Horng Chau, and Christopher Saldaña. An interrogative survey of explainable AI in manufacturing. *IEEE Transactions on Industrial Informatics*, 2024.

[2] Amr Alkhatib, Roman Bresson, Henrik Boström, and Michalis Vazirgiannis. Prediction via Shapley value regression. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 1056–1101. PMLR, 2025.

[3] David Alvarez Melis and Tommi Jaakkola. Towards robust interpretability with self-explaining neural networks. *Advances in Neural Information Processing Systems*, 31, 2018.

[4] Andrea Apicella, Luca Di Lorenzo, Francesco Isgrò, Andrea Pollastro, and Roberto Prevete. Strategies to exploit XAI to improve classification systems. In *World Conference on Explainable Artificial Intelligence*, pages 147–159. Springer, 2023.

[5] 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. *IEEE Access*, 2024.

[6] Nicola Capuano, Giuseppe Fenza, Vincenzo Loia, and Claudio Stanzione. Explainable artificial intelligence in cybersecurity: A survey. *IEEE Access*, 10:93575–93600, 2022.

[7] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9630–9640, 2021. doi: 10.1109/ICCV48922.2021.00951.

[8] Hila Chefer, Shir Gur, and Lior Wolf. Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 397–406, 2021.

[9] 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.

[10] Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. This looks like that: deep learning for interpretable image recognition. *Advances in neural information processing systems*, 32, 2019.

[11] Shiqi Chen, Jinghan Zhang, Tongyao Zhu, Wei Liu, Siyang Gao, Miao Xiong, Manling Li, and Junxian He. Bring reason to vision: Understanding perception and reasoning through model merging. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 9803–9817. PMLR, 2025.

[12] Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883, 2017.

[13] 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.

[14] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3606–3613, 2014.

[15] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011.

[16] Ian Connick Covert, Chanwoo Kim, and Su-In Lee. Learning to estimate Shapley values with vision transformers. In *The Eleventh International Conference on Learning Representations*, 2023.

[17] 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.

[18] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

[19] Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, Eyal Orgad, Rahim Entezari, Giannis Daras, Sarah Pratt, Vivek Ramanujan, Yonatan Bitton, Kalyani Marathe, Stephen Mussmann, Richard Vencu, Mehdi Cherti, Ranjay Krishna, Pang Wei Koh, Olga Saukh, Alexander Ratner, Shuran Song, Hannaneh Hajishirzi, Ali Farhadi, Romain Beaumont, Sewoong Oh, Alex Dimakis, Jenia Jitsev, Yair Carmon, Vaishaal Shankar, and Ludwig Schmidt. DataComp: In search of the next generation of multimodal datasets. In *Advances in Neural Information Processing Systems (Datasets and Benchmarks Track)*, volume 36, 2023.

[20] Yossi Gandelsman, Alexei A. Efros, and Jacob Steinhardt. Interpreting CLIP's image representation via text-based decomposition. In *International Conference on Learning Representations (ICLR)*, 2024.

[21] Jacob Gildenblat and contributors. PyTorch library for CAM methods. https://github.com/jacobgil/pytorch-grad-cam, 2021.

[22] Shizhan Gong, Haoyu Lei, Qi Dou, and Farzan Farnia. Boosting the visual interpretability of CLIP via adversarial fine-tuning. In *International Conference on Learning Representations (ICLR)*, 2025.

[23] Laura Gomezjurado Gonzalez, Hiroki Naganuma, Kotaro Yoshida, Takafumi Horie, Yuji Naraki, and Ryotaro Shimizu. On fairness of task arithmetic: The role of task vectors. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=B19MBDrvlM.

[24] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15979–15988, 2022. doi: 10.1109/CVPR52688.2022.01553.

[25] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. EuroSAT: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019.

[26] Rami Ibrahim and M Omair Shafiq. Explainable convolutional neural networks: a taxonomy, review, and future directions. *ACM Computing Surveys*, 55(10):1–37, 2023.

[27] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. OpenCLIP, July 2021. URL https://doi.org/10.5281/zenodo.5143773.

[28] Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*, 2023.

[29] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. In *Proceedings of the 34th Conference on Uncertainty in Artificial Intelligence*, volume R16 of *Proceedings of Machine Learning Research*, pages 875–884. PMLR, 2018.

[30] Sarthak Jain and Byron C Wallace. Attention is not explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3543–3556, 2019.

[31] Yang Ji, Ying Sun, Yuting Zhang, Zhigaoyuan Wang, Yuanxin Zhuang, Zheng Gong, Dazhong Shen, Chuan Qin, Hengshu Zhu, and Hui Xiong. A comprehensive survey on self-interpretable neural networks. *Proceedings of the IEEE*, 113(8):783–813, 2025. doi: 10.1109/JPROC.2025.3635153.

[32] Weina Jin, Xiaoxiao Li, Mostafa Fatehi, and Ghassan Hamarneh. Guidelines and evaluation of clinical explainable AI in medical image analysis. *Medical Image Analysis*, 84:102684, February 2023. ISSN 1361-8415,1361-8423. doi: 10.1016/j.media.2022.102684. URL `http://dx.doi.org/10.1016/j.media.2022.102684`.

[33] Maurice Kendall and Jean D. Gibbons. *Rank Correlation Methods*. Oxford University Press, New York, 5th edition, 1990. ISBN 0195208374.

[34] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV). In *International conference on machine learning*, pages 2668–2677. PMLR, 2018.

[35] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *International conference on machine learning*, pages 5338–5348. PMLR, 2020.

[36] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3D object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013.

[37] Alex Krizhevsky. Learning multiple layers of features from tiny images, 2009. URL `https://api.semanticscholar.org/CorpusID:18268744`.

[38] Yann LeCun, Corinna Cortes, and Christopher J.C. Burges. The MNIST database of handwritten digits. `http://yann.lecun.com/exdb/mnist/`, 1998.

[39] Alex Li, Yuandong Tian, Beidi Chen, Deepak Pathak, and Xinlei Chen. On the surprising effectiveness of attention transfer for vision transformers. *Advances in Neural Information Processing Systems*, 37:113963–113990, 2024.

[40] Yi Li, Hualiang Wang, Yiqun Duan, Jiheng Zhang, and Xiaomeng Li. A closer look at the explainability of Contrastive language-image pre-training. *Pattern Recognition*, 162:111409, 2025.

[41] Scott M Lundberg and Su-In Lee. A Unified Approach to Interpreting Model Predictions. *Advances in Neural Information Processing Systems*, pages 4765–4774, 2017.

[42] Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one? In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

[43] Lucas Monteiro Paes, Dennis Wei, and Flavio Calmon. Selective explanations. *Advances in Neural Information Processing Systems*, 37:55562–55590, 2024.

[44] Sajid Nazir, Diane M Dickson, and Muhammad Usman Akram. Survey of explainable artificial intelligence techniques for biomedical imaging with deep neural networks. *Computers in Biology and Medicine*, 156:106668, 2023.

[45] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Baolin Wu, Andrew Y Ng, et al. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, page 4. Granada, 2011.

[46] Shiwen Ni, Dingwei Chen, Chengming Li, Xiping Hu, Ruifeng Xu, and Min Yang. Forgetting before learning: Utilizing parametric arithmetic for knowledge updating in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5716–5731, 2024.

[47] Vitali Petsiuk, Abir Das, and Kate Saenko. RISE: Randomized input sampling for explanation of black-box models. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2018. arXiv:1806.07421.

[48] Yao Qiang, Chengyin Li, Hui Zhu, Prashant Khanduri, and Dongxiao Zhu. Interpretability-aware vision transformer. In *2025 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2025. doi: 10.1109/IJCNN64981.2025.11227842.

[49] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, pages 8748–8763, 2021.

[50] Andrew Slavin Ross, Michael C Hughes, and Finale Doshi-Velez. Right for the right reasons: Training differentiable models by constraining their explanations. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, pages 2662–2670. International Joint Conferences on Artificial Intelligence Organization, 2017.

[51] 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. *International Journal of Computer Vision*, 128(2):336–359, 2020.

[52] Sofia Serrano and Noah A Smith. Is attention interpretable? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2931–2951, 2019.

[53] SHAP (GitHub). SHAP: A game theoretic approach to explain the output of any machine learning model, 2024. URL `https://github.com/shap/shap`. Accessed: 2024-2-2.

[54] Daiki Shirafuji, Makoto Takenaka, and Shinya Taguchi. Bias vector: Mitigating biases in language models with task arithmetic approach. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 2799–2813, 2025.

[55] Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. The German traffic sign recognition benchmark: a multi-class classification competition. In *The 2011 international joint conference on neural networks*, pages 1453–1460. IEEE, 2011.

[56] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[57] Guanchu Wang, Yu-Neng Chuang, Fan Yang, Mengnan Du, Chia-Yuan Chang, Shaochen Zhong, Zirui Liu, Zhaozhuo Xu, Kaixiong Zhou, Xuanting Cai, et al. TVE: Learning meta-attribution for transferable vision explainer. In *International Conference on Machine Learning*, pages 50248–50267. PMLR, 2024.

[58] Rui Wang, Xiaoqian Wang, and David Inouye. Shapley explanation networks. In *International Conference on Learning Representations*, 2021.

[59] Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International conference on machine learning*, pages 23965–23998. PMLR, 2022.

[60] Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, et al. Robust fine-tuning of zero-shot models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7959–7971, 2022.

[61] Junyi Wu, Bin Duan, Weitai Kang, Hao Tang, and Yan Yan. Token transformation matters: Towards faithful post-hoc explanation for vision transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10926–10935, 2024. doi: 10.1109/CVPR52733.2024.01039.

[62] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[63] Jianxiong Xiao, Krista A Ehinger, James Hays, Antonio Torralba, and Aude Oliva. SUN database: Exploring a large collection of scene categories. *International Journal of Computer Vision*, 119:3–22, 2016.

[64] Chenghao Yang, Fan Yin, He He, Kai-Wei Chang, Xiaofei Ma, and Bing Xiang. Efficient Shapley values estimation by amortization for text classification. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8666–8680, 2023.

[65] Kotaro Yoshida, Yuji Naraki, Takafumi Horie, Ryosuke Yamaki, Ryotaro Shimizu, Yuki Saito, Julian McAuley, and Hiroki Naganuma. Mastering task arithmetic: τJp as a key indicator for weight disentanglement. In *The Thirteenth International Conference on Learning Representations*, 2025.

[66] Kotaro Yoshida, Yuji Naraki, Takafumi Horie, Ryotaro Shimizu, Ioannis Mitliagkas, and Hiroki Naganuma. DisTaC: Conditioning task vectors via distillation for robust model merging. In *The Fourteenth International Conference on Learning Representations*, 2026. URL `https://openreview.net/forum?id=W70w5JCzdq`.

[67] Yuya Yoshikawa and Tomoharu Iwata. Neural generators of sparse local linear models for achieving both accuracy and interpretability. *Information Fusion*, 81:116–128, 2022. doi: 10.1016/j.inffus.2021.11.009.

[68] Yuya Yoshikawa and Tomoharu Iwata. Gaussian process regression with interpretable sample-wise feature weights. *IEEE Transactions on Neural Networks and Learning Systems*, 34(9): 5789–5803, 2023. doi: 10.1109/TNNLS.2021.3131234.

[69] Yuya Yoshikawa, Masanari Kimura, Ryotaro Shimizu, and Yuki Saito. Explaining black-box model predictions via two-level nested feature attributions with consistency property. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 6875–6885, 2025. doi: 10.24963/ijcai.2025/765.

[70] Lu Yu, Wei Xiang, Juan Fang, Yi-Ping Phoebe Chen, and Lianhua Chi. Ex-ViT: A novel explainable vision transformer for weakly supervised semantic segmentation. *Pattern Recognition*, 142:109666, 2023.

[71] Yifei Zhang, James Song, Siyi Gu, Tianxu Jiang, Bo Pan, Guangji Bai, and Liang Zhao. Saliency-Bench: A comprehensive benchmark for evaluating visual explanations, 2025. URL `https://arxiv.org/abs/2310.08537`.

[72] Chenyang Zhao, Kun Wang, Xingyu Zeng, Rui Zhao, and Antoni B. Chan. Gradient-based visual explanation for transformer-based CLIP. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pages 61072–61091, 2024.

Appendix Contents

A. Ablation Study: Shared Head vs. Separated Head
B. Circuit Overlap between Classification and Explanation
  B.1 Classification and Explanation Heads Largely Overlap
  B.2 Circuit Overlap Is Present before Explanation Training
  B.3 Where the Transferred Effect Is Mediated
C. Feature Pool and ViT-L/14 Predictability Analysis
D. Computing Environment for Experiments
E. Experiments with ViT-L/14
  E.1 Setup
  E.2 Quantitative Results
  E.3 Qualitative Evaluation
F. Transferring Explainability Across Source and Target Datasets
  F.1 Accuracy and Explanation Error Trade-off
  F.2 Predictability Analysis on 56 Pairs
G. Theoretical Interpretation
H. Results on Comparison with Kernel SHAP
I. Discussion on Computational Complexity of Explanation Generation
J. Faithfulness of the Transferred Self-Explanations
K. Control-Vector Ablations
L. Comparison with Amortized and Transferable Explainers
  L.1 Amortized Explainer with the Same Supervision
  L.2 Transferable Vision Explainer
M. Selecting λ_2 without Target-Side Explanation Supervision
N. Task-Dependent Explanations on Two-Task Images
O. Limitations
P. Impact Statement
Q. Additional Visualization Examples

Table 4: Comparison of shared-head (proposed) and separated-head architectures when finetuning on each target dataset with explanation supervision (target+X), averaged across the 10 target datasets. The shared-head model keeps $g_{\boldsymbol{W}}$ frozen from the pretrained model, while the separated-head model has separate classification and explanation heads, both initialized from the pretrained $g_{\boldsymbol{W}}$, and trains the entire model including the feature extractor and both heads. **Bold and underlined values** indicate the best among all settings for each metric, while **bold values** indicate that there is no statistically significant difference from the best, as determined by a paired t-test at a significance level of 0.05.

Head	α	Accuracy (↑)	E-RMSE (↓)	Rank corr. (↑)	IoU@1 (↑)	IoU@5 (↑)	IoU@10 (↑)
Separated	0.0	0.474 ± 0.322	$\mathbf{1.058 \pm 0.242}$	$\mathbf{0.218 \pm 0.125}$	$\underline{\mathbf{0.250 \pm 0.227}}$	$\mathbf{0.280 \pm 0.198}$	$\mathbf{0.309 \pm 0.150}$
Separated	0.1	0.886 ± 0.122	$\mathbf{1.066 \pm 0.244}$	$\mathbf{0.211 \pm 0.126}$	$\mathbf{0.246 \pm 0.227}$	$\mathbf{0.275 \pm 0.199}$	$\mathbf{0.305 \pm 0.150}$
Separated	0.2	0.891 ± 0.118	$\mathbf{1.067 \pm 0.244}$	0.211 ± 0.126	0.243 ± 0.228	$\mathbf{0.275 \pm 0.199}$	$\mathbf{0.305 \pm 0.149}$
Separated	0.3	0.893 ± 0.116	$\mathbf{1.068 \pm 0.244}$	0.210 ± 0.126	$\mathbf{0.244 \pm 0.226}$	$\mathbf{0.274 \pm 0.199}$	$\mathbf{0.305 \pm 0.150}$
Separated	0.4	0.896 ± 0.114	$\mathbf{1.069 \pm 0.243}$	0.210 ± 0.125	$\mathbf{0.243 \pm 0.228}$	$\mathbf{0.274 \pm 0.198}$	$\mathbf{0.304 \pm 0.149}$
Separated	0.5	0.896 ± 0.112	$\mathbf{1.069 \pm 0.242}$	0.210 ± 0.125	$\mathbf{0.243 \pm 0.225}$	$\mathbf{0.274 \pm 0.198}$	$\mathbf{0.304 \pm 0.149}$
Separated	0.6	0.899 ± 0.111	1.069 ± 0.242	0.210 ± 0.124	$\mathbf{0.241 \pm 0.225}$	$\mathbf{0.273 \pm 0.198}$	0.305 ± 0.149
Separated	0.7	0.900 ± 0.110	1.069 ± 0.241	0.209 ± 0.123	$\mathbf{0.239 \pm 0.223}$	$\mathbf{0.273 \pm 0.196}$	0.304 ± 0.148
Separated	0.8	$\mathbf{0.901 \pm 0.110}$	1.066 ± 0.243	0.211 ± 0.122	$\mathbf{0.239 \pm 0.224}$	$\mathbf{0.273 \pm 0.196}$	0.306 ± 0.149
Separated	0.9	$\mathbf{0.902 \pm 0.109}$	1.062 ± 0.239	0.214 ± 0.118	$\mathbf{0.235 \pm 0.222}$	0.274 ± 0.194	0.308 ± 0.147
Separated	1.0	$\underline{\mathbf{0.910 \pm 0.099}}$	1.562 ± 0.080	-0.146 ± 0.063	0.008 ± 0.005	0.033 ± 0.013	0.076 ± 0.020
Shared (proposed)	0.8	$\mathbf{0.909 \pm 0.100}$	$\underline{\mathbf{1.028 \pm 0.224}}$	$\underline{\mathbf{0.242 \pm 0.104}}$	$\mathbf{0.248 \pm 0.217}$	$\underline{\mathbf{0.291 \pm 0.189}}$	$\underline{\mathbf{0.328 \pm 0.140}}$

A Ablation Study: Shared Head vs. Separated Head

To investigate whether the shared-head design of our self-explaining model harms classification and explanation quality, we compare the proposed shared-head architecture with a separated-head variant. In the shared-head model (proposed), the domain-specific head $g_{\boldsymbol{W}}$ is shared between classification (mapping the `[CLS]` token to logits) and explanation (mapping patch tokens to attributions), and is kept frozen from the pretrained model. In the separated-head model, a dedicated explanation head $g_{\boldsymbol{W}_{\text{exp}}}$ is introduced separately from the classification head $g_{\boldsymbol{W}}$, where both heads are initialized from the pretrained $g_{\boldsymbol{W}}$. The entire model, including the feature extractor and both heads, is then trained jointly.

Table 4 shows the results averaged across the 10 target datasets when finetuning on each target dataset with explanation supervision (target+X). At $\alpha = 0.8$ (used as the default in the main paper), the shared-head model attains a lower E-RMSE and higher Rank corr., IoU@5, and IoU@10 than the separated-head model at every $\alpha \in \{0.0, 0.1, \ldots, 1.0\}$, with comparable classification accuracy. No separated-head setting matches the shared-head model on accuracy and explanation quality at once. The settings whose accuracy shows no significant difference from the best ($\alpha \geq 0.8$) are significantly worse on E-RMSE, Rank corr., and IoU@10, and the settings that are close on explanation quality ($\alpha \leq 0.5$) are significantly less accurate. On IoU@1, the separated-head model at $\alpha = 0$ is slightly higher (0.250 vs. 0.248), with no significant difference. The shared-head model attains this without any training of the head, whereas the separated-head model trains $g_{\boldsymbol{W}_{\text{exp}}}$ with explanation supervision.

Why the gap reflects head trainability, not architecture Because $g_{\boldsymbol{W}}$ and $g_{\boldsymbol{W}_{\text{exp}}}$ are initialized to the same pretrained weights and both heads are linear, freezing both heads in the separated-head model makes it mathematically equivalent to the shared-head model. The two heads then remain weight-identical throughout training, producing the same outputs and the same gradient flow into the feature extractor. The shared-head row in Table 4 can therefore be read as the architecture-only counterpart of the separated-head condition with both heads frozen, and the gap between this row and the trainable-head separated rows isolates the effect of head trainability rather than the architectural choice of separating the heads.

B Circuit Overlap between Classification and Explanation

Our self-explaining model in Section 4.1 produces the classification logits and the patch-level attributions in parallel from the backbone features. A natural concern is whether this parallel-output structure is only a surface-level coupling at the final head, while classification and explanation are actually computed by disjoint internal sub-circuits of the backbone. In that case, the two capabilities would not interact mechanistically inside the model. In this appendix, we test this concern by quantifying the head-level overlap of classification and explanation circuits in self-explaining ViT image classifiers, and further examine whether this overlap is induced by the explanation supervision

$\mathcal{L}_{\text{exp}}$ or is already a property of the architecture before any explanation training. Appendices B.1 and B.2 use, on both backbones, the 10 targets of Section 5.1 with CIFAR10 in place of CIFAR100, as in Section 5.2. The target sets of Appendix B.3 are stated there.

Head-level effect Let L and H denote the number of transformer layers and the number of attention heads per layer, respectively (e.g., $L = H = 12$ for ViT-B/32 and $L = 24, H = 16$ for ViT-L/14). We follow the standard multi-head attention notation: within each transformer block, the attention sublayer concatenates its H per-head outputs (each in $\mathbb{R}^{d_h}$, with $d_h = d_{\text{model}}/H$ and d_{model} the residual-stream dimension) and applies an output projection $\boldsymbol{W}_O \in \mathbb{R}^{d_{\text{model}} \times d_{\text{model}}}$ to produce the post-projection attention output of the block [56]. For each dataset D and each loss $\mathcal{L} \in \{\mathcal{L}_{\text{cls}}, \mathcal{L}_{\text{exp}}\}$, we attach a forward hook to the post-projection attention output $\boldsymbol{y}$ of every block and back-propagate $\mathcal{L}$ over a fixed batch of 256 samples. With the row-vector convention $\boldsymbol{y} = \boldsymbol{z}\boldsymbol{W}_O^\top + \boldsymbol{b}_O$, we recover the pre-projection concatenated-head activation $\boldsymbol{z} = (\boldsymbol{y} - \boldsymbol{b}_O)(\boldsymbol{W}_O^\top)^+$ with the pseudoinverse and its gradient $\nabla_{\boldsymbol{z}}\mathcal{L} = (\nabla_{\boldsymbol{y}}\mathcal{L})\boldsymbol{W}_O$. Reshaping both to $\mathbb{R}^{H \times d_h}$ gives the activation $\boldsymbol{z}_{b,s,h}$ of head h at sample b and token s. We average the absolute per-token attribution score of Michel et al. [42] over the B samples and S tokens:

$$e_D[\ell, h] = \frac{1}{BS} \sum_{b,s} \left| \boldsymbol{z}_{b,s,h} \cdot \nabla_{\boldsymbol{z}_{b,s,h}} \mathcal{L} \right|, \tag{9}$$

and write e_D^{cls} and e_D^{exp} for $\mathcal{L}_{\text{cls}}$ and $\mathcal{L}_{\text{exp}}$. The absolute inner product is a first-order estimate of the magnitude of the change in $\mathcal{L}$ when that head's output at that token is zeroed. Taking the absolute value per token before averaging differs from the head-level gate of Michel et al. [42], whose gradient sums the token contributions before the absolute value. Rescaling each gradient by the corresponding activation suppresses the early-layer mass that raw gradient norms accumulate via back-propagation chaining and is a common choice for per-head importance estimation in the mechanistic interpretability literature.

Coupling metric For each dataset D, let $\boldsymbol{e}_D^{\text{cls}}, \boldsymbol{e}_D^{\text{exp}} \in \mathbb{R}^{LH}$ denote the flattened per-head effects for $\mathcal{L}_{\text{cls}}$ and $\mathcal{L}_{\text{exp}}$. We define the *circuit-overlap coefficient* as the Pearson correlation

$$\rho_{\text{circ}}[D] = \text{Pearson}(\boldsymbol{e}_D^{\text{cls}}, \boldsymbol{e}_D^{\text{exp}}). \tag{10}$$

A value close to 1 indicates that the same heads carry both classification and explanation signal; a value close to 0 indicates head-level specialization between the two objectives. We further define a *per-layer* version

$$\rho_{\text{circ}}^{(\ell)}[D] = \text{Pearson}\big(\boldsymbol{e}_D^{\text{cls}}[\ell, :],\ \boldsymbol{e}_D^{\text{exp}}[\ell, :]\big), \qquad \ell \in \{1, \ldots, L\}, \tag{11}$$

which restricts the correlation to the H heads within a single transformer layer ℓ and is computed from the same per-head attribution effects as $\rho_{\text{circ}}[D]$.

B.1 Classification and Explanation Heads Largely Overlap

Computing $\rho_{\text{circ}}[D]$ on the self-explaining checkpoint $\boldsymbol{\theta}_{\text{ft}\star}^D$ across the 10 target datasets yields a positive coupling on every dataset, with mean $\bar{\rho}_{\text{circ}} = 0.49$ on ViT-B/32 (range $[0.22, 0.75]$) and $\bar{\rho}_{\text{circ}} = 0.90$ on ViT-L/14 (range $[0.83, 0.96]$). The only marginally low value is GTSRB on ViT-B/32 ($\rho_{\text{circ}} = 0.22$). It is fully resolved on ViT-L/14 ($\rho_{\text{circ}} = 0.96$), consistent with capacity scaling subsuming the apparent specialization. This comparison also differs in explanation supervision (Kernel SHAP for ViT-B/32 vs. Grad-CAM for ViT-L/14, see Appendix E). Grad-CAM by construction couples the explanation circuit to the classification gradients and may therefore contribute to the higher overlap on ViT-L/14.

Layer-wise profile Computing $\rho_{\text{circ}}^{(\ell)}$ within each transformer layer reveals a consistent pattern across both backbones: early-to-mid layers exhibit near-perfect coupling, while late layers progressively decouple. On ViT-B/32, layers $\ell \in \{1, \ldots, 6\}$ attain $\rho_{\text{circ}}^{(\ell)} \in [0.90, 0.98]$, after which coupling drops to $\rho_{\text{circ}}^{(12)} = 0.43$ (Figure 5). The same pattern holds on ViT-L/14. Layers $\ell \in \{1, \ldots, 9\}$ stay above 0.97, coupling then declines gradually, and the last two blocks reach $\rho_{\text{circ}}^{(23)} = 0.43$ and $\rho_{\text{circ}}^{(24)} = 0.49$ (figure omitted for space). This early-shared / late-diverged structure is consistent with the finding of Gandelsman et al. [20] that CLS-token direct effects in CLIP-trained ViTs concentrate in the last attention layers. This suggests that the network reuses a shared backbone for both objectives until late blocks, where head-level specialization between the classification and explanation circuits emerges.

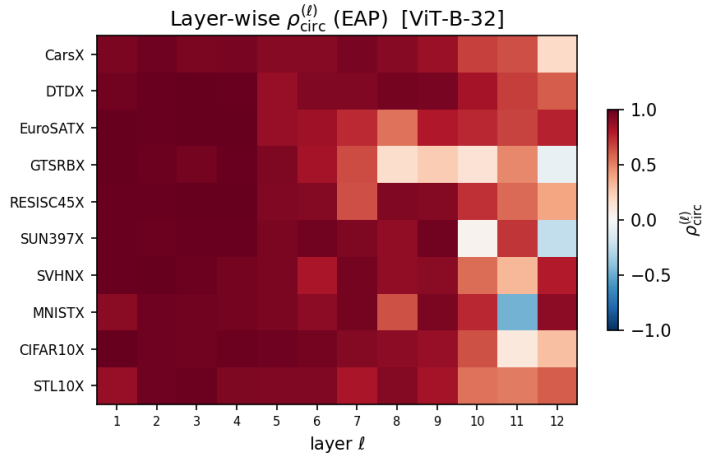
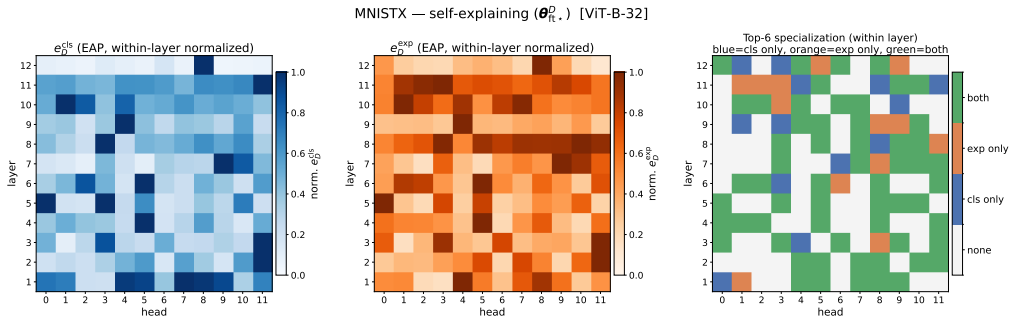


Figure 5: Layer-wise circuit-overlap profile $\rho^{(\ell)}_{\text{circ}}$ (per-head attribution) on the ViT-B/32 self-explaining checkpoint. Each row corresponds to one of the 10 target datasets and each column to one of the 12 transformer layers; cell color encodes $\rho^{(\ell)}_{\text{circ}}$ for that (dataset, layer) pair. The heatmap is uniformly dark in the early-to-mid layers and lightens toward the final layers ($\ell \in \{10, 11, 12\}$). This is the early-shared / late-diverged structure described in the text.

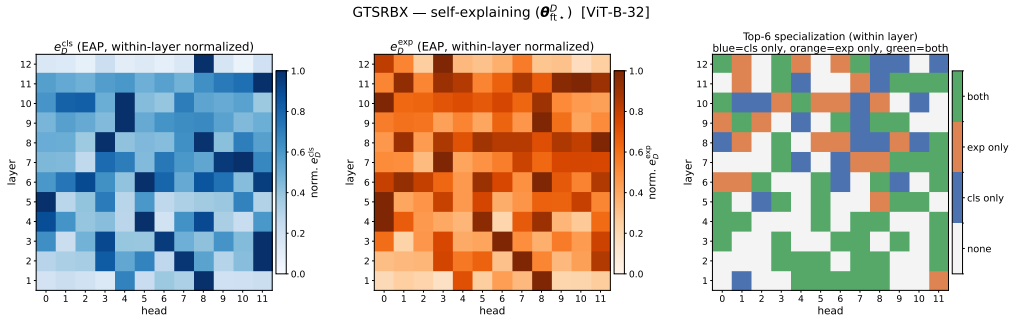
Per-dataset head-level visualizations Figure 6 shows the per-head attribution effects $e^{\text{cls}}_D[\ell, h]$ and $e^{\text{exp}}_D[\ell, h]$ for the highest- and lowest-coupling target datasets on ViT-B/32 (MNIST and GTSRB, respectively). The third panel of each row marks, for each transformer layer, the heads that fall into the within-layer top-k for e^{cls}_D only, e^{exp}_D only, or both, with $k = 6$ chosen for visual clarity. These maps show where the shared (green) heads lie. They are not by themselves evidence of overlap, because two random top-6 sets out of 12 heads share at least one head with probability above 0.99, and the quantitative evidence is ρ_{circ}. The analogous maps for ViT-L/14 are omitted for space, and we compare the backbones with the layer-wise $\rho^{(\ell)}_{\text{circ}}$ instead. GTSRB inverts its rank between backbones. It is the lowest-coupling dataset on ViT-B/32 ($\rho_{\text{circ}} = 0.22$) but the highest-coupling one on ViT-L/14 ($\rho_{\text{circ}} = 0.96$). On ViT-B/32, its layer-wise coupling falls below 0.5 from layer 8 onward (Figure 6(b)). On ViT-L/14, it stays at or above 0.74 up to layer 22 and drops only in the last two blocks (-0.10 and 0.28).

B.2 Circuit Overlap Is Present before Explanation Training

A natural hypothesis is that the strong cls–exp coupling reported above is itself induced by jointly optimizing $\mathcal{L}_{\text{cls}}$ and $\mathcal{L}_{\text{exp}}$ during finetuning. We test this hypothesis by repeating the measurement on three checkpoints per dataset: (i) the classification-only finetuned model $\boldsymbol{\theta}^D_{\text{ft}}$, which has never seen explanation supervision; (ii) the self-explaining model $\boldsymbol{\theta}^D_{\text{ft}\star}$, which has been jointly optimized for classification and explanation; and (iii) the transferred model $\tilde{\boldsymbol{\theta}}^T_{\text{ft}\star}$ obtained by task arithmetic (8), with the source fixed to ImageNet+X for every target T. Figure 7 reports ρ_{circ} on ViT-B/32; the corresponding ViT-L/14 figure is omitted for space and the L/14 statistics are summarized in text below. On both backbones, conditions (i) and (ii) show no detectable difference (B-32: $\bar{\rho}_{\text{circ}} = 0.54$ vs. 0.49, paired Wilcoxon $p = 0.63$; L-14: 0.90 vs. 0.90, $p = 0.70$). The cls–exp coupling thus already exists in the classification-only checkpoint, and this evaluation detects no additional coupling from joint optimization. The transferred condition (iii), in contrast, changes the coupling. On ViT-L/14, the target checkpoints attain $\bar{\rho}_{\text{circ}} = 0.90$ in both (i) and (ii), but the ImageNet-source self-explaining checkpoint attains only 0.58. The mean of the transferred condition drops to 0.77, toward the source ($p = 0.006$ vs. (i) and $p = 0.002$ vs. (ii)). On ViT-B/32, the source coupling (0.56) is close to the target means, and the transferred mean rises to 0.60 ($p = 0.06$ vs. (i) and $p = 0.05$ vs. (ii)). This rise passes the source value, so a pull toward the source is clear only on ViT-L/14, where the source and target coupling levels differ. This source-pull effect on ViT-L/14 is consistent with the parameter-arithmetic interpretation of task arithmetic, in which $\tilde{\boldsymbol{\theta}}^T_{\text{ft}\star}$ inherits part of the head-level structure of the source.



(a) MNIST, the highest-coupling target on ViT-B/32 ($\rho_{\mathrm{circ}} = 0.75$).



(b) GTSRB, the lowest-coupling target on ViT-B/32 ($\rho_{\mathrm{circ}} = 0.22$).

Figure 6: Per-head attribution effects on the ViT-B/32 self-explaining checkpoint for the two extreme-coupling target datasets. Left and middle panels: within-layer normalized $e_D^{\mathrm{cls}}[\ell, h]$ and $e_D^{\mathrm{exp}}[\ell, h]$ over the $L = 12$ layers and $H = 12$ heads. Right panel: within-layer top-6 specialization map (blue: cls-only, orange: exp-only, green: both, white: neither).

Since the coupling is present before any explanation supervision, it is not created by training the explanation output, which is consistent with both tokens flowing through a shared attention backbone. This answers the concern raised at the start of this appendix at a descriptive level, since the two parallel outputs share much of their head-level computation inside the backbone. Causal evidence on where the transferred effect is mediated comes from the parameter interventions of Appendices B.3 and K.

B.3 Where the Transferred Effect Is Mediated

The analyses above describe the self-explaining checkpoints. This subsection asks where the explainability vector $\boldsymbol{\tau}_\star^{\mathrm{S}}$ itself acts when it is added to a target. We use three analyses. The first measures how the norm of $\boldsymbol{\tau}_\star^{\mathrm{S}}$ is distributed over the blocks. The second adds only part of $\boldsymbol{\tau}_\star^{\mathrm{S}}$ to the target and checks how much of the improvement remains, using the layer-restricted controls of Appendix K. The third measures how the attributions of individual heads change when $\boldsymbol{\tau}_\star^{\mathrm{S}}$ is added. The first and third show where the change is large, but not whether that change matters for the explanations. Only the second tests this directly.

Distribution of the vector norm Decomposing the squared norm of $\boldsymbol{\tau}_\star^{\mathrm{S}}$ ($\alpha = 0.8$, source ImageNet+X) exactly over transformer blocks and module types, the last six blocks carry 59.2% of $\|\boldsymbol{\tau}_\star^{\mathrm{S}}\|^2$ (first six blocks 39.4%, non-block parameters 1.4%). Table 5 shows that this back-loading comes mainly from the pre-MLP LayerNorm. 89.1% of the squared norm of its change sits in the last six blocks, concentrated in blocks 10–11, and it accounts for 81.6% of the difference in squared norm between the two halves. Without it, the last six blocks hold 52.3% of the squared norm of the remaining block parameters. The attention projections are close to an even split.

Layer-restricted intervention Adding only the part of $\boldsymbol{\tau}_\star^{\mathrm{S}}$ that lies in the last six blocks recovers 84.0% of the Rank corr. improvement and 81.9% of the IoU@10 improvement of the full vector over the no-vector model on 10-target means. These means are over the CIFAR10 set of Appendix K, in which CIFAR10 replaces SUN397. MNIST is the outlier at 58% and 51%. The part in the first six blocks, in contrast, recovers essentially nothing and is indistinguishable from adding no vector (Table 16). On 10-target means, neither restricted vector changes classification accuracy by more

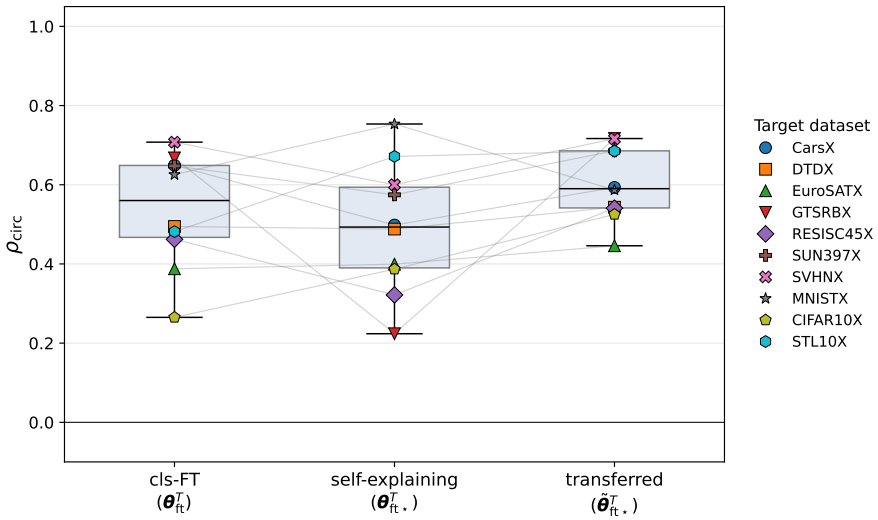


Figure 7: Circuit-overlap coefficient ρ_{circ} on ViT-B/32 measured under per-head attribution for three training conditions: cls-only finetuned $\boldsymbol{\theta}^T_{\text{ft}}$, self-explaining $\boldsymbol{\theta}^T_{\text{ft}\star}$, and transferred $\tilde{\boldsymbol{\theta}}^T_{\text{ft}\star}$ (with source fixed to $S =$ ImageNet+X). Each marker color/shape encodes one of the 10 target datasets (legend) and gray lines connect the same target across conditions. Conditions (i) and (ii) show no detectable difference, while the transferred condition has a slightly higher mean (0.60) than both and than the ImageNet-source coupling level ($\bar{\rho}^{S=\text{ImageNet+X}}_{\text{circ}} = 0.56$). The 10 targets of this analysis include CIFAR10 in place of CIFAR100 of Table 1.

Table 5: Share of each module type's squared ℓ_2 norm of $\boldsymbol{\tau}^{\text{S}}_\star$ that lies in the last six transformer blocks of ViT-B/32 (an even split would give 50%). The two norm columns are ℓ_2 norms, and the share is computed from their squares. Modules are the pre-MLP and pre-attention LayerNorms, the MLP down- and up-projections, and the attention input (QKV) and output projections.

Module	Norm, blocks 1–6	Norm, blocks 7–12	Share in blocks 7–12
LayerNorm before MLP	1.572	4.500	89.1%
LayerNorm before attention	1.319	1.959	68.8%
MLP down-projection	2.917	3.392	57.5%
MLP up-projection	3.854	4.033	52.3%
Attention QKV projection	3.594	3.300	45.7%
Attention output projection	1.669	1.520	45.3%

than 0.3 points. Most of the transferred effect is therefore mediated by parameter changes in the later blocks. The two restricted vectors do not add up to the full vector, however. The first six blocks recover nothing alone, yet the full vector exceeds the last six blocks by 16.0% of the Rank corr. improvement and 18.1% of the IoU@10 improvement. We do not separate this remainder into the contributions of the first six blocks, the non-block parameters, and their interactions.

Per-head attribution changes Using the per-head attribution effect $e_D[\ell, h]$ in (9), we measured how the effects change when $0.7\,\boldsymbol{\tau}^{\text{S}}_\star$ is added to $\boldsymbol{\theta}^{\text{T}}_{\text{ft}}$ on three targets (Cars, GTSRB, and RESISC45), using 256 samples each. Table 6 reports, for each loss, the share of the summed absolute change that falls in the last six blocks, in the last two blocks, and in the five most-changed of the 144 heads. The change in the explanation-loss effects leans mildly toward the later blocks. The last six blocks hold 53.5% of it on average, against 50% for a uniform spread, and the last two blocks hold 19.7%, against 16.7%. The five most-changed heads account for 10.8% of the change, against 3.5% for a uniform spread. For GTSRB and RESISC45, four of these five heads lie in the last block, and for Cars two do. Most of the change is therefore spread over many heads, although the largest individual changes sit in the last block. The classification-loss effects shift mildly toward the earlier blocks (43.6% in the last six). On GTSRB and RESISC45, their summed absolute change is three to four orders of magnitude smaller than that of the explanation-loss effects, so their shares on these two targets should not be over-read. This per-head picture is weaker and less clean than the norm and

Table 6: Where the per-head attribution effects $e_D[\ell, h]$ change when $0.7\,\boldsymbol{\tau}_\star^{\mathrm{S}}$ is added to $\boldsymbol{\theta}_{\mathrm{ft}}^{\mathrm{T}}$ (256 samples per target). Each entry is the share of the summed absolute change over the 144 heads that falls in the last six blocks (blocks 7–12), in the last two blocks (blocks 11–12), or in the five heads with the largest absolute change. The last row gives the shares for a uniform spread.

	Explanation loss			Classification loss		
Target	Blocks 7–12	Blocks 11–12	Top-5 heads	Blocks 7–12	Blocks 11–12	Top-5 heads
Cars	47.7%	15.1%	8.7%	43.4%	8.7%	6.8%
GTSRB	54.0%	20.9%	11.1%	45.2%	13.8%	10.1%
RESISC45	58.8%	23.1%	12.5%	42.3%	8.7%	8.0%
Mean	53.5%	19.7%	10.8%	43.6%	10.4%	8.3%
Uniform	50.0%	16.7%	3.5%	50.0%	16.7%	3.5%

layer-restriction results above, and we read it as partial corroboration rather than as an independent localization.

Interpretation The part of $\boldsymbol{\tau}_\star^{\mathrm{S}}$ in the early blocks has no measurable effect on its own, and these blocks are shared between classification and explanation (Appendix B.1). The explanation mechanism is concentrated in the later blocks, where most of the vector's effect arises, on top of the target's own representations. This is consistent with the source-pull effect of Appendix B.2, where, on ViT-L/14, adding the vector shifts the transferred model's head-level coupling toward that of the source, and with the two-task experiment of Appendix N, where the same vector yields task-appropriate attributions on two tasks. These analyses establish where the transferred effect is mediated, not a complete causal account of how the attributions are computed.

C Feature Pool and ViT-L/14 Predictability Analysis

This appendix characterizes when and how the predictability examined in Section 5.2 holds, with (i) formal definitions for the six features, (ii) the hypothesis-level rationale for the six features and the collinearity check that consolidates redundant ℓ_2/cosine variants, and (iii) results on ViT-L/14 with an oracle-normalized target R_q that complements the raw Δ_q target.

Formal definition of the predictor features The classification-circuit features (rows 1–3 of Table 7) build on the head-level attribution effect vectors $\boldsymbol{e}^{\mathrm{cls}}(\boldsymbol{\theta}; \mathcal{D}), \boldsymbol{e}^{\mathrm{exp}}(\boldsymbol{\theta}; \mathcal{D}) \in \mathbb{R}^{LH}$ defined in Appendix B (same forward-hook setup, 256 samples, per-head attribution of Michel et al. [42], flattened in row-major order), made explicit in the model parameter $\boldsymbol{\theta}$ and the dataset $\mathcal{D}$ used to compute the loss. We use $\boldsymbol{a} = \boldsymbol{e}^{\mathrm{cls}}(\tilde{\boldsymbol{\theta}}_{\mathrm{ft}\star}^{\mathrm{T}}; \mathcal{D}^{\mathrm{T}})$, $\boldsymbol{b} = \boldsymbol{e}^{\mathrm{cls}}(\boldsymbol{\theta}_{\mathrm{ft}}^{\mathrm{T}}; \mathcal{D}^{\mathrm{T}})$, and $\boldsymbol{c} = \boldsymbol{e}^{\mathrm{exp}}(\boldsymbol{\theta}_{\mathrm{ft}\star}^{\mathrm{S}}; \mathcal{D}^{\mathrm{S}})$ as the transferred-cls, target-cls, and source-exp effect vectors respectively, where $\tilde{\boldsymbol{\theta}}_{\mathrm{ft}\star}^{\mathrm{T}}$ is the transferred model of (8), write $\mathrm{Top}_k(\cdot)$ for the index set of the k largest entries with $k = 16$ for ViT-B/32 and $k = 32$ for ViT-L/14, and use the standard Jaccard index $J(A, B) = |A \cap B|/|A \cup B|$. The $\boldsymbol{\tau}$-output-shift features (rows 4–6) build on [CLS] feature shifts on unlabeled target images. Let $f_{\boldsymbol{\theta}}^{[\mathtt{CLS}]}(\boldsymbol{x}) \in \mathbb{R}^K$ denote the [CLS] output of the feature extractor with parameters $\boldsymbol{\theta}$ on image $\boldsymbol{x}$, and let $\mathcal{X}_{\mathrm{unlab}}^{\mathrm{T}} = \{\boldsymbol{x}_i\}_{i=1}^{N_T}$ be unlabeled target images. Writing $\boldsymbol{h}_i^{\mathrm{base}} = f_{\boldsymbol{\theta}_{\mathrm{base}}}^{[\mathtt{CLS}]}(\boldsymbol{x}_i)$ for the zero-shot backbone and $\boldsymbol{h}_i^{\mathrm{taued}} = f_{\boldsymbol{\theta}_{\mathrm{base}}+\boldsymbol{\tau}_\star^{\mathrm{S}}}^{[\mathtt{CLS}]}(\boldsymbol{x}_i)$ for the $\boldsymbol{\tau}$-perturbed backbone, define the per-image shift $\boldsymbol{\delta}_i = \boldsymbol{h}_i^{\mathrm{taued}} - \boldsymbol{h}_i^{\mathrm{base}}$ and the image-mean centroid $\bar{\boldsymbol{h}}^{\bullet} = \frac{1}{N_T}\sum_{i=1}^{N_T} \boldsymbol{h}_i^{\bullet}$ for $\bullet \in \{\mathrm{base}, \mathrm{taued}\}$. Row 6 averages features across images first and then takes the cosine, factoring out per-image directional noise and isolating the systematic $\boldsymbol{\tau}$-induced bias.

Why these six features The classification-circuit features (rows 1–3) test how the head-level classification structure of the target relates to the transferred model and to the source explanation circuit. They are the transferred–target cls-effect cosine, the top-k Jaccard overlap, and the source-exp vs. target-cls cross-cosine, all motivated by the head-level overlap of Appendix B. The $\boldsymbol{\tau}$-output-shift features (rows 4–6) test whether $\boldsymbol{\tau}_\star^{\mathrm{S}}$ non-trivially perturbs target image content. They are the per-image ℓ_2 shift mean and std on the zero-shot backbone, plus the centroid-cosine that filters per-image directional noise. Within each family we considered ℓ_2-magnitude and cosine variants in parallel. We collapsed pairs with $|r| \gtrsim 0.8$, where r is the Pearson correlation between two candidate features computed once on the 10 pairs of each backbone (no target variable is used). For example, cos

Table 7: Formal definition of the six predictor features used in Section 4.4. Classification-circuit features compare flattened per-head attribution effect vectors. $\boldsymbol{\tau}$-output-shift features compare [CLS] representations of unlabeled target images under a frozen zero-shot backbone with and without adding $\boldsymbol{\tau}^{\mathrm{S}}_{\star}$. The index $i \in \{1, \ldots, N_T\}$ runs over the unlabeled target images $\boldsymbol{x}_i \in \mathcal{X}^{\mathrm{T}}_{\mathrm{unlab}}$. We use $k = 16$ (ViT-B/32) and $k = 32$ (ViT-L/14) for the top-k Jaccard.

#	Class	Identifier	Definition
1	cls-circuit	`cos_tr_vs_T_ft_cls`	$\cos(\boldsymbol{a}, \boldsymbol{b})$
2	cls-circuit	`jacc_tr_vs_T_ft_cls`	$J\big(\mathrm{Top}_k(\boldsymbol{a}),\ \mathrm{Top}_k(\boldsymbol{b})\big)$
3	cls-circuit	`cos_S_ftstar_exp_vs_T_ft_cls`	$\cos(\boldsymbol{c}, \boldsymbol{b})$
4	τ-shift	`tau_feat_shift_l2_mean`	$\frac{1}{N_T}\sum_{i=1}^{N_T} \lVert\boldsymbol{\delta}_i\rVert_2$
5	τ-shift	`tau_feat_shift_l2_std`	$\mathrm{std}_i\big(\lVert\boldsymbol{\delta}_i\rVert_2\big)$
6	τ-shift	`tau_centroid_cos`	$\cos(\bar{\boldsymbol{h}}^{\mathrm{base}}, \bar{\boldsymbol{h}}^{\mathrm{taued}})$

Table 8: Held-out Pearson correlation (LOTO, $N = 10$) on ViT-B/32, full grid corresponding to Table 3.

Method	E-MSE	IoU@1	IoU@5	IoU@10	Rank corr.
Lasso	$+\mathbf{0.92}$	$+0.19$	$+\mathbf{0.70}$	$+0.81$	$+0.77$
Ridge	$+0.91$	$+\mathbf{0.33}$	$+0.63$	$+\mathbf{0.85}$	$+\mathbf{0.82}$

tracks $-\ell_2$ at $|r| \approx 0.80$–0.96 across backbones, and the complement-of-top-k tracks the Jaccard at $r = -1.00$ by construction. Keeping one representative per cluster yields the 3+3 features of Table 7. The pruning uses feature–feature correlations only and is therefore target-independent. The target Δ_q is used only for regression fitting and regularization-strength selection, both of which use only the training targets of each leave-one-target-out fold. As a post-hoc observation, feeding all variants to the regularizer instead of the pruned pool gives $r = +0.80$ rather than $+0.92$ for ViT-B/32 Lasso on E-MSE and no non-trivial ViT-L/14 result under R_q, which we attribute to near-duplicate features splitting the regularization budget. This observation was not used to choose the pool.

Full ViT-B/32 results For completeness, Table 8 reports held-out Pearson r for ViT-B/32 across all five explanation-quality metrics. The main paper (Table 3) shows three representative metrics under the page budget.

ViT-L/14 results with an oracle-normalized target R_q On ViT-L/14, the raw Δ_q target is dominated by per-target variance, with GTSRB as an outlier whose large Δ_q disproportionately drives the regression. We therefore evaluate ViT-L/14 with an oracle-normalized target that absorbs per-target scale. For each (S, T) pair, in addition to $b_q = q(\boldsymbol{\theta}^{\mathrm{T}}_{\mathrm{ft}})$ and $a_q = q(\tilde{\boldsymbol{\theta}}^{\mathrm{T}}_{\mathrm{ft}\star})$ from Section 5.2, let $o_q = q(\boldsymbol{\theta}^{\mathrm{T}}_{\mathrm{ft}\star})$ denote the target oracle, which is the explanation-quality score of the target self-explaining model trained with explanation supervision (the "target+X" model of Section 5.1). We define

$$R_q(S, T) = \frac{b_q - a_q}{b_q - o_q}, \tag{12}$$

the fraction of the explanation-quality gap from the target classification-only baseline $\boldsymbol{\theta}^{\mathrm{T}}_{\mathrm{ft}}$ to the target oracle $\boldsymbol{\theta}^{\mathrm{T}}_{\mathrm{ft}\star}$ that is closed by the transferred model $\tilde{\boldsymbol{\theta}}^{\mathrm{T}}_{\mathrm{ft}\star}$. When the oracle improves on the baseline, which holds for every target and metric here, $R_q = 1$ matches the oracle, $R_q = 0$ indicates no improvement, $R_q < 0$ indicates degradation, and $R_q > 1$ exceeds the oracle. The oracle score o_q enters only when the predictor is fitted and evaluated. Predicting the outcome on a new target uses the six model-internal features alone and requires no target-side explanation supervision.

Table 9 reports held-out Pearson r between predicted and actual R_q on ViT-L/14 with the same 6-feature pool and the same nested-LOTO protocol as Section 5.2. Three of the five explanation-quality metrics are statistically significant under R_q at uncorrected one-sided permutation $p < 0.05$, computed as in Section 5.2: IoU@1 (Lasso $r = +0.82$, $p = 0.016$), E-MSE (Ridge $r = +0.81$, $p = 0.034$), and Rank corr. (Lasso $r = +0.70$, $p = 0.046$). The remaining two metrics, IoU@5 and IoU@10, are not predictable under R_q on this backbone. This residual backbone–metric specificity is discussed in Appendix O.

Table 9: Held-out Pearson correlation (LOTO, $N = 10$) on ViT-L/14, using the oracle-normalized target R_q defined in Eq. (12). Best per metric is in **bold**.

Method	E-MSE	IoU@1	IoU@5	IoU@10	Rank corr.
Lasso	$+0.81$	$\mathbf{+0.82}$	$+0.21$	-0.06	$\mathbf{+0.70}$
Ridge	$\mathbf{+0.81}$	$+0.81$	$\mathbf{+0.23}$	$\mathbf{-0.03}$	$+0.49$

D Computing Environment for Experiments

The experiments were conducted in parallel on Rocky Linux machines with two Intel Xeon Platinum 8558 2.1GHz (48 cores) CPUs, 2,048 GB of RAM, and eight NVIDIA H200 SXM GPUs with 141GB of memory. The code was implemented in Python 3.9.21 using PyTorch 2.6. The additional experiments in Appendices B.3 and J–N and the leave-one-source-out evaluation in Appendix F.2 were run on NVIDIA RTX 6000 Ada GPUs with 48GB of memory, using the same code base.

E Experiments with ViT-L/14

To validate the effectiveness of the proposed method with a larger vision transformer, we conduct experiments using ViT-L/14 as the backbone of the self-explaining image classifier.

E.1 Setup

The model input is a 224×224 image, which is split into 16×16 patches of size 14×14. Therefore, the number of patches M is 256.

The GT patch-level attributions for ImageNet and the 10 target datasets, Cars [36], DTD [14], EuroSAT [25], GTSRB [55], RESISC45 [12], SUN397 [63], SVHN [45], MNIST [38], CIFAR100 [37], and STL10 [15], are obtained by applying Grad-CAM [51], implemented by `pytorch-grad-cam` [21], to the finetuned ViT-L/14-based image classifiers for each dataset. Specifically, we apply Grad-CAM to the layer normalization of the final residual block in the Vision Transformer encoder. The obtained attributions are pixel-wise values. We therefore aggregate them into patch-level attribution values by averaging the pixel-wise attribution values within each patch.

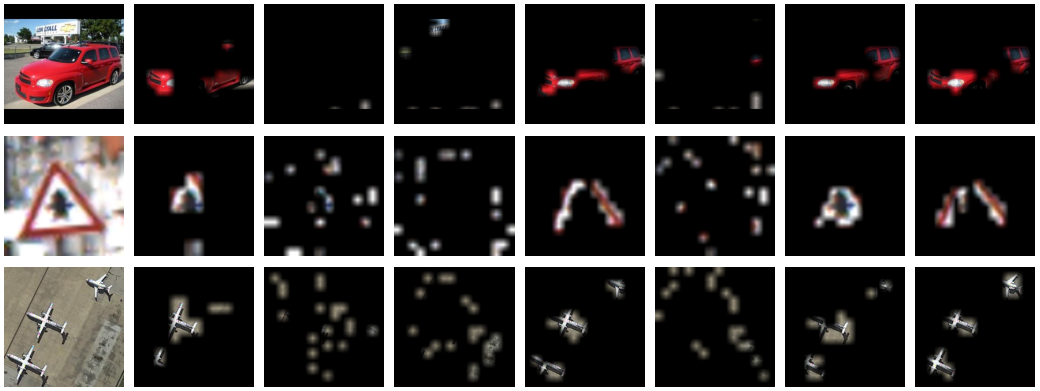
E.2 Quantitative Results

Table 10 shows the average performances of the explainability transferred by the proposed method with $\alpha = 0.8$ in (4) and $\lambda_1 = 1.0, \lambda_2 = 0.7$ in (8), in comparison to the baseline models with the parameters $\boldsymbol{\theta}_{\text{base}}$, $\boldsymbol{\theta}^{\text{S}}_{\text{ft}}$, $\boldsymbol{\theta}^{\text{S}}_{\text{ft}\star}$, and $\boldsymbol{\theta}^{\text{T}}_{\text{ft}}$. For reference, we also show the performances of the model finetuned on the target dataset with explanation supervision, i.e., the oracle $\boldsymbol{\theta}^{\text{T}}_{\text{ft}\star}$. The table shows that compared to the baselines, the proposed method achieves the second-best accuracy while achieving the best or second-best explainability performances. The accuracy gap to the oracle $\boldsymbol{\theta}^{\text{T}}_{\text{ft}\star}$ is 0.015 ($\approx 1.6\%$ relative), and the E-RMSE gap is $\approx 1.4\%$ relative. The proposed method thus transfers explainability acquired on ImageNet+X while largely preserving classification accuracy. The method with the best accuracy is the model $\boldsymbol{\theta}^{\text{T}}_{\text{ft}}$ finetuned on the target dataset without explanation supervision. However, this model shows the worst or second-worst explainability performances. In contrast, the best Rank corr. is achieved by the model $\boldsymbol{\theta}^{\text{S}}_{\text{ft}\star}$ finetuned on ImageNet+X, but this model has the lowest accuracy. The proposed method combines the high target-domain accuracy of $\boldsymbol{\theta}^{\text{T}}_{\text{ft}}$ with the explainability vector extracted from $\boldsymbol{\theta}^{\text{S}}_{\text{ft}\star}$, attaining classification accuracy close to that of $\boldsymbol{\theta}^{\text{T}}_{\text{ft}}$ (0.928 vs. 0.940) and explanation quality close to the oracle $\boldsymbol{\theta}^{\text{T}}_{\text{ft}\star}$.

Overall, the classification accuracy is higher than that of the ViT-B/32-based models shown in Table 1, consistent with the larger model capacity of ViT-L/14. On the other hand, the explainability performance scores are lower than those of the ViT-B/32-based models. The two settings differ in the backbone, the GT generator (Grad-CAM instead of Kernel SHAP), and the number of patches (256 instead of 49), so the scores are not directly comparable. Possible causes include the different GT generator and the finer patch grid, which also lowers the chance level of IoU@1 from $1/49$ to $1/256$.

Table 10: Average performances and their standard deviations of ViT-L/14-based self-explaining models over the 10 target datasets when transferring explainability learned on ImageNet+X, compared with those of baselines. The proposed method's results are reported at $\alpha = 0.8$ and $\lambda_2 = 0.7$ (with $\lambda_1 = 1.0$ as in the common setup). Here, the last row shows the oracle performances achieved by fine-tuning on the target datasets with explanation supervision (target+X). **Bold and underlined values** indicate the best among the proposed method and the baselines for each metric, while **bold values** indicate that there is no statistically significant difference from the best, as determined by a paired t-test at a significance level of 0.05.

	Param.	Accuracy (↑)	E-RMSE (↓)	Rank corr. (↑)	IoU@1 (↑)	IoU@5 (↑)	IoU@10 (↑)
Zero-shot	$\boldsymbol{\theta}_{\text{base}}$	0.630 ± 0.151	1.470 ± 0.043	-0.050 ± 0.028	$\mathbf{0.003 \pm 0.002}$	$\mathbf{0.007 \pm 0.004}$	0.014 ± 0.006
Finetuned on ImageNet	$\boldsymbol{\theta}^{\text{S}}_{\text{ft}}$	0.542 ± 0.190	1.486 ± 0.052	-0.072 ± 0.042	0.002 ± 0.001	0.005 ± 0.003	0.011 ± 0.005
Finetuned on ImageNet+X	$\boldsymbol{\theta}^{\text{S}}_{\text{ft}\star}$	0.501 ± 0.197	$\mathbf{1.256 \pm 0.078}$	$\underline{\mathbf{0.121 \pm 0.059}}$	$\mathbf{0.011 \pm 0.007}$	$\mathbf{0.030 \pm 0.015}$	$\mathbf{0.053 \pm 0.024}$
Finetuned on target	$\boldsymbol{\theta}^{\text{T}}_{\text{ft}}$	$\underline{\mathbf{0.940 \pm 0.065}}$	1.493 ± 0.058	-0.069 ± 0.038	0.001 ± 0.001	0.005 ± 0.003	0.010 ± 0.005
Proposed	$\tilde{\boldsymbol{\theta}}^{\text{T}}_{\text{ft}\star}$	0.928 ± 0.072	$\underline{\mathbf{1.253 \pm 0.106}}$	$\mathbf{0.117 \pm 0.072}$	$\underline{\mathbf{0.015 \pm 0.017}}$	$\underline{\mathbf{0.042 \pm 0.050}}$	$\underline{\mathbf{0.067 \pm 0.063}}$
Finetuned on target+X	$\boldsymbol{\theta}^{\text{T}}_{\text{ft}\star}$	0.943 ± 0.069	1.236 ± 0.086	0.123 ± 0.049	0.016 ± 0.012	0.045 ± 0.032	0.075 ± 0.048



(a) Input (b) Ground-truth (c) Zero-shot (d) Finetuned on ImageNet (e) Finetuned on ImageNet+X (f) Finetuned on target (g) Proposed (h) Finetuned on target+X

Figure 8: Visualization examples highlighting the top-20 patches of each attribution produced using ViT-L/14 self-explaining image classifiers. Rows show (from top to bottom) the results for Cars, GTSRB, and RESISC45 datasets.

E.3 Qualitative Evaluation

To evaluate the predicted patch-level attributions qualitatively, we visualize the top-20 patches in the attributions produced by the proposed method and the other methods in Figure 8. The visualization results are similar to those of ViT-B/32 shown in Figure 3. In particular, the patches highlighted by the proposed method (depicted in Figure 8(g)) are more similar to the GT ones (depicted in Figure 8(b)) than those highlighted by the three baseline methods that do not use any explanation supervision (depicted in Figures 8(c), (d), and (f)). Also, the model finetuned on ImageNet+X (Figure 8(e)) also generates high-quality explanations, similar to the proposed method, as in the quantitative evaluation. However, due to its low classification performance shown in Table 10, it is not practical to use this single model for both classification and explanation simultaneously. Although the model finetuned on the target dataset with explanation supervision (Figure 8(h)) produces the explanations most similar to the GT ones overall, there are cases where the proposed method produces better explanations, such as in the GTSRB example shown in the second row of Figure 8.

F Transferring Explainability Across Source and Target Datasets

In this preliminary experiment, we use eight labeled image datasets, namely Cars [36], DTD [14], EuroSAT [25], GTSRB [55], RESISC45 [12], SUN397 [63], SVHN [45], and MNIST [38]. We extend each of these datasets to include GT patch-level attributions generated using Kernel SHAP [41, 53], which explain the classification results of the CLIP image classifier finetuned on each dataset. Here, the number of perturbations for each image in Kernel SHAP is set to 500. Similar methods for

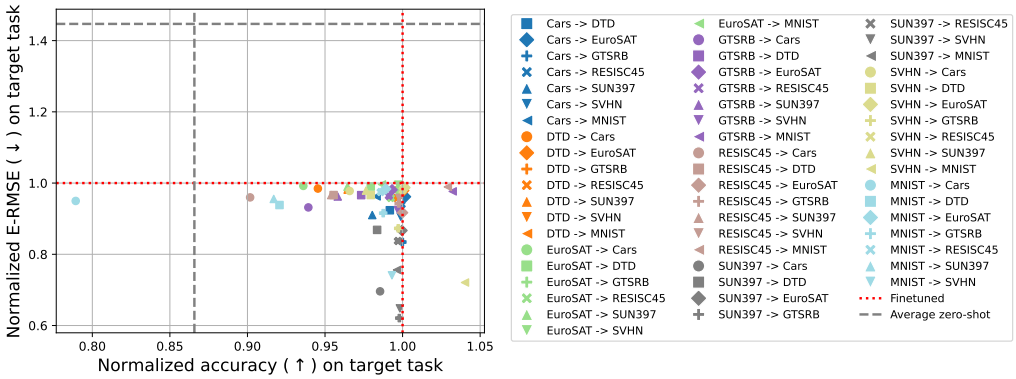


Figure 9: Normalized accuracies (larger is better) and normalized E-RMSEs (lower is better) in all the pairs of the eight benchmark datasets when fixed in $\alpha = 0.8$, $\lambda_1 = 1.0$, and $\lambda_2 = 0.7$. The color and shape of each marker indicate the source and target domains of the corresponding dataset pair, respectively.

generating explanation supervision have been used in previous studies [64, 43]. We create 56 dataset pairs from the eight datasets, where one dataset in each pair is used as the source dataset and the other as the target dataset.

In the proposed method, we first learn the three types of model parameters $\boldsymbol{\theta}^{\mathrm{S}}_{\mathrm{ft}}$, $\boldsymbol{\theta}^{\mathrm{S}}_{\mathrm{ft}\star}$, and $\boldsymbol{\theta}^{\mathrm{T}}_{\mathrm{ft}}$ for each dataset pair, and then obtain the model parameters $\tilde{\boldsymbol{\theta}}^{\mathrm{T}}_{\mathrm{ft}\star}$ by applying the proposed explainability transfer (8). We evaluate the quality of $\tilde{\boldsymbol{\theta}}^{\mathrm{T}}_{\mathrm{ft}\star}$ in terms of the classification accuracy and the explanation error using explanation root mean squared error (E-RMSE), $\sqrt{\mathcal{L}_{\mathrm{exp}}(\boldsymbol{\theta}; \mathcal{D}^{\mathrm{T}})}$, where $\mathcal{L}_{\mathrm{exp}}$ is defined in (5) and $\mathcal{D}^{\mathrm{T}}$ is the target dataset with explanation supervision.

F.1 Accuracy and Explanation Error Trade-off

The first result, illustrated in Figure 9, shows the normalized accuracies and the normalized E-RMSEs for all the dataset pairs. Here, the normalized scores are obtained by dividing by the corresponding scores of $\boldsymbol{\theta}^{\mathrm{T}}_{\mathrm{ft}}$. The red dotted line indicates the performances of $\boldsymbol{\theta}^{\mathrm{T}}_{\mathrm{ft}}$ (both the accuracy and E-RMSE are one), while the gray dashed line indicates those of the zero-shot classifier parameterized by $\boldsymbol{\theta}_{\mathrm{base}}$. In successful cases, the E-RMSE decreases from one while the accuracy stays close to one. The E-RMSE decreases on all 56 pairs, and 55 of them keep at least 90% of the accuracy. In particular, the effectiveness when using SUN397 as the source dataset is notable. This is likely because SUN397 is a large-scale dataset containing diverse scene images, allowing the model to learn generalizable explanatory capabilities for various datasets. The only pair below 90% is the transfer from handwritten digit recognition (MNIST) to vehicle type recognition (Cars), which keeps 79% of the accuracy. When transferring between the two digit recognition tasks (MNIST and SVHN), the accuracy is maintained or improved while the E-RMSE improves substantially. This suggests that transfer between very different domains can have adverse effects on accuracy.

F.2 Predictability Analysis on 56 Pairs

The 56 source–target pairs above also provide a held-out test of the predictability finding of Section 5.2, independent of the $N = 10$ ImageNet-source leave-one-target-out (LOTO) evaluation in the main paper. We refit the same six-feature regression (three classification-circuit features and three $\boldsymbol{\tau}$-output-shift features, defined in Appendix C) with the same definition of Δ_q (here with E-MSE as the error metric). We use the same nested-LOTO protocol across the eight target datasets. Each outer fold withholds one target together with its seven source pairings and trains on the remaining 49 pairs, which gives eight outer folds in total, with seven inner folds for regularization-strength selection. A simple variance decomposition explains why 6 features alone are insufficient on this larger pair set. Source identity already accounts for 55.2% of the variance of $\Delta_{\text{E-MSE}}$ and 53.1% of $\Delta_{\text{IoU@10}}$. The per-source mean $\Delta_{\text{E-MSE}}$ ranges from 0.022 (EuroSAT+X) to 0.391 (SUN397+X). No source-universal coefficient vector can fit this $18.1\times$ spread. We therefore augment the design matrix with seven source dummy variables, which are one-hot indicators of the eight source datasets with Cars+X

Table 11: 56-pair held-out test of transfer-success predictability (ViT-B/32, Δ_q target, nested LOTO over the eight target datasets). Each cell reports held-out Pearson r. "No source dummies" uses the 6 features only. "With source dummies" adds the seven source dummy variables as fixed effects.

	no source dummies (6 features)		**with source dummies (6+7)**	
Metric	Lasso	Ridge	Lasso	Ridge
E-MSE	-0.08	$+0.04$	$+0.59$	$+0.60$
IoU@1	-0.04	$+0.09$	$+0.65$	$+0.66$
IoU@5	-0.46	-0.07	$+0.55$	$+0.51$
IoU@10	-0.41	-0.00	$+0.60$	$+0.58$
Rank corr.	$+0.05$	$+0.07$	$+0.60$	$+0.61$

as the reference. They act as source fixed effects and give each source its own intercept. Table 11 reports both conditions. Without source dummies, the six-feature regression collapses on all five explanation-quality metrics (Pearson $|r| \leq 0.5$, and $r \leq -0.41$ for Lasso on IoU@5 and IoU@10). With source dummies, five of five explanation-quality metrics achieve held-out $r \in [+0.51, +0.66]$ for both Lasso and Ridge, and the resulting regression reduces held-out MSE by $1.3\times$–$1.8\times$ relative to a predictor that outputs the mean of Δ_q. On this larger pair set, source-mean variation thus needs to enter the model as an intercept rather than through the six features themselves.

The model with source dummies evaluated above absorbs per-source intercepts that are observed during training. Extrapolating to a genuinely unseen source dataset would require either (a) a leave-one-source-out protocol with explicit source-property features, or (b) committing to the 6-feature-only column of Table 11 and accepting that source-mean variation is then unmodelled. We treat this as a limitation of the present regression setup rather than as a conclusion about the underlying signal in the six features.

Bootstrap intervals and leave-one-source-out evaluation To make the scope of the predictability finding explicit, we rerun the predictor pipeline behind Table 11 with only the splits changed. First, for the $N = 10$ leave-one-target-out setting of Section 5.2, Table 12 adds bootstrap 95% intervals (10^4 resamples of the held-out (actual, predicted) pairs, seed 42) to the held-out correlations. The intervals exclude zero for E-MSE, Rank corr., IoU@10, and IoU@5 with Lasso, and include zero for IoU@1, matching the significance pattern reported there. Second, we evaluate leave-one-source-out (LOSO) on the 56 pairs. It uses eight outer folds, each withholding the seven pairs that share one source dataset, with the regularization strength selected by inner LOSO on the remaining 49 pairs. Source dummies cannot be used here, since the held-out source has no dummy, so LOSO is the analogue of the no-source-dummies column of Table 11. Because the seven predictions of a fold share one fitted model, we report a source-level cluster bootstrap interval (resampling whole held-out sources) next to the pair-level interval. The cluster interval is the appropriate one for this design and is wider. Table 13 shows the result for the six base features. Lasso collapses to a constant predictor on all five metrics, and its large negative pooled r is a leave-one-out mean-shift artifact rather than an anti-predictive signal. Ridge is flat, with every cluster interval including zero, and no cell beats the mean squared error of a constant predictor. Adding meta-features does not change this. We try the task-vector cosine $\cos(\boldsymbol{\tau}_{\mathrm{ft}}^{\mathrm{S}}, \boldsymbol{\tau}_{\mathrm{ft}}^{\mathrm{T}})$, a class-text similarity between the template-averaged CLIP text embeddings of the source and target class names, and, as an upper-bound probe only, five oracle features that depend on the target's own $\boldsymbol{\tau}_{\star}^{\mathrm{T}}$ and are unavailable before transfer. Across 4 feature sets $\times$ 5 metrics $\times$ 2 regressors, none of the 40 configurations is simultaneously non-degenerate, significant at $p < 0.05$, positive with both bootstrap intervals excluding zero, and better than the constant predictor. The predictability claim of this paper is therefore restricted to a fixed source with held-out targets. Whether held-out-source prediction is learnable with other features remains open.

G Theoretical Interpretation

In this section, we provide a theoretical interpretation of the explainability vectors $\boldsymbol{\tau}_{\star}^{\mathrm{S}}$ and $\boldsymbol{\tau}_{\star}^{\mathrm{T}}$.

$\boldsymbol{\tau}_{\star}$ as Optimal Perturbation First, we show that the explainability vector $\boldsymbol{\tau}_{\star}^{\mathrm{S}}$ can be interpreted as the optimal perturbation under a local quadratic approximation around $\boldsymbol{\theta}_{\mathrm{ft}}^{\mathrm{S}}$ of the loss function in (4). Note that the same derivation can be applied to $\boldsymbol{\tau}_{\star}^{\mathrm{T}}$. The loss function in (4) consists of the classification loss $\mathcal{L}_{\mathrm{cls}}$ and the explanation loss $\mathcal{L}_{\mathrm{exp}}$. We perform a local expansion around $\boldsymbol{\theta}_{\mathrm{ft}}^{\mathrm{S}}$ for

Table 12: Held-out Pearson correlation of the transfer-success predictor on ViT-B/32 (LOTO, $N = 10$, Δ_q target) with bootstrap 95% intervals (10^4 resamples of the held-out predictions). Point estimates match Table 8.

	Lasso		Ridge	
Metric	r	95% CI	r	95% CI
E-MSE	$+0.92$	$[+0.84, +0.98]$	$+0.91$	$[+0.82, +0.98]$
IoU@1	$+0.19$	$[-0.43, +0.55]$	$+0.33$	$[-0.02, +0.85]$
IoU@5	$+0.70$	$[+0.48, +0.94]$	$+0.63$	$[+0.33, +0.91]$
IoU@10	$+0.81$	$[+0.66, +0.94]$	$+0.85$	$[+0.71, +0.95]$
Rank corr.	$+0.77$	$[+0.56, +0.92]$	$+0.82$	$[+0.69, +0.94]$

Table 13: Leave-one-source-out evaluation on the 56 pairs (ViT-B/32, Δ_q target, six base features, no source dummies). Pooled r over the 56 held-out predictions with pair-level and source-level cluster bootstrap 95% intervals. "Degenerate" marks folds in which Lasso selects no feature and predicts a constant, whose negative pooled r is a leave-one-out mean-shift artifact. No cell beats the mean squared error of a constant predictor.

Metric	Regressor	Pooled r	Pair 95% CI	Cluster 95% CI	Note
E-MSE	Lasso	-0.74	$[-0.88, -0.54]$	$[-0.85, -0.28]$	degenerate
	Ridge	-0.21	$[-0.41, -0.01]$	$[-0.53, +0.15]$	worse than constant
IoU@1	Lasso	-0.76	$[-0.89, -0.55]$	$[-0.82, -0.22]$	degenerate
	Ridge	-0.16	$[-0.34, +0.03]$	$[-0.47, +0.21]$	worse than constant
IoU@5	Lasso	-0.14	$[-0.71, -0.01]$	$[-0.76, -0.02]$	degenerate
	Ridge	-0.12	$[-0.32, +0.04]$	$[-0.35, +0.20]$	worse than constant
IoU@10	Lasso	-0.72	$[-0.87, -0.50]$	$[-0.80, -0.32]$	degenerate
	Ridge	-0.14	$[-0.33, +0.03]$	$[-0.45, +0.14]$	worse than constant
Rank corr.	Lasso	-0.75	$[-0.88, -0.55]$	$[-0.88, -0.31]$	degenerate
	Ridge	-0.26	$[-0.46, -0.05]$	$[-0.59, +0.10]$	worse than constant

$\mathcal{L}_{\text{cls}}$. Let $\boldsymbol{\theta} = \boldsymbol{\theta}_{\text{ft}}^{\text{S}} + \boldsymbol{\tau}$ where $\boldsymbol{\tau} \in \mathbb{R}^D$ is a small perturbation. Here, $\bar{\mathcal{D}}^{\text{S}} = \{(\boldsymbol{x}^{(n)}, y^{(n)})\}_{n=1}^{N^{\text{S}}}$ denotes the classification-only subset of $\mathcal{D}^{\text{S}}$, i.e., $\mathcal{D}^{\text{S}}$ with the GT patch-level attributions $\boldsymbol{\phi}^{(n)}$ removed. This allows the same derivation to apply to the target domain, where explanation supervision is unavailable, by replacing $\bar{\mathcal{D}}^{\text{S}}$ with $\bar{\mathcal{D}}^{\text{T}}$. Since $\boldsymbol{\theta}_{\text{ft}}^{\text{S}}$ approximately satisfies $\nabla\mathcal{L}_{\text{cls}}(\boldsymbol{\theta}_{\text{ft}}^{\text{S}}; \bar{\mathcal{D}}^{\text{S}}) = \mathbf{0}$, a second-order Taylor expansion yields

$$\mathcal{L}_{\text{cls}}(\boldsymbol{\theta}_{\text{ft}}^{\text{S}} + \boldsymbol{\tau}; \bar{\mathcal{D}}^{\text{S}}) \approx \mathcal{L}_{\text{cls}}(\boldsymbol{\theta}_{\text{ft}}^{\text{S}}; \bar{\mathcal{D}}^{\text{S}}) + \frac{1}{2}\boldsymbol{\tau}^{\top}\boldsymbol{H}^{\text{S}}\boldsymbol{\tau}, \tag{13}$$

where $\boldsymbol{H}^{\text{S}} = \nabla^2\mathcal{L}_{\text{cls}}(\boldsymbol{\theta}_{\text{ft}}^{\text{S}}; \bar{\mathcal{D}}^{\text{S}})$ is the Hessian matrix of $\mathcal{L}_{\text{cls}}$ at $\boldsymbol{\theta}_{\text{ft}}^{\text{S}}$. Similarly, we perform a first-order Taylor expansion of $\mathcal{L}_{\text{exp}}$ around $\boldsymbol{\theta}_{\text{ft}}^{\text{S}}$ as follows:

$$\mathcal{L}_{\text{exp}}(\boldsymbol{\theta}_{\text{ft}}^{\text{S}} + \boldsymbol{\tau}; \mathcal{D}^{\text{S}}) \approx \mathcal{L}_{\text{exp}}(\boldsymbol{\theta}_{\text{ft}}^{\text{S}}; \mathcal{D}^{\text{S}}) + (\boldsymbol{g}^{\text{S}})^{\top}\boldsymbol{\tau}, \tag{14}$$

where $\boldsymbol{g}^{\text{S}} = \nabla\mathcal{L}_{\text{exp}}(\boldsymbol{\theta}_{\text{ft}}^{\text{S}}; \mathcal{D}^{\text{S}})$ is the gradient vector of $\mathcal{L}_{\text{exp}}$ at $\boldsymbol{\theta}_{\text{ft}}^{\text{S}}$. Substituting (13) and (14) into (4), and solving $\boldsymbol{\tau}$ that minimizes the approximated loss function under the assumptions $\alpha > 0$ and a positive definite $\boldsymbol{H}^{\text{S}}$, which make the minimizer unique, we obtain:

$$\boldsymbol{\tau}_{\star}^{\text{S}} \approx -\frac{1-\alpha}{\alpha}(\boldsymbol{H}^{\text{S}})^{-1}\boldsymbol{g}^{\text{S}}. \tag{15}$$

The negative gradient $-\boldsymbol{g}^{\text{S}}$ represents the direction in which the explanation loss $\mathcal{L}_{\text{exp}}$ decreases most rapidly around $\boldsymbol{\theta}_{\text{ft}}^{\text{S}}$. However, simply moving $\boldsymbol{\theta}_{\text{ft}}^{\text{S}}$ in that direction could significantly degrade classification performance. The Hessian matrix $\boldsymbol{H}^{\text{S}}$ represents the curvature of the classification loss $\mathcal{L}_{\text{cls}}$, and preconditioning $\boldsymbol{g}^{\text{S}}$ with its inverse $(\boldsymbol{H}^{\text{S}})^{-1}$ damps movement along parameter directions where the classification loss has high curvature. The hyperparameter α controls the magnitude of $\boldsymbol{\tau}_{\star}^{\text{S}}$. As α approaches one, the factor $(1-\alpha)/\alpha$ and hence $\boldsymbol{\tau}_{\star}^{\text{S}}$ shrink. The transferred parameters (8) then stay close to $\boldsymbol{\theta}_{\text{base}} + \lambda_1\boldsymbol{\tau}_{\text{ft}}^{\text{T}}$, so classification accuracy is prioritized.

Risk of Classification Accuracy Drop Because $\boldsymbol{\theta}_{\text{ft}}^{\text{T}}$ is a local minimum of the target classification loss $\mathcal{L}_{\text{cls}}$, no parameter shift, including the addition of the source-domain explainability vector $\boldsymbol{\tau}_{\star}^{\text{S}}$, decreases the classification loss under the local quadratic approximation. The increase comes from

the components of the shift along high-curvature directions of the target loss. Formally, based on the approximated classification loss (13), the increase in classification loss when adding the source-domain explainability vector $\boldsymbol{\tau}_\star^{\mathrm{S}}$ to the target-domain parameters $\boldsymbol{\theta}_{\mathrm{ft}}^{\mathrm{T}}$ is proportional to

$$R(\boldsymbol{\tau}_\star^{\mathrm{S}}; \boldsymbol{H}^{\mathrm{T}}) = (\boldsymbol{\tau}_\star^{\mathrm{S}})^\top \boldsymbol{H}^{\mathrm{T}} \boldsymbol{\tau}_\star^{\mathrm{S}}, \tag{16}$$

which equals twice the second-order term $\frac{1}{2}(\boldsymbol{\tau}_\star^{\mathrm{S}})^\top \boldsymbol{H}^{\mathrm{T}} \boldsymbol{\tau}_\star^{\mathrm{S}}$ in the local quadratic approximation around $\boldsymbol{\theta}_{\mathrm{ft}}^{\mathrm{T}}$. With $\lambda_1 = 1$ in (8), the transferred parameters are $\boldsymbol{\theta}_{\mathrm{ft}}^{\mathrm{T}} + \lambda_2 \boldsymbol{\tau}_\star^{\mathrm{S}}$. The increase is then $\frac{\lambda_2^2}{2} R(\boldsymbol{\tau}_\star^{\mathrm{S}}; \boldsymbol{H}^{\mathrm{T}})$, so λ_2 scales this risk quadratically. To understand the property of this function, we perform eigenvalue decomposition of $\boldsymbol{H}^{\mathrm{T}}$ as

$$\begin{aligned} \boldsymbol{H}^{\mathrm{T}} &= \boldsymbol{U} \operatorname{diag}(\nu_1, \nu_2, \ldots, \nu_D) \boldsymbol{U}^\top, \\ &\quad (\nu_1 \geq \nu_2 \geq \cdots \geq \nu_D \geq 0). \end{aligned} \tag{17}$$

Here, $\boldsymbol{U} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_D]$ is the matrix of orthonormal eigenvectors of $\boldsymbol{H}^{\mathrm{T}}$, and $\nu_1, \ldots, \nu_D$ are the corresponding eigenvalues. Expanding $\boldsymbol{\tau}_\star^{\mathrm{S}}$ in the eigenvector basis of $\boldsymbol{H}^{\mathrm{T}}$ yields

$$\boldsymbol{\tau}_\star^{\mathrm{S}} = \sum_{i=1}^{D} a_i \boldsymbol{u}_i, \quad a_i = (\boldsymbol{\tau}_\star^{\mathrm{S}})^\top \boldsymbol{u}_i. \tag{18}$$

Then (16) can be rewritten as

$$R(\boldsymbol{\tau}_\star^{\mathrm{S}}; \boldsymbol{H}^{\mathrm{T}}) = \sum_{i=1}^{D} \nu_i a_i^2. \tag{19}$$

This expression indicates that if $\boldsymbol{\tau}_\star^{\mathrm{S}}$ has a large component in the direction of the eigenvectors associated with large eigenvalues of $\boldsymbol{H}^{\mathrm{T}}$, the increase in classification loss $R(\boldsymbol{\tau}_\star^{\mathrm{S}}; \boldsymbol{H}^{\mathrm{T}})$ will be significant.

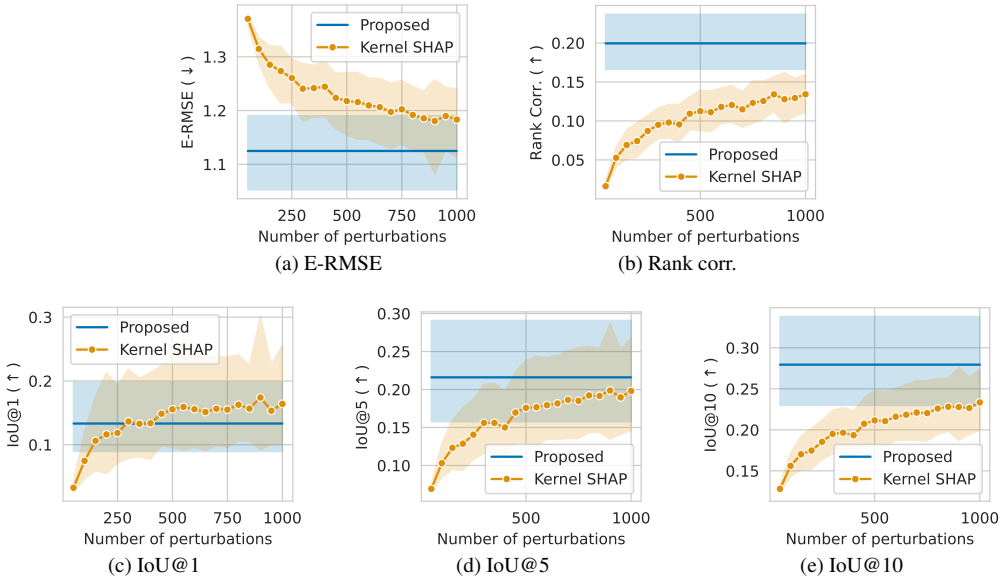
If a precomputed Hessian matrix $\boldsymbol{H}^{\mathrm{T}}$ is available, we can select the explainability vector $\boldsymbol{\tau}_\star^{\mathrm{S}}$ that minimizes $R(\boldsymbol{\tau}_\star^{\mathrm{S}}; \boldsymbol{H}^{\mathrm{T}})$ among candidates learned in different source domains, without requiring any target-domain explanation supervision (the Hessian itself is precomputed once on $\bar{\mathcal{D}}^{\mathrm{T}}$). However, storing the precomputed $\boldsymbol{H}^{\mathrm{T}}$, a dense matrix whose size scales quadratically with the number of model parameters, is infeasible. In practice, it is therefore reasonable to approximate it with a diagonal matrix.

H Results on Comparison with Kernel SHAP

Figure 10 shows the explainability scores of the proposed method and Kernel SHAP with varying number of perturbations P when transferring explainability learned on ImageNet+X to the 10 target datasets. Kernel SHAP is applied to $\boldsymbol{\theta}_{\mathrm{ft}}^{\mathrm{T}}$ with perturbation samples independent of those used for the GT, which is itself SHAP-500 (Section 5.1.4). For Rank corr., the proposed method remains clearly higher than Kernel SHAP even at 1,000 perturbations (0.200 vs. 0.134). For IoU@5 and IoU@10, the proposed method is higher at every number of perturbations, but the gap narrows to 0.02 and 0.05 at 1,000 perturbations and the 95% confidence intervals overlap for large numbers of perturbations. For E-RMSE, the proposed method is lower than Kernel SHAP at every number of perturbations, but the gap narrows to 0.06 at 1,000 perturbations and the 95% confidence intervals overlap for large numbers of perturbations. For IoU@1, the proposed method matches Kernel SHAP with about 300 perturbations, and Kernel SHAP is higher with more perturbations.

I Discussion on Computational Complexity of Explanation Generation

As shown in Figure 1, the difference between our self-explaining model, i.e., "prediction-with-explanation" model, and the "prediction-only" model used when generating patch-level attributions with Kernel SHAP is that the former applies the domain-specific head $g_{\boldsymbol{W}}$ to both the [CLS] token and the M image patches, while the latter applies it only to the [CLS] token. The domain-specific head $g_{\boldsymbol{W}}$ calculates matrix multiplication $\boldsymbol{W}^\top \boldsymbol{u}$ for input feature vector $\boldsymbol{u} \in \mathbb{R}^K$ and weight matrix $\boldsymbol{W} \in \mathbb{R}^{K \times C}$, where C is the number of classes. The computational complexity of this matrix



(a) E-RMSE (b) Rank corr.

(c) IoU@1 (d) IoU@5 (e) IoU@10

Figure 10: Explainability scores of the proposed method and Kernel SHAP averaged across the target datasets with varying number of perturbations. The shaded area represents the 95% confidence interval.

multiplication is $\mathcal{O}(KC)$. Denoting the computational complexity of the feature extractor $f_{\boldsymbol{\theta}}$ as F, the computational complexity of our self-explaining image classifier per explanation is $\mathcal{O}(F+MKC)$ where M is the number of patches in an image. In practice, since MKC is typically much smaller than F, the computational complexity can be approximated as $\mathcal{O}(F)$.

In contrast, the computational complexity of Kernel SHAP per explanation is $\mathcal{O}\left(PF + PM^2\right)$, where P is the number of perturbations and PM^2 is associated with solving the least-squares problem. In practice, since PF is much larger than PM^2, the computational complexity can be approximated as $\mathcal{O}(PF)$.

In summary, the per-explanation complexities of our self-explaining image classifier and Kernel SHAP both scale linearly with the feature-extractor cost F, but Kernel SHAP multiplies it by P.

Costs and amortization Table 14 attaches measured and estimated costs to this analysis (CLIP ViT-B/32, one NVIDIA RTX 6000 Ada GPU unless noted). The source side is a one-time cost. Generating the Kernel SHAP supervision for the 1.28M training images of ImageNet-1k at $P = 500$ takes 91.4 single-GPU hours, extrapolated from the measured 0.257 s per image at $P = 500$. The original run took about 29.2 hours of wall-clock time, reconstructed from file timestamps, on an unrecorded number of GPUs. Training $\boldsymbol{\theta}^{\mathrm{S}}_{\mathrm{ft}}$ and $\boldsymbol{\theta}^{\mathrm{S}}_{\mathrm{ft}\star}$ takes 9.0 GPU-hours, estimated as twice the 4.5 GPU-hours measured for one training run with the identical protocol. The source side thus costs 100.4 GPU-hours in total. As with other pretrained transferable explainers, this cost is borne once by the provider of the explainability vector. Transferring the vector to a target is a parameter addition that takes under a second on a CPU, and the target side pays no training cost. At inference, our single joint forward pass produces the prediction and the explanation in about 0.39 ms per image (a proxy measured on the single forward pass of $\boldsymbol{\theta}^{\mathrm{T}}_{\mathrm{ft}}$, which has the same architecture), against 257 ms for Kernel SHAP with $P = 500$ on $\boldsymbol{\theta}^{\mathrm{T}}_{\mathrm{ft}}$ (a ratio of about 664×, larger than the asymptotic $P = 500$, whose causes we do not separate), 0.8 ms for the amortized explainer of Appendix L (two forwards), and 7.2 ms for TVE (three forwards). These are per-image averages under each evaluation's own batch setting (batched forwards for the proposed method and the amortized explainer, one image at a time for Kernel SHAP), not single-image latencies, so the ratio depends on these conditions. The one-time cost breaks even against running Kernel SHAP per explanation after 1.41M served explanations in total. For $T = 10$ targets this is about 141k explanations per target, or 1.6 days if each target serves one image per second, after which every further explanation is essentially free relative to a prediction. At the scale of our evaluation alone (10 targets × 500 images for the faithfulness metrics) the source cost is not recouped. The argument for amortization is a deployment-scale one.

Table 14: Costs for CLIP ViT-B/32 (NVIDIA RTX 6000 Ada). GPU-hours for the source side are extrapolated or estimated single-GPU equivalents (see text). Per-image times are for prediction plus explanation, averaged per image under each evaluation's batch setting (not single-image latencies), and the time of the proposed method is a proxy.

Stage	Item	Cost
Source (one-time)	Kernel SHAP supervision for the ImageNet-1k training set (1.28M images, $P = 500$)	91.4 GPU-h
	Training $\boldsymbol{\theta}_{\mathrm{ft}}^{\mathrm{S}}$ and $\boldsymbol{\theta}_{\mathrm{ft}\star}^{\mathrm{S}}$	9.0 GPU-h
	Total	100.4 GPU-h
Per target	Transfer by parameter addition (8) (CPU)	0.8 s
Per image	Proposed (one forward, prediction and explanation)	0.39 ms
	$\boldsymbol{\theta}_{\mathrm{ft}}^{\mathrm{T}}$ + amortized explainer (two forwards)	0.8 ms
	TVE (three forwards)	7.2 ms
	$\boldsymbol{\theta}_{\mathrm{ft}}^{\mathrm{T}}$ + Kernel SHAP, $P = 500$ (≈ 500 forwards)	257 ms
Break-even vs. Kernel SHAP	Served explanations in total ($T = 10$: per target)	1.41M (141k)

J Faithfulness of the Transferred Self-Explanations

This appendix gives the procedure and the per-target results of the within-model faithfulness evaluation summarized in Section 5.1.5. GT agreement (Table 1) measures how well the transferred model reproduces the explanation target chosen by the user. Faithfulness asks a different question, namely whether the patches highlighted by the transferred model matter for that model's own predictions. Since faithfulness is a property of an (explanation, model) pair, every attribution below is scored on the predicted probabilities of the same transferred model $\tilde{\boldsymbol{\theta}}_{\mathrm{ft}\star}^{\mathrm{T}}$.

Target set Appendices J, K, M, and B.3, and Appendix L except Table 20, use a 10-target set in which CIFAR10 replaces SUN397 of Table 1. The nine shared targets use the same checkpoints and the same evaluation pipeline as Table 1. On this set, the mean accuracy of $\boldsymbol{\theta}_{\mathrm{ft}}^{\mathrm{T}}$ is 0.914 and that of the transferred model is 0.911, which is why 10-target means in these appendices differ from those of Table 1. We refer to it as the CIFAR10 set where the distinction matters.

Procedure For each target we draw 500 test images by class-proportional stratified sampling (seed 42, at least one image per class) and keep the images that $\tilde{\boldsymbol{\theta}}_{\mathrm{ft}\star}^{\mathrm{T}}$ classifies correctly ($N = 338$–492 per target, Table 15), so that the attribution for the true class y is also that for the predicted class. Four attributions are compared: (i) the self-explanation, i.e., the attribution $\tilde{\boldsymbol{\Phi}}_c$ for the predicted class c, obtained in the same forward pass as the prediction; (ii) Kernel SHAP on $\tilde{\boldsymbol{\theta}}_{\mathrm{ft}\star}^{\mathrm{T}}$ itself with $P = 500$ perturbations, following the GT-generation protocol of Section 5.1 (the model-specific reference); (iii) the GT attribution of Section 5.1, i.e., Kernel SHAP applied to the separate model $\boldsymbol{\theta}_{\mathrm{ft}}^{\mathrm{T}}$; and (iv) a random attribution (standard normal entries, seed 42). Each attribution is scored with four metrics computed on the predicted probability of $\tilde{\boldsymbol{\theta}}_{\mathrm{ft}\star}^{\mathrm{T}}$ for the predicted class. The first two are Fidelity+ and Fidelity− adapted from the fidelity metrics of TVE [57], i.e., the probability drop when the top-$k\%$ patches are masked (higher is better) and when the bottom-$k\%$ patches are masked (lower is better), averaged over $k \in \{10, 20, 30, 40, 50\}$. The other two are deletion and insertion scores in the style of Petsiuk et al. [47], i.e., the probability averaged over the steps at which the top $1, 11, 21, \ldots$ patches by attribution (every 10 patches below the number of patches) are removed (lower is better) or are the only patches kept (higher is better). For CLIP ViT-B/32 with 49 patches these are five steps $(1, 11, 21, 31, 41)$, and for the 196-patch grid of TVE in Appendix L they are 20 steps $(1, 11, \ldots, 191)$. Masked patches are set to zero in the normalized input, and unlike the original AUC the two endpoints of the curve are not included.

Results Table 15 lists the raw values. The self-explanation matches or exceeds the reference on 36 of the 40 target–metric comparisons and beats random on all 40. Comparisons use unrounded values, so the two cells that tie at three decimals in Table 15 count as a match (Del on Cars) and a miss (Ins on EuroSAT). The random attribution is the worst row on every metric for every target, and the GT attribution of the separate model $\boldsymbol{\theta}_{\mathrm{ft}}^{\mathrm{T}}$ also beats random everywhere. Kernel SHAP with $P = 500$ is a finite-sample Monte Carlo estimate rather than a faithfulness ceiling, so a single-forward-pass explanation can exceed it. The result thus indicates that, on 36 of the 40 comparisons, the transferred

Table 15: Raw faithfulness scores on the transferred model $\tilde{\boldsymbol{\theta}}^{\mathrm{T}}_{\mathrm{ft}\star}$ (CIFAR10 set, class-stratified 500-image subsets restricted to correctly classified images, with N in parentheses). All four attributions are scored on the predicted probabilities of $\tilde{\boldsymbol{\theta}}^{\mathrm{T}}_{\mathrm{ft}\star}$. Best value per target and metric in bold.

Dataset (N)	Attribution	Fid+ (↑)	Fid− (↓)	Ins (↑)	Del (↓)
Cars (338)	Self-explanation (ours)	**0.820**	**0.094**	**0.493**	0.148
	Kernel SHAP on $\tilde{\boldsymbol{\theta}}^{\mathrm{T}}_{\mathrm{ft}\star}$	0.789	0.255	0.353	0.148
	GT (Kernel SHAP on $\boldsymbol{\theta}^{\mathrm{T}}_{\mathrm{ft}}$)	0.799	0.173	0.422	**0.147**
	Random	0.508	0.515	0.175	0.311
CIFAR10 (492)	Self-explanation (ours)	**0.409**	**0.029**	**0.737**	**0.492**
	Kernel SHAP on $\tilde{\boldsymbol{\theta}}^{\mathrm{T}}_{\mathrm{ft}\star}$	0.316	0.046	0.677	0.553
	GT (Kernel SHAP on $\boldsymbol{\theta}^{\mathrm{T}}_{\mathrm{ft}}$)	0.280	0.049	0.677	0.585
	Random	0.147	0.140	0.555	0.682
CIFAR100 (445)	Self-explanation (ours)	**0.570**	**0.110**	**0.575**	**0.351**
	Kernel SHAP on $\tilde{\boldsymbol{\theta}}^{\mathrm{T}}_{\mathrm{ft}\star}$	0.531	0.115	0.551	0.381
	GT (Kernel SHAP on $\boldsymbol{\theta}^{\mathrm{T}}_{\mathrm{ft}}$)	0.450	0.118	0.536	0.426
	Random	0.262	0.255	0.399	0.538
DTD (395)	Self-explanation (ours)	0.343	**0.147**	**0.532**	**0.482**
	Kernel SHAP on $\tilde{\boldsymbol{\theta}}^{\mathrm{T}}_{\mathrm{ft}\star}$	**0.351**	0.174	0.501	0.496
	GT (Kernel SHAP on $\boldsymbol{\theta}^{\mathrm{T}}_{\mathrm{ft}}$)	0.318	0.163	0.518	0.512
	Random	0.260	0.247	0.432	0.561
EuroSAT (491)	Self-explanation (ours)	**0.181**	0.067	0.672	**0.698**
	Kernel SHAP on $\tilde{\boldsymbol{\theta}}^{\mathrm{T}}_{\mathrm{ft}\star}$	0.156	0.044	0.672	0.714
	GT (Kernel SHAP on $\boldsymbol{\theta}^{\mathrm{T}}_{\mathrm{ft}}$)	0.166	**0.033**	**0.699**	0.701
	Random	0.076	0.085	0.632	0.765
GTSRB (491)	Self-explanation (ours)	**0.893**	**0.006**	**0.789**	**0.211**
	Kernel SHAP on $\tilde{\boldsymbol{\theta}}^{\mathrm{T}}_{\mathrm{ft}\star}$	0.798	0.034	0.717	0.260
	GT (Kernel SHAP on $\boldsymbol{\theta}^{\mathrm{T}}_{\mathrm{ft}}$)	0.781	0.035	0.736	0.265
	Random	0.287	0.266	0.415	0.543
MNIST (446)	Self-explanation (ours)	**0.768**	**0.057**	**0.664**	**0.280**
	Kernel SHAP on $\tilde{\boldsymbol{\theta}}^{\mathrm{T}}_{\mathrm{ft}\star}$	0.665	0.250	0.460	0.323
	GT (Kernel SHAP on $\boldsymbol{\theta}^{\mathrm{T}}_{\mathrm{ft}}$)	0.661	0.212	0.493	0.320
	Random	0.418	0.430	0.326	0.449
RESISC45 (482)	Self-explanation (ours)	**0.468**	0.077	**0.598**	**0.434**
	Kernel SHAP on $\tilde{\boldsymbol{\theta}}^{\mathrm{T}}_{\mathrm{ft}\star}$	0.384	**0.071**	0.597	0.494
	GT (Kernel SHAP on $\boldsymbol{\theta}^{\mathrm{T}}_{\mathrm{ft}}$)	0.335	0.083	0.592	0.522
	Random	0.179	0.186	0.475	0.612
STL10 (492)	Self-explanation (ours)	**0.439**	**0.027**	**0.743**	**0.473**
	Kernel SHAP on $\tilde{\boldsymbol{\theta}}^{\mathrm{T}}_{\mathrm{ft}\star}$	0.287	0.065	0.658	0.579
	GT (Kernel SHAP on $\boldsymbol{\theta}^{\mathrm{T}}_{\mathrm{ft}}$)	0.325	0.038	0.699	0.556
	Random	0.147	0.145	0.555	0.676
SVHN (481)	Self-explanation (ours)	**0.710**	**0.032**	**0.727**	**0.323**
	Kernel SHAP on $\tilde{\boldsymbol{\theta}}^{\mathrm{T}}_{\mathrm{ft}\star}$	0.545	0.040	0.657	0.399
	GT (Kernel SHAP on $\boldsymbol{\theta}^{\mathrm{T}}_{\mathrm{ft}}$)	0.582	0.039	0.683	0.380
	Random	0.190	0.185	0.481	0.598

self-explanation is at least as faithful to the model it accompanies as the reference, with one forward pass instead of $P = 500$. On the unstratified first 500 test images per target, the self-explanation matches or exceeds the reference on 31 of the 36 comparisons for the nine targets other than EuroSAT and beats random on 36 of 40. All four misses against random are on EuroSAT, whose first 500 test images contain only two of its 10 classes, which is why we use stratified sampling.

K Control-Vector Ablations

This appendix details the control experiment summarized in Table 2, which asks whether the subtraction $\boldsymbol{\tau}^{\mathrm{S}}_{\star} = \boldsymbol{\theta}^{\mathrm{S}}_{\mathrm{ft}\star} - \boldsymbol{\theta}^{\mathrm{S}}_{\mathrm{ft}}$ specifically isolates explanation ability, or whether a similar effect could be

Table 16: Control vectors: 10-target means on the CIFAR10 set (full test split). Each row adds the indicated vector to $\boldsymbol{\theta}_{\text{base}} + \boldsymbol{\tau}^{\text{T}}_{\text{ft}}$ with $\lambda_2 = 0.7$. The first row is the proposed method. Labels (a)–(f) refer to the control vectors defined above.

Added vector	Acc. (↑)	E-RMSE (↓)	Rank corr. (↑)	IoU@1 (↑)	IoU@10 (↑)
Proposed ($\boldsymbol{\tau}^{\text{S}}_{\star}$)	0.911	1.118	0.204	0.132	0.281
(e) No vector	0.914	1.563	−0.147	0.009	0.077
(a) Equal-norm random	0.909	1.560	−0.144	0.010	0.078
(b) Front-6 blocks	0.912	1.563	−0.147	0.009	0.077
(b) Back-6 blocks	0.913	1.189	0.148	0.109	0.244
(c) Full FT, raw	0.872	1.109	0.215	0.132	0.283
(f) Full FT, norm-matched	0.911	1.327	0.046	0.053	0.167
(d) Shuffled supervision	0.909	1.154	0.190	0.079	0.264

produced by a perturbation of the same magnitude, by source-specific optimization effects, or by an attribution prior that does not depend on the content of the explanation supervision. Every control vector $\boldsymbol{\tau}_{\text{ctrl}}$ enters the transfer rule (8) in place of $\boldsymbol{\tau}^{\text{S}}_{\star}$ with the same coefficients ($\lambda_1 = 1.0$, $\lambda_2 = 0.7$), and the resulting model is evaluated with the pipeline of Table 1 (full test split, GT-agreement metrics) on the CIFAR10 set (Appendix J).

Control vectors

(a) Equal-norm random: a Gaussian random direction over all parameters, rescaled so that its global ℓ_2 norm equals $\|\boldsymbol{\tau}^{\text{S}}_{\star}\| = 10.48$ (seed 42). Any effect would come from the magnitude of the perturbation alone.

(b) Layer-restricted: $\boldsymbol{\tau}^{\text{S}}_{\star}$ restricted to transformer blocks 1–6 (front-6) or 7–12 (back-6), with all other parameters set to zero and no rescaling (norms 6.58 and 8.07). This localizes where the effect resides.

(c) Full finetuning vector, raw: $\boldsymbol{\theta}^{\text{S}}_{\text{ft}\star} - \boldsymbol{\theta}_{\text{base}}$, i.e., the source vector without the subtraction of $\boldsymbol{\tau}^{\text{S}}_{\text{ft}}$ (norm 26.6, 2.5× that of $\boldsymbol{\tau}^{\text{S}}_{\star}$). If the subtraction were unnecessary, this vector should behave like $\boldsymbol{\tau}^{\text{S}}_{\star}$.

(d) Shuffled supervision: the explanation maps of ImageNet+X are permuted across images (seed 42), which destroys the correspondence between an image and its explanation while preserving the distribution and format of the maps. The source model is retrained on this supervision with the same hyperparameters and initialization as $\boldsymbol{\theta}^{\text{S}}_{\text{ft}\star}$, and the control vector is $\boldsymbol{\theta}^{\text{S}'}_{\text{ft}\star} - \boldsymbol{\theta}^{\text{S}}_{\text{ft}}$. Whatever this vector transfers is a content-free attribution prior. This control is a single retraining with one permutation seed.

(e) No vector: $\boldsymbol{\tau}_{\text{ctrl}} = \mathbf{0}$, i.e., $\boldsymbol{\theta}^{\text{T}}_{\text{ft}}$ itself, the baseline for all control vectors.

(f) Full finetuning vector, norm-matched: (c) rescaled to $\|\boldsymbol{\tau}^{\text{S}}_{\star}\|$, which separates the direction of (c) from its magnitude.

Results Table 16 reports the 10-target means on all five metrics. Table 2 in the main text shows three of them for six of the eight vectors. Table 17 lists the per-target results.

We define the recovery of a control vector on a metric as its improvement over no vector (e) divided by the improvement of $\boldsymbol{\tau}^{\text{S}}_{\star}$ over (e), both computed on unrounded 10-target means. The control vectors behave as follows.

- The equal-norm random vector (a) and the front-6 vector (b) are indistinguishable from no vector (e) on every metric. A perturbation of the same norm, or of the right direction confined to the first six blocks, contributes no explanation-specific signal. Note that $0.7\|\boldsymbol{\tau}^{\text{S}}_{\star}\| = 7.3$ is 2.4–3.0× the norm of the targets' own task vectors $\boldsymbol{\tau}^{\text{T}}_{\text{ft}}$ (measured on Cars, GTSRB, and RESISC45), so this is not a small perturbation.
- The back-6 vector (b) recovers 84.0% of the Rank corr. improvement, 81.5% of the IoU@1 improvement, and 81.9% of the IoU@10 improvement. The per-target recovery is not uniform. MNIST is the clear outlier (58% Rank corr. and 51% IoU@10), while the other nine targets lie between 75% and 95% on Rank corr. Together with the front-6 result, this is a causal localization of the transferred effect in the later blocks (Appendix B.3).

Table 17: Per-target results of the control-vector ablation (CIFAR10 set, full test split, $\lambda_1 = 1.0$, $\lambda_2 = 0.7$). Each row adds the indicated vector to $\boldsymbol{\theta}_{\text{base}} + \boldsymbol{\tau}_{\text{ft}}^{\text{T}}$. The labels (a)–(f) refer to the list in the text. 10-target means are in Table 16.

Dataset	Vector	Accuracy (↑)	E-RMSE (↓)	Rank corr. (↑)	IoU@1 (↑)	IoU@10 (↑)
Cars	Proposed ($\boldsymbol{\tau}_\star^{\text{S}}$)	0.702	1.068	0.249	0.169	0.350
	(e) No vector	0.729	1.618	−0.195	0.004	0.053
	(a) Equal-norm random	0.713	1.616	−0.194	0.004	0.054
	(b) Front-6 blocks	0.717	1.616	−0.191	0.004	0.055
	(b) Back-6 blocks	0.720	1.101	0.227	0.137	0.328
	(c) Full FT, raw	0.506	1.068	0.250	0.155	0.348
	(f) Full FT, norm-matched	0.706	1.269	0.111	0.062	0.216
	(d) Shuffled supervision	0.704	1.153	0.207	0.062	0.286
CIFAR10	Proposed ($\boldsymbol{\tau}_\star^{\text{S}}$)	0.976	1.184	0.172	0.072	0.221
	(e) No vector	0.976	1.532	−0.124	0.012	0.089
	(a) Equal-norm random	0.974	1.531	−0.123	0.014	0.089
	(b) Front-6 blocks	0.977	1.533	−0.126	0.012	0.089
	(b) Back-6 blocks	0.976	1.235	0.130	0.059	0.200
	(c) Full FT, raw	0.969	1.173	0.185	0.077	0.222
	(f) Full FT, norm-matched	0.977	1.351	0.031	0.036	0.146
	(d) Shuffled supervision	0.974	1.226	0.148	0.044	0.193
CIFAR100	Proposed ($\boldsymbol{\tau}_\star^{\text{S}}$)	0.878	1.223	0.143	0.099	0.222
	(e) No vector	0.877	1.525	−0.105	0.011	0.090
	(a) Equal-norm random	0.874	1.524	−0.104	0.011	0.090
	(b) Front-6 blocks	0.879	1.525	−0.105	0.011	0.091
	(b) Back-6 blocks	0.880	1.256	0.116	0.081	0.205
	(c) Full FT, raw	0.844	1.218	0.146	0.103	0.223
	(f) Full FT, norm-matched	0.879	1.336	0.054	0.051	0.161
	(d) Shuffled supervision	0.874	1.272	0.111	0.060	0.190
DTD	Proposed ($\boldsymbol{\tau}_\star^{\text{S}}$)	0.769	1.242	0.141	0.048	0.172
	(e) No vector	0.778	1.503	−0.103	0.011	0.085
	(a) Equal-norm random	0.770	1.499	−0.098	0.016	0.086
	(b) Front-6 blocks	0.770	1.499	−0.098	0.010	0.087
	(b) Back-6 blocks	0.773	1.289	0.102	0.039	0.157
	(c) Full FT, raw	0.741	1.232	0.151	0.054	0.177
	(f) Full FT, norm-matched	0.779	1.363	0.032	0.028	0.129
	(d) Shuffled supervision	0.771	1.230	0.157	0.034	0.177
EuroSAT	Proposed ($\boldsymbol{\tau}_\star^{\text{S}}$)	0.986	1.203	0.143	0.072	0.189
	(e) No vector	0.986	1.437	−0.066	0.015	0.103
	(a) Equal-norm random	0.985	1.436	−0.065	0.016	0.104
	(b) Front-6 blocks	0.983	1.438	−0.068	0.017	0.104
	(b) Back-6 blocks	0.986	1.262	0.090	0.053	0.169
	(c) Full FT, raw	0.969	1.197	0.148	0.071	0.191
	(f) Full FT, norm-matched	0.983	1.337	0.021	0.036	0.144
	(d) Shuffled supervision	0.982	1.212	0.138	0.050	0.181

- The raw full finetuning vector (c) explains as well as $\boldsymbol{\tau}_\star^{\text{S}}$ (its 10-target means are within 0.011 of the proposed method on every explanation metric) but costs 3.9 points of accuracy on average, concentrated on Cars (-19.6 points). Norm-matching it (f) removes the accuracy loss but collapses explanation quality (Rank corr. 0.046 vs. 0.204). The explanation effect of the raw vector therefore rides on its surplus norm, and the subtraction of $\boldsymbol{\tau}_{\text{ft}}^{\text{S}}$ extracts the explanation component at an accuracy-preserving norm.
- The shuffled-supervision vector (d) recovers most of the coarse ranking structure (96% of the Rank corr. and 92% of the IoU@10 improvement), so a considerable part of the transferred effect is a content-free attribution prior. Content specificity shows up at the top-1 patch. The vector recovers only 57.4% of the IoU@1 improvement and is below $\boldsymbol{\tau}_\star^{\text{S}}$ on IoU@1 for 9 of the 10 targets (e.g., Cars 0.062 vs. 0.169, GTSRB 0.219 vs. 0.366). It ties or exceeds $\boldsymbol{\tau}_\star^{\text{S}}$ on Rank corr. or IoU@10 on three targets (DTD, GTSRB, and SVHN).

L Comparison with Amortized and Transferable Explainers

Table 19 compares the deployment options discussed in Section 5.1.5; this appendix details the two alternatives, an amortized explainer trained with the same supervision as our source model and Transferable Vision Explainer (TVE) [57].

Table 18: Continuation of Table 17 for the remaining five targets.

Dataset	Vector	Accuracy (↑)	E-RMSE (↓)	Rank corr. (↑)	IoU@1 (↑)	IoU@10 (↑)
GTSRB	Proposed ($\boldsymbol{\tau}_\star^{\mathrm{S}}$)	0.989	0.889	0.307	0.366	0.453
	(e) No vector	0.989	1.638	−0.187	0.003	0.065
	(a) Equal-norm random	0.987	1.623	−0.175	0.003	0.070
	(b) Front-6 blocks	0.988	1.637	−0.188	0.002	0.064
	(b) Back-6 blocks	0.989	0.923	0.280	0.313	0.433
	(c) Full FT, raw	0.983	0.888	0.310	0.367	0.453
	(f) Full FT, norm-matched	0.989	1.202	0.116	0.132	0.271
	(d) Shuffled supervision	0.988	0.937	0.303	0.219	0.470
MNIST	Proposed ($\boldsymbol{\tau}_\star^{\mathrm{S}}$)	0.900	1.077	0.230	0.075	0.323
	(e) No vector	0.895	1.657	−0.258	0.005	0.045
	(a) Equal-norm random	0.887	1.656	−0.255	0.006	0.045
	(b) Front-6 blocks	0.893	1.659	−0.261	0.003	0.044
	(b) Back-6 blocks	0.898	1.322	0.025	0.066	0.187
	(c) Full FT, raw	0.826	1.062	0.264	0.074	0.318
	(f) Full FT, norm-matched	0.886	1.483	−0.110	0.032	0.076
	(d) Shuffled supervision	0.903	1.091	0.219	0.087	0.318
RESISC45	Proposed ($\boldsymbol{\tau}_\star^{\mathrm{S}}$)	0.955	1.185	0.175	0.104	0.227
	(e) No vector	0.957	1.472	−0.057	0.017	0.102
	(a) Equal-norm random	0.956	1.470	−0.055	0.017	0.103
	(b) Front-6 blocks	0.956	1.471	−0.056	0.017	0.103
	(b) Back-6 blocks	0.957	1.218	0.150	0.082	0.210
	(c) Full FT, raw	0.934	1.176	0.181	0.109	0.230
	(f) Full FT, norm-matched	0.956	1.308	0.080	0.044	0.164
	(d) Shuffled supervision	0.951	1.207	0.167	0.074	0.207
STL10	Proposed ($\boldsymbol{\tau}_\star^{\mathrm{S}}$)	0.978	1.094	0.236	0.133	0.278
	(e) No vector	0.979	1.559	−0.149	0.011	0.081
	(a) Equal-norm random	0.976	1.557	−0.146	0.011	0.081
	(b) Front-6 blocks	0.980	1.562	−0.154	0.007	0.078
	(b) Back-6 blocks	0.980	1.132	0.206	0.113	0.262
	(c) Full FT, raw	0.981	1.083	0.246	0.141	0.282
	(f) Full FT, norm-matched	0.984	1.255	0.112	0.062	0.198
	(d) Shuffled supervision	0.974	1.161	0.199	0.040	0.240
SVHN	Proposed ($\boldsymbol{\tau}_\star^{\mathrm{S}}$)	0.974	1.018	0.241	0.178	0.377
	(e) No vector	0.974	1.686	−0.222	0.003	0.055
	(a) Equal-norm random	0.973	1.685	−0.222	0.003	0.055
	(b) Front-6 blocks	0.974	1.687	−0.226	0.003	0.053
	(b) Back-6 blocks	0.974	1.158	0.149	0.148	0.289
	(c) Full FT, raw	0.968	0.994	0.269	0.174	0.386
	(f) Full FT, norm-matched	0.974	1.367	0.013	0.048	0.165
	(d) Shuffled supervision	0.973	1.047	0.245	0.124	0.382

Table 19: Deployment options for explaining a finetuned target classifier (CIFAR10 set, 10-target means): the classifier whose predictions are explained; the accuracy of the classifier the user deploys (TVE: canonical / best of four probe pipelines); forward passes per image for prediction plus explanation; inference parameters beyond the deployed classifier; and wins over a random attribution on the classifier each method explains (10 targets × 4 faithfulness metrics; not a ranking across methods, since the explained classifiers differ).

Method	Explains	Acc.	Fwd.	+Params	Non-random
Proposed ($\tilde{\boldsymbol{\theta}}_{\mathrm{ft}\star}^{\mathrm{T}}$)	finetuned target itself	0.911	1	0	40/40
$\boldsymbol{\theta}_{\mathrm{ft}}^{\mathrm{T}}$ + Kernel SHAP ($P = 500$)	$\boldsymbol{\theta}_{\mathrm{ft}}^{\mathrm{T}}$	0.914	≈ 500	0	40/40
$\boldsymbol{\theta}_{\mathrm{ft}}^{\mathrm{T}}$ + amortized ($\alpha = 0$)	$\boldsymbol{\theta}_{\mathrm{ft}}^{\mathrm{T}}$	0.914	2	87.8M	38/40
TVE [57] (public checkpoint)	separate linear probe	0.779 / 0.817	3	244.9M	37/40

We do not compare absolute faithfulness values across methods, because the four methods explain different classifiers on different backbones and patch grids. The comparison across methods uses only the deployment characteristics and the wins over a random attribution.

L.1 Amortized Explainer with the Same Supervision

The $\alpha = 0$ model of the objective (4) is trained on ImageNet+X with the explanation loss only and is therefore an amortized explainer with the same supervision and the same backbone as $\boldsymbol{\theta}_{\mathrm{ft}\star}^{\mathrm{S}}$. We deploy it to each target by replacing the head $g_{\boldsymbol{W}}$ with the target's class-text embeddings, exactly as for zero-shot classification, with no target-side training of any kind. Its own classification accuracy is

Table 20: GT agreement of the $\alpha = 0$ amortized explainer (Amort.) on the 10 targets of Table 1, next to the zero-shot model (ZS) and the proposed method (Prop.). Underlined cells violate the ordering ZS $<$ Amort. $<$ Prop.

	E-RMSE (↓)			Rank corr. (↑)			IoU@1 (↑)			IoU@10 (↑)		
Dataset	ZS	Amort.	Prop.	ZS	Amort.	Prop.	ZS	Amort.	Prop.	ZS	Amort.	Prop.
Cars	1.623	1.118	1.068	−0.204	0.225	0.249	0.002	0.107	0.169	0.048	0.305	0.350
CIFAR100	1.508	1.233	1.223	−0.087	0.134	0.143	0.015	0.085	0.099	0.095	0.212	0.222
DTD	1.462	1.246	1.242	−0.061	0.145	0.141	0.021	0.037	0.048	0.102	0.177	0.172
EuroSAT	1.377	1.243	1.203	−0.009	0.109	0.143	0.018	0.032	0.072	0.115	0.170	0.189
GTSRB	1.543	1.011	0.889	−0.142	0.261	0.307	0.006	0.241	0.366	0.073	0.371	0.453
MNIST	1.536	1.136	1.077	−0.117	0.193	0.230	0.009	0.064	0.075	0.081	0.274	0.323
RESISC45	1.427	1.209	1.185	−0.016	0.163	0.175	0.016	0.080	0.104	0.116	0.213	0.227
STL10	1.548	1.132	1.094	−0.133	0.208	0.236	0.010	0.141	0.133	0.087	0.256	0.278
SUN397	1.519	1.259	1.248	−0.108	0.118	0.130	0.010	0.079	0.088	0.081	0.195	0.205
SVHN	1.534	1.111	1.018	−0.074	0.217	0.241	0.013	0.106	0.178	0.104	0.298	0.377
Mean	1.508	1.170	1.125	−0.095	0.177	0.200	0.012	0.097	0.133	0.090	0.247	0.279

between 0.003 and 0.110 across the targets, so in Table 19 it is paired with $\boldsymbol{\theta}_{\text{ft}}^{\text{T}}$, which provides the prediction while the amortized model provides the attribution (two forward passes). Our evaluation uses its attribution for the true class. The visual tower and head of the second model add 87.8M parameters.

GT agreement Table 20 evaluates the amortized explainer with the metrics and the 10 targets of Table 1 (including SUN397), next to the zero-shot model and the proposed method. The expected ordering (zero-shot $<$ amortized $<$ proposed) holds on all four 10-target means and in 37 of the 40 target–metric cells. The three exceptions are underlined in the table, where the amortized explainer exceeds the proposed method (Rank corr. and IoU@10 on DTD, IoU@1 on STL10). On IoU@5, which the table omits, the ordering holds on all 10 targets.

Faithfulness We score the amortized attribution on the image subsets of Appendix J, which keep the images that the transferred model classifies correctly. For each image we use the attribution for the true class y and compute the four metrics on the probability that $\boldsymbol{\theta}_{\text{ft}}^{\text{T}}$ assigns to y, as for the other attributions in Appendix J. It beats the random attribution on 38 of the 40 target–metric comparisons. The two losses are Fid+ and Del on EuroSAT, where Kernel SHAP on $\boldsymbol{\theta}_{\text{ft}}^{\text{T}}$ beats random on all four metrics.

L.2 Transferable Vision Explainer

TVE pretrains an explainer on ImageNet with a self-supervised objective and produces attributions for classifiers built on the backbone encoder used in its pretraining. Its training objective is self-supervised and does not use explanation supervision such as our source explanations. We use the official implementation[2] and checkpoint[3] (two ViT-Base backbones, 244.9M parameters in total) and its canonical configuration, in which a per-target linear probe is trained on the frozen encoder. In our evaluation TVE therefore explains this probe, not our finetuned CLIP classifier. Applying TVE to our classifier would require training an explainer for the CLIP backbone, which we do not do.

Probe protocol Probes are logistic-regression classifiers. The L-BFGS probes select the inverse regularization strength $C \in \{10^{-3}, \dots, 10^{3}\}$ on a 10% validation split and are refit on the full training set. The AdamW probes are trained for 50 epochs with learning rate 10^{-3}, a cosine schedule, and no weight decay. To rule out an under-tuned probe, Table 21 reports four pre-specified pipelines, which combine two optimizers (AdamW and L-BFGS) with two input normalizations. One normalization uses the ViT-MAE statistics of TVE's explainer (canonical, since the explainer and the encoder share one input tensor), and the other uses the encoder's native statistics (TVE's best case). The optimizer changes mean accuracy by at most 0.4 points, and the normalization changes it by 3.8 points. Even in the best case (0.817) the probe is 9.7 points below $\boldsymbol{\theta}_{\text{ft}}^{\text{T}}$ (0.914 on the same targets), and 13.5 points below it in the canonical configuration (0.779). This gap reflects our evaluation setting, in which the probe uses TVE's public frozen encoder instead of the finetuned CLIP backbone.

Faithfulness This check verifies that our TVE setup produces working explanations, so that the comparison in Table 19 is not driven by a broken baseline. Table 22 scores TVE's attributions

[2] `https://github.com/guanchuwang/TVE`

[3] `https://huggingface.co/Anonymous-researcher/leta`

Table 21: Test accuracy of TVE's per-target linear probe under four pipelines (optimizer × input normalization). "MAE + L-BFGS" is the canonical configuration used in Tables 19 and 22; "native" is TVE's best case.

Dataset	MAE+AdamW	MAE+L-BFGS	native+AdamW	native+L-BFGS
Cars	0.375	0.379	0.499	0.510
CIFAR10	0.934	0.938	0.965	0.970
CIFAR100	0.770	0.791	0.851	0.863
DTD	0.693	0.686	0.731	0.740
EuroSAT	0.953	0.954	0.958	0.955
GTSRB	0.643	0.643	0.722	0.724
MNIST	0.967	0.979	0.976	0.978
RESISC45	0.866	0.867	0.867	0.869
STL10	0.986	0.987	0.994	0.994
SVHN	0.562	0.565	0.565	0.564
Mean	**0.775**	**0.779**	**0.813**	**0.817**

Table 22: Faithfulness of TVE (canonical configuration) versus a random attribution, scored on the probe's own predicted class ($N = 500$ test images per dataset; first 500 images of the test split, except †EuroSAT, which uses a class-stratified subset because its first 500 test images cover only two of 10 classes). TVE's value is in bold where it beats random.

Dataset	Fid+ (↑) TVE / random	Fid− (↓) TVE / random	Ins (↑) TVE / random	Del (↓) TVE / random	Probe acc.
Cars	**0.228** / 0.162	**0.111** / 0.163	**0.192** / 0.143	**0.114** / 0.151	0.379
CIFAR10	**0.272** / 0.154	**0.098** / 0.166	**0.652** / 0.564	**0.506** / 0.591	0.938
CIFAR100	**0.328** / 0.226	**0.161** / 0.221	**0.443** / 0.366	**0.318** / 0.381	0.791
DTD	**0.129** / 0.113	**0.098** / 0.109	**0.610** / 0.563	**0.569** / 0.580	0.686
EuroSAT†	**0.148** / 0.111	**0.085** / 0.110	**0.651** / 0.613	**0.613** / 0.627	0.954
GTSRB	**0.422** / 0.200	**0.142** / 0.202	**0.493** / 0.407	**0.262** / 0.421	0.643
MNIST	**0.613** / 0.402	0.460 / 0.394	**0.428** / 0.421	**0.273** / 0.429	0.979
RESISC45	**0.227** / 0.139	**0.107** / 0.131	**0.587** / 0.536	**0.470** / 0.554	0.867
STL10	**0.072** / 0.032	**0.032** / 0.035	**0.784** / 0.736	**0.718** / 0.761	0.987
SVHN	**0.201** / 0.116	0.137 / 0.113	0.311 / 0.318	**0.268** / 0.323	0.565

(canonical configuration, with the attribution taken as the difference between the classifier logits of TVE's positive and negative outputs, as in the public TVE code) against a random attribution on the probe's own predicted class, with the four metrics of Appendix J, on 500 test images per target and TVE's 14×14 patch grid. TVE beats random on 37 of the 40 comparisons (the three losses are Fid− on MNIST and SVHN and insertion on SVHN). TVE thus explains its probe better than a random attribution. The table does not include our method, because faithfulness values depend on the classifier whose probabilities are scored, and the two methods explain different classifiers on different patch grids. Our method is compared with TVE in Table 19, where the difference lies in which classifier is explained and at what accuracy and cost.

M Selecting λ_2 without Target-Side Explanation Supervision

The experiments of Section 5 use $\alpha = 0.8$ and $\lambda_2 = 0.7$ for all targets. These values are a representative operating point on the accuracy–explanation trade-off rather than optimal values, because the right balance depends on the user's task. α is a source-side training hyperparameter and can be selected on source validation data alone. We fix $\alpha = 0.8$ as a representative source-side point rather than by such a rule, and Figure 2 shows the sensitivity to $\alpha \in \{0.1, \ldots, 0.9\}$. This appendix documents the sweep behind λ_2 and two held-out selection protocols that never use the explanation supervision of the target being selected for.

Sweep Table 23 reports the 10-target means (CIFAR10 set, $\lambda_1 = 1.0$, $\alpha = 0.8$) over the grid $\lambda_2 \in \{0.4, 0.5, \ldots, 1.2\}$. Explanation metrics improve up to a plateau at $\lambda_2 \approx 1.0$–1.1 (Rank corr. spread 0.002 over $[0.9, 1.1]$), while accuracy decreases monotonically over the whole grid (2.0 points from 0.4 to 1.2). Accuracy falls by 0.3 points from 0.4 to 0.7 and by 1.7 points from 0.7 to 1.2. The grid does not include $\lambda_2 = 0$, so neither protocol below can select "no explainability vector".

Table 23: Sensitivity of the proposed method to λ_2 (10-target means on the CIFAR10 set, full test split, $\lambda_1 = 1.0$, $\alpha = 0.8$).

λ_2	Accuracy (↑)	Rank corr. (↑)	IoU@10 (↑)	E-RMSE (↓)
0.4	0.914	0.120	0.221	1.233
0.5	0.914	0.161	0.255	1.175
0.6	0.913	0.186	0.272	1.138
0.7 (default)	0.911	0.204	0.281	1.118
0.8	0.909	0.213	0.285	1.109
0.9	0.907	0.217	0.286	1.104
1.0	0.903	0.219	0.287	1.102
1.1	0.899	0.219	0.286	1.102
1.2	0.894	0.218	0.285	1.103

Table 24: Leave-one-dataset-out selection of λ_2 on the CIFAR10 set. The Explanation-only columns give the λ_2 maximizing the mean Rank corr. or IoU@10 of the nine other targets. The Accuracy-constrained columns give the λ_2 maximizing the mean Rank corr. of the nine other targets among the values whose mean accuracy is within ε of the best on the grid. The last row gives the mean held-out Rank corr. gain of the selected value over $\lambda_2 = 0.7$.

	Explanation-only		Accuracy-constrained	
Held-out target	Rank corr.	IoU@10	$\varepsilon = 0.5$pt	$\varepsilon = 1.0$pt
Cars	1.1	1.0	0.9	1.1
CIFAR10	1.0	1.0	0.7	0.9
CIFAR100	1.1	1.0	0.7	0.9
DTD	1.0	1.0	0.8	1.0
EuroSAT	1.1	1.0	0.7	0.9
GTSRB	1.1	0.9	0.7	0.9
MNIST	1.1	1.1	0.7	0.9
RESISC45	1.0	1.0	0.7	0.9
STL10	1.0	1.0	0.7	0.9
SVHN	1.1	1.0	0.7	0.9
Within 0.7 ± 0.1	0/10	0/10	9/10	0/10
Mean (chosen $-$ 0.7) on Rank corr.	$+0.015$	—	$+0.001$	$+0.014$

Leave-one-dataset-out selection on explanation metrics For each held-out target, we choose the λ_2 that maximizes the mean explanation metric over the other nine targets (the metric is fixed in advance, and the held-out target's labels are never used) and evaluate the held-out target at that value. The Explanation-only columns of Table 24 show that this protocol picks 1.0–1.1 for every held-out target under Rank corr. and for nine of the 10 under IoU@10 (GTSRB picks 0.9), and picks 0.7 ± 0.1 for none of them. If λ_2 had been chosen by maximizing explanation metrics on the benchmark targets, the result would have been 0.9–1.1, not 0.7. Compared with $\lambda_2 = 0.7$, the selected value raises held-out Rank corr. by 0.015 on average and by at most 0.033 (MNIST), and the value selected under IoU@10 changes held-out IoU@10 by at most 0.011 (SVHN), so the conclusions of Table 1 do not depend on landing on a narrow optimum. This argument rules out selection by argmax on target explanation metrics. It does not rule out that the choice of 0.7 was informed by the aggregate sweep, and we report it as a representative operating point.

Accuracy-constrained selection This protocol fixes in advance a tolerance ε, in accuracy points, on the accuracy loss that the deployment can accept. Among the grid values of λ_2 whose accuracy is within ε of the best accuracy on the grid, it selects the one with the best explanation metric on other datasets. The rule thus needs no explanation labels on the target being selected for. We run this rule leave-one-dataset-out on the benchmark records, with accuracy and Rank corr. both averaged over the other nine targets, so the held-out target's own labels are not used. In deployment, the accuracy term would instead come from the target's validation classification labels, which we do not evaluate separately. The Accuracy-constrained columns of Table 24 show that with $\varepsilon = 0.5$ points it selects exactly 0.7 for eight targets and 0.7 ± 0.1 for nine of the 10, while $\varepsilon = 1.0$ points moves the selection to 0.9 for most targets. This procedure and the value $\varepsilon = 0.5$ points were both chosen after seeing the explanation-only result above, for illustration. We include it as a practical selection rule that recovers the adopted operating point, not as evidence that 0.7 was derived this way.

Table 25: Explanations on two-task images (2,000 shared test images). Region rates are the fraction of the top-k self-explanation patches ($k = \min(5, |\text{region}|)$) that fall in each object region. Chance is the region's share of the 49 patches. The last row uses a fixed top-5 (near-identical to the adaptive k in this setting).

	Digit model	Clothing model
Accuracy, with $\boldsymbol{\tau}_\star^{\mathrm{S}}$ / without	0.9915 / 0.9925	0.9165 / 0.9195
Top-k patches in the digit region	**56.8**% (3.7× chance)	8.3%
Top-k patches in the clothing region	42.5%	**91.6**% (4.1× chance)
Chance rate (digit / clothing region)	15.2% / 22.2%	
Own-region rate without $\boldsymbol{\tau}_\star^{\mathrm{S}}$ (top-5)	1.0%	4.6%

N Task-Dependent Explanations on Two-Task Images

Visual explanations depend on the task. The same image calls for attention to different regions depending on the task, for example when it is classified by object category or by scene. A natural concern is therefore whether a single explainability vector learned on one source can yield task-appropriate explanations on a target whose task differs. This appendix tests this directly by constructing images that carry two classification tasks at once.

Setup We composite an MNIST digit [38] and a FashionMNIST clothing item [62] at non-overlapping random positions of a 224×224 canvas (12,000 training and 2,000 test images), so that every image has both a digit label and a clothing label, together with the ground-truth set of patches covering each object (7.5 patches on average for the digit, 10.9 for the clothing item, out of $M = 49$). Two prediction-only models are finetuned with classification supervision only, one for the digit task and one for the clothing task (test accuracy 0.9925 and 0.9195, respectively, using the MNIST prompt template for digits and the generic "a photo of a {class}" template for clothing). The same explainability vector $\boldsymbol{\tau}_\star^{\mathrm{S}}$ learned on ImageNet+X is then added to both models with $\lambda_2 = 0.7$, as in Section 5.1. On the shared 2,000 test images, we measure which object region the top-k patches of each model's self-explanation fall into. The self-explanation is the attribution for the predicted class, and $k = \min(5, |\text{region}|)$ handles small digit regions. The chance rate is the region's share of the 49 patches, which is 15.2% for the digit region and 22.2% for the clothing region.

Results Table 25 shows that the same vector steers the attributions of each model to the region of its own task. The digit model places 56.8% of its top-k patches on the digit (3.7× chance), and the clothing model places 91.6% on the clothing item (4.1× chance). Adding the vector lowers accuracy by at most 0.3 points. The per-image paired difference between the two models in the digit-region rate has a bootstrap 95% interval of $[+0.475, +0.496]$, and that in the clothing-region rate has $[-0.501, -0.481]$ (10^4 resamples). A fixed top-5 variant and a continuous attribution-mass variant give the same conclusion. The effect is attributable to the vector. Without $\boldsymbol{\tau}_\star^{\mathrm{S}}$, the digit model places only 1.0% of its top-5 patches on the digit, which is below chance. The clothing model then places only 4.6% on the clothing item and 65.8% on the digit instead. Classification-only finetuning with a frozen head does not directly supervise patch-level attributions. Figure 11 shows examples in which the heatmap moves between the digit and the clothing item within the same pixels as the task changes. The separation is not complete, since 42.5% of the digit model's top patches fall on the larger clothing region. Composited images are also simpler than natural scenes, and the validity on natural images rests on the 10-dataset experiments with the same vector.

Interpretation This experiment supports the following reading of the transfer rule (8), which Appendix K supports independently. The target-side terms, namely the target task vector $\boldsymbol{\tau}_{\mathrm{ft}}^{\mathrm{T}}$ and the class-text embeddings in $g_{\boldsymbol{W}}$, determine which evidence the model attends to. The vector $\boldsymbol{\tau}_\star^{\mathrm{S}}$ adds the ability to read that task-specific evidence out as patch-level attributions. The result is consistent with this reading, since the same vector serves both tasks without any information about them.

O Limitations

Architecture The current method is specifically designed for Vision Transformer (ViT) models pretrained with vision-language objectives such as CLIP. It relies on the patch token structure of ViTs and the shared linear head $g_{\boldsymbol{W}}$ constructed from text embeddings. It also assumes that the source and target models are finetuned from the same pretrained checkpoint with this head kept

frozen. Prediction-only classifiers whose head is replaced or trained for the target task, as is common in transfer learning, are not directly covered, because their parameters do not lie in a shared space across label sets. Extending the approach to other architectures, such as CNNs or ViTs trained with different strategies (e.g., supervised pretraining [18], MAE [24], DINO [7]), would require architecture-specific design considerations and is an important direction for future work.

Explanation type This work focuses on patch-level attributions as the form of visual explanation. Other types of explanations, such as concept-based explanations [34, 35] or prototype-based explanations [10], are not covered by the current framework. Generalizing the idea of transferring explainability via task arithmetic to these alternative explanation forms is a promising future direction.

Evaluation scope Our primary metrics measure agreement with a user-specified explanation target. Faithfulness is evaluated within the transferred model (Section 5.1.5 and Appendix J), and we do not rank explanation quality across methods whose explained classifiers differ. The control experiments (Appendix K) show that part of the transferred effect is a content-free attribution prior. A vector trained on shuffled explanation supervision recovers most of the coarse ranking metrics but only 57.4% of the IoU@1 improvement, and it is below the vector trained on real supervision on IoU@1 for 9 of the 10 targets. The two-task experiment (Appendix N) uses composited images with exact region masks, and its validity on natural images rests on the 10-dataset experiments.

When the target must not be modified Our method adds a vector to the parameters of the target classifier, and its accuracy may change slightly. When the deployed model must remain unmodified, e.g., for certification or compliance reasons, external explainers are the appropriate choice. Transferable explainers such as TVE [57] require an explainer trained for the target's backbone, and in our evaluation with its public checkpoint TVE explains a separate linear probe (Appendix L). Post-hoc methods such as Kernel SHAP come at the cost of hundreds of forward passes per image (Section 5.1.5).

Scope of the predictability finding The predictability examined in Section 5.2 holds for a fixed source with held-out targets, and on most explanation-quality metrics but not all. On ViT-B/32, IoU@1 prediction is weak (Ridge $r = +0.33$, the better of the two regressors, Appendix C), so the six features do not capture which single patch is most important. On ViT-L/14, the raw Δ_q target is not predictable due to per-target variance. Under the oracle-normalized target R_q (Appendix C), three of five metrics become significant at uncorrected $p < 0.05$ (IoU@1, E-MSE, and Rank corr.), while IoU@5 and IoU@10 remain unpredictable. R_q requires per-target oracle scores only at predictor-fitting time. Predicting the outcome on a new target itself uses only the six model-internal features and no target-side explanation supervision. Closing the remaining metric gap, e.g., by enlarging N beyond 10 source–target pairs or by introducing scale-aware features, remains future work. The held-out 56-pair evaluation in Appendix F.2 also relies on source fixed effects to absorb per-source intercepts. Under leave-one-source-out evaluation, where such intercepts are unavailable, none of 40 predictor configurations (including ones given oracle features of the target) meets our criteria. The feasibility result is therefore confined to source datasets observed at predictor-fitting time, and prediction for genuinely unseen sources remains open.

P Impact Statement

This paper presents a method for transferring visual explainability from self-explaining to prediction-only VLM-based image classifiers. Our approach requires explanation supervision only once on the source domain and reuses it across target domains, so users need no target-side explanation supervision or training, and inference stays a single forward pass. The one-time source-side cost (about 100 GPU-hours in our setup, Appendix I) is borne by the provider of the explainability vector. This can lower a barrier to adopting explainable AI in resource-constrained scenarios. As a result, this work may help users inspect the decisions of AI systems and improve their transparency.

However, there are potential risks associated with our method. Since the transferred explanation capability is not necessarily optimized for the target domain of the prediction-only model, it may produce lower-quality or misleading explanations. Users relying on such explanations could make incorrect decisions based on misinterpreted model behavior. Plausible-looking explanations can also raise trust beyond what is warranted, especially since part of the transferred effect is a content-free attribution prior (Appendix K). We encourage practitioners to carefully evaluate the quality of transferred explanations using faithfulness checks such as those of Appendix J, or a small set of

explanation labels when available, and, when necessary, to finetune the model with a small amount of labeled explanation data to mitigate these risks.

Q Additional Visualization Examples

We provide additional randomly selected visualization examples for each target dataset to complement the examples shown in Figure 3. The examples use CLIP ViT-B/32 and the same models as Table 1. Each figure shows 10 randomly selected samples from the corresponding dataset, highlighting the top-10 patches of each attribution, as in Figure 3.

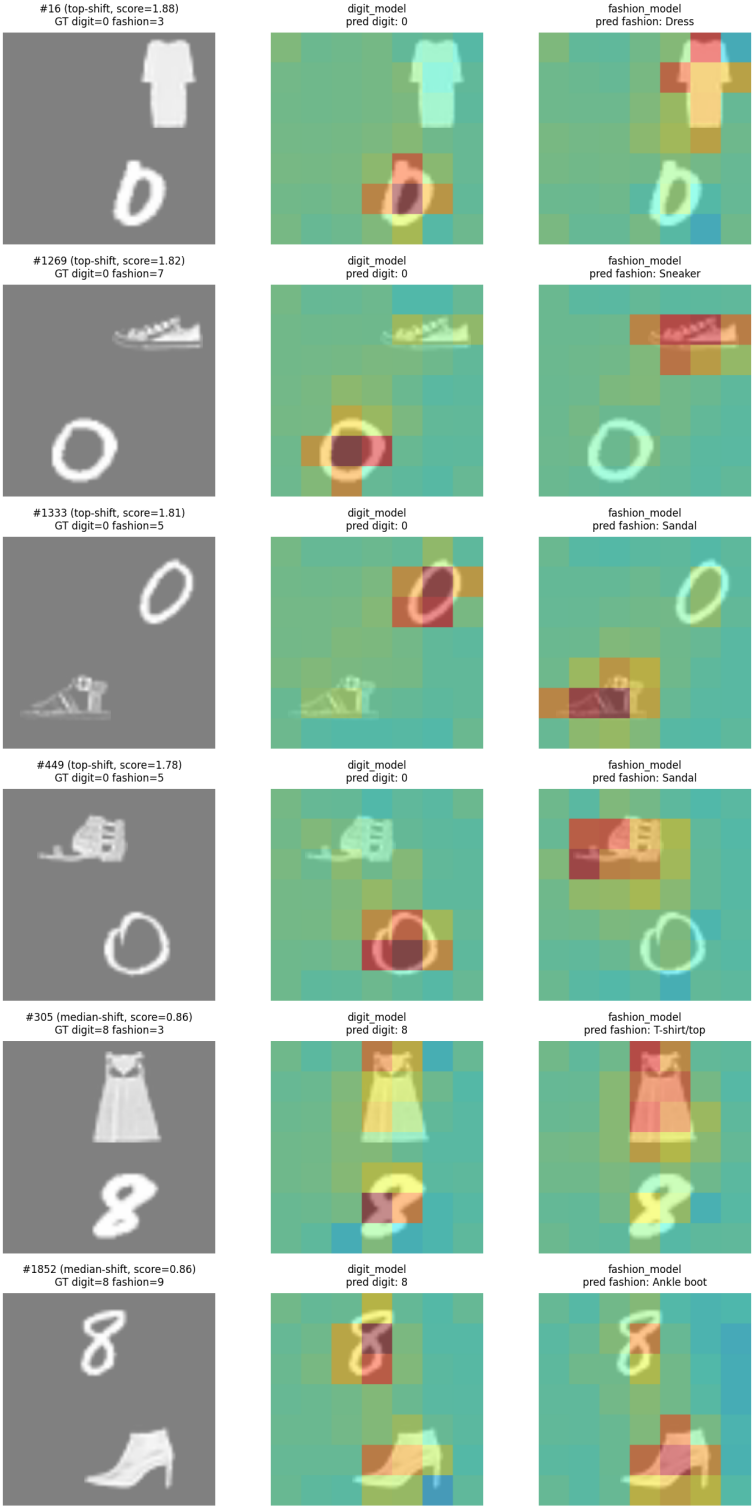
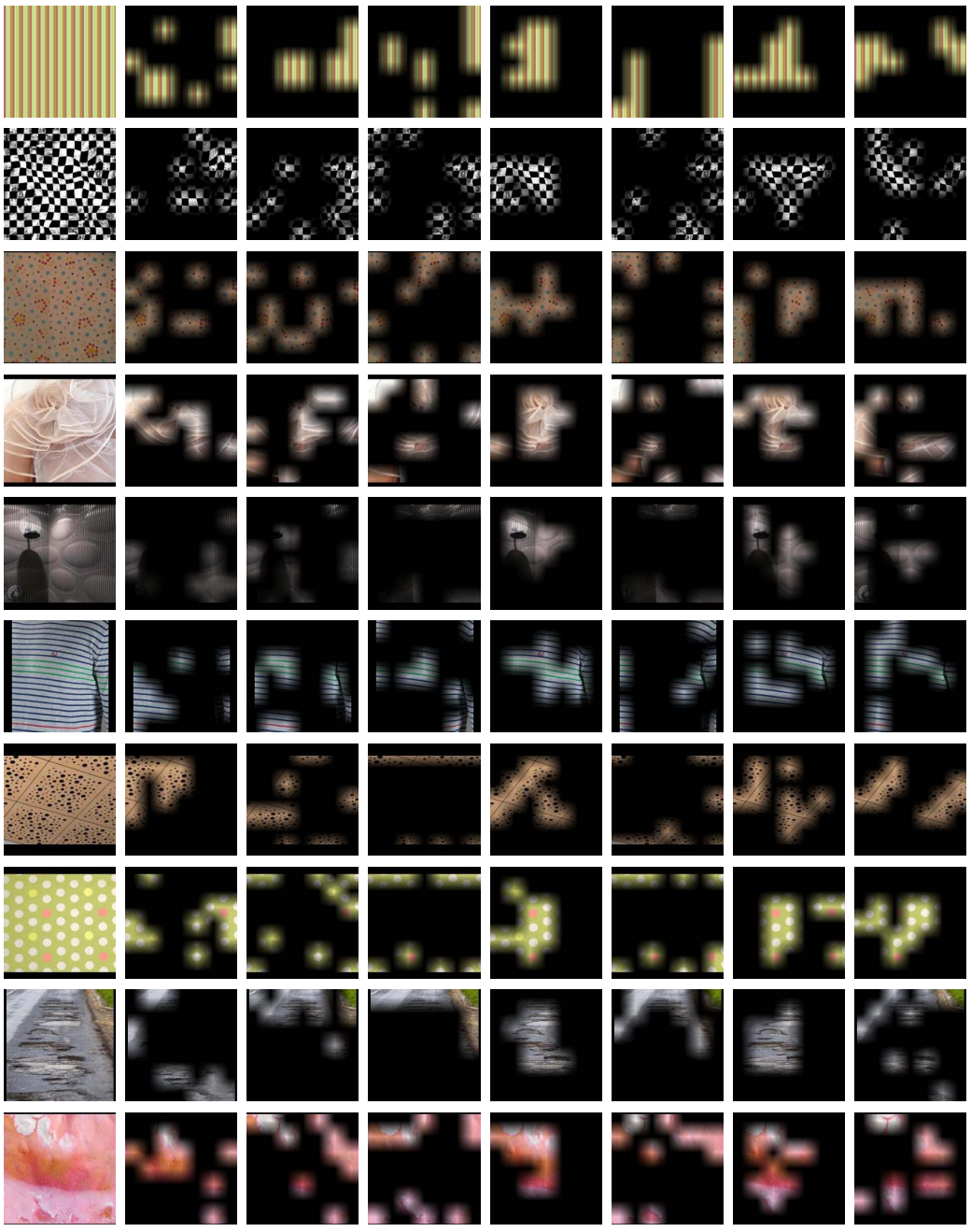


Figure 11: Self-explanations of the digit model (middle) and the clothing model (right) on the same composited test images (left), both obtained by adding the same $\tau_\star^{\mathrm{S}}$ to a classification-only model. Heatmaps show the attribution for the predicted class (7×7 patches, shared color scale per row). The first four rows are the test images with the largest attribution shift between the two models, and the last two are images at the median shift. All six are classified correctly by both models.



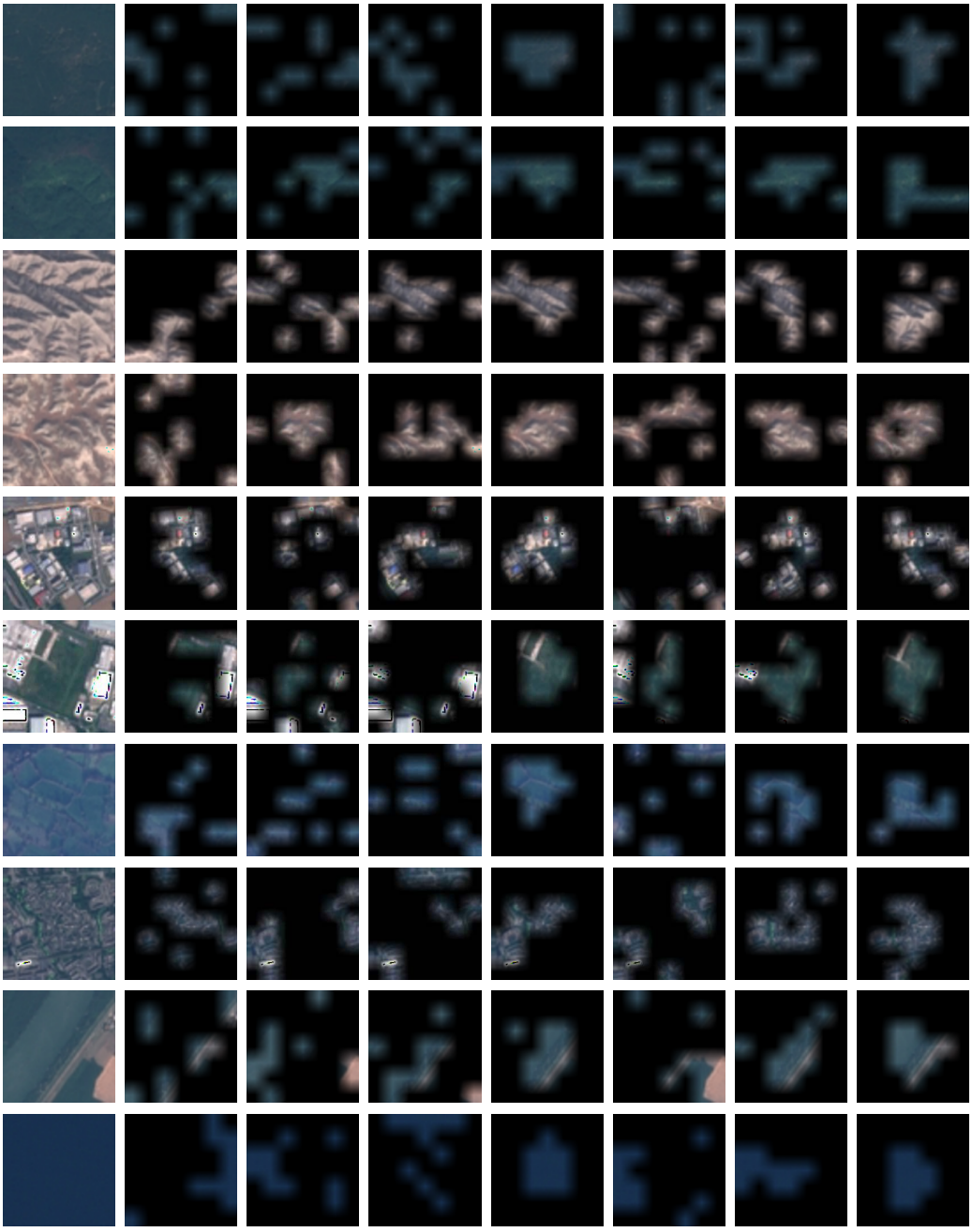
(a) Input (b) Ground-truth (c) Zero-shot (d) Finetuned on ImageNet (e) Finetuned on ImageNet+X (f) Finetuned on target (g) Proposed (h) Finetuned on target+X

Figure 12: Randomly selected visualization examples for Cars.



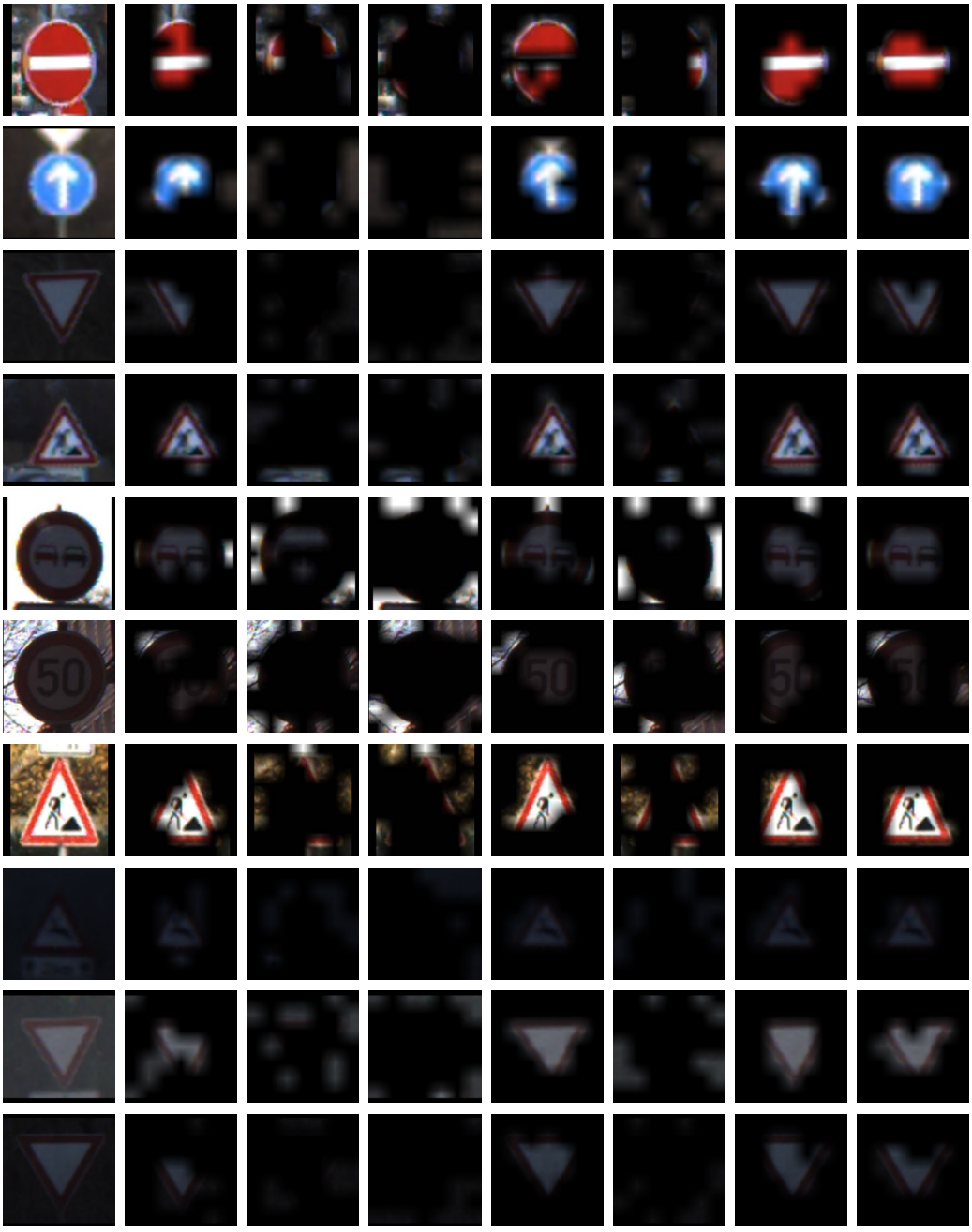
(a) Input (b) Ground-truth (c) Zero-shot (d) Finetuned on ImageNet (e) Finetuned on ImageNet+X (f) Finetuned on target (g) Proposed (h) Finetuned on target+X

Figure 13: Randomly selected visualization examples for DTD.



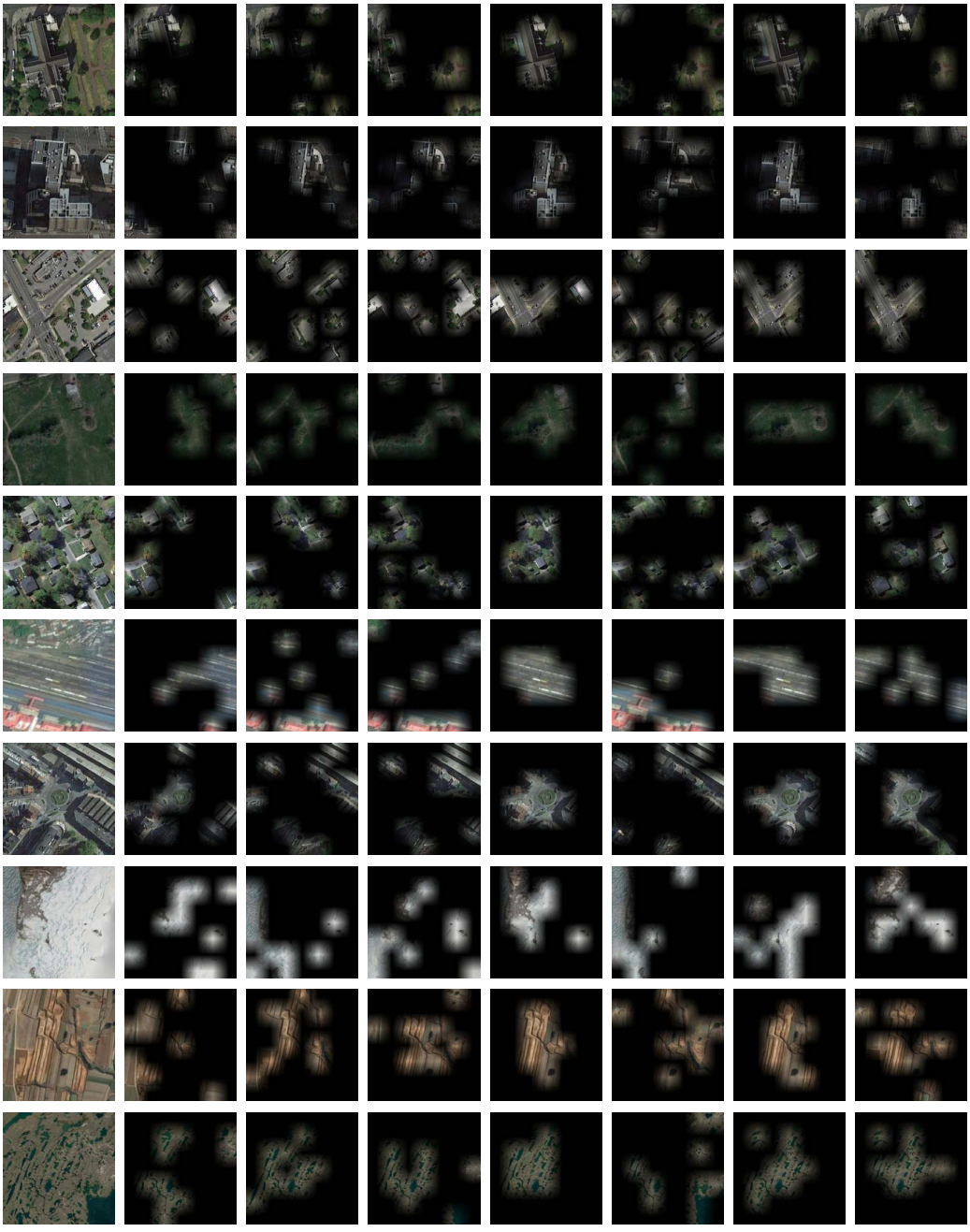
(a) Input (b) Ground-truth (c) Zero-shot (d) Finetuned on ImageNet (e) Finetuned on ImageNet+X (f) Finetuned on target (g) Proposed (h) Finetuned on target+X

Figure 14: Randomly selected visualization examples for EuroSAT.



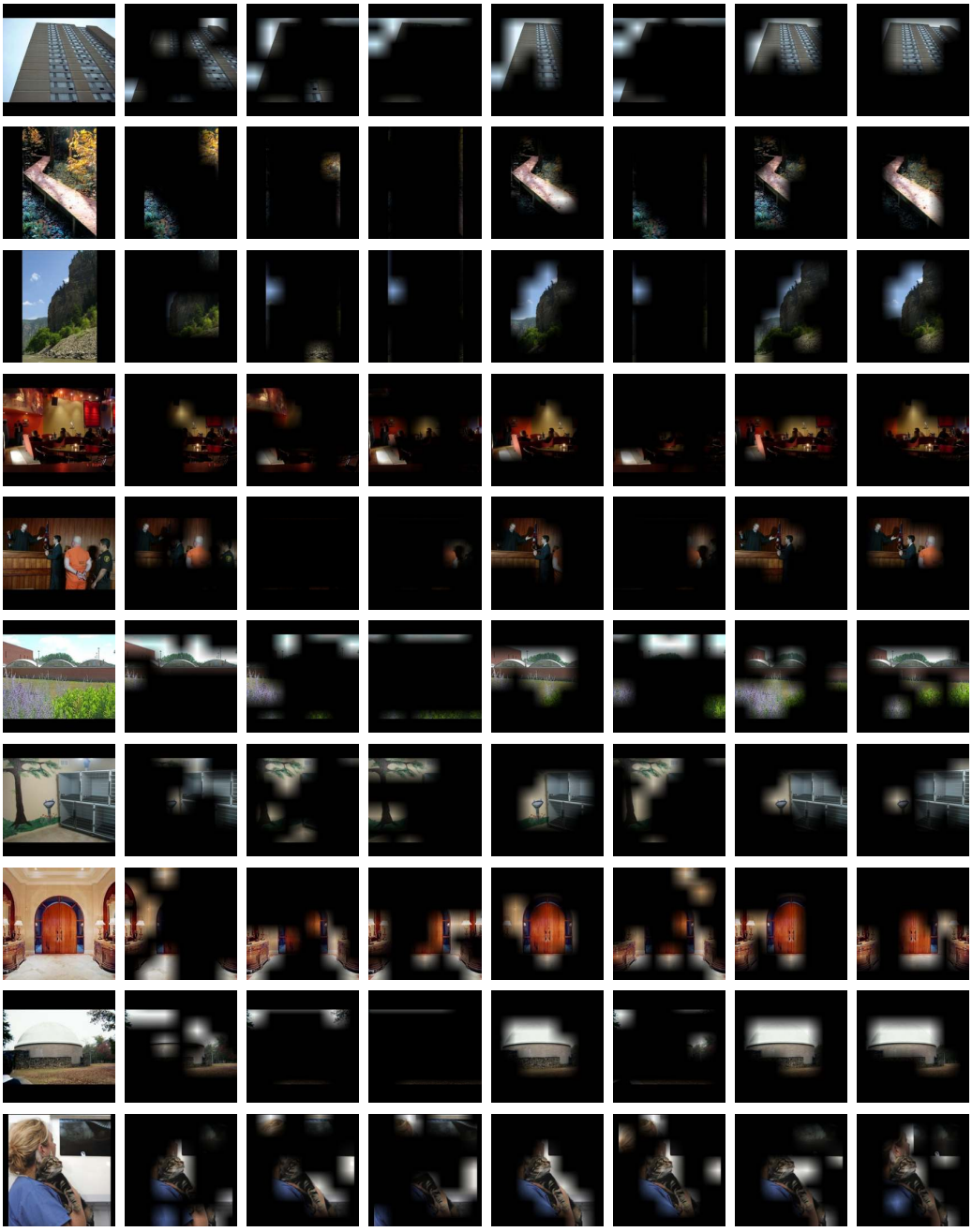
(a) Input (b) Ground-truth (c) Zero-shot (d) Finetuned on ImageNet (e) Finetuned on ImageNet+X (f) Finetuned on target (g) Proposed (h) Finetuned on target+X

Figure 15: Randomly selected visualization examples for GTSRB.



(a) Input (b) Ground-truth (c) Zero-shot (d) Finetuned on ImageNet (e) Finetuned on ImageNet+X (f) Finetuned on target (g) Proposed (h) Finetuned on target+X

Figure 16: Randomly selected visualization examples for RESISC45.



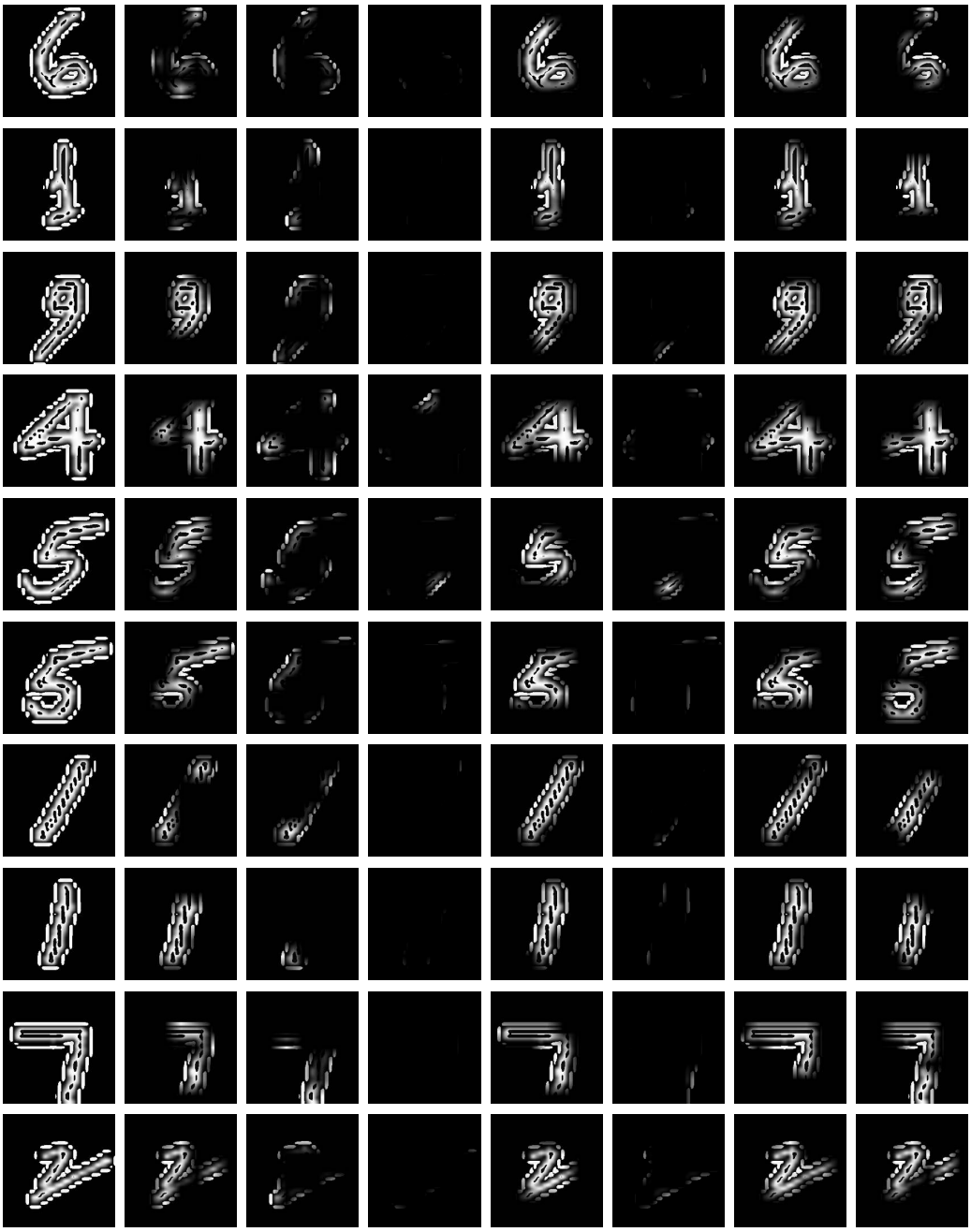
(a) Input (b) Ground-truth (c) Zero-shot (d) Finetuned on ImageNet (e) Finetuned on ImageNet+X (f) Finetuned on target (g) Proposed (h) Finetuned on target+X

Figure 17: Randomly selected visualization examples for SUN397.



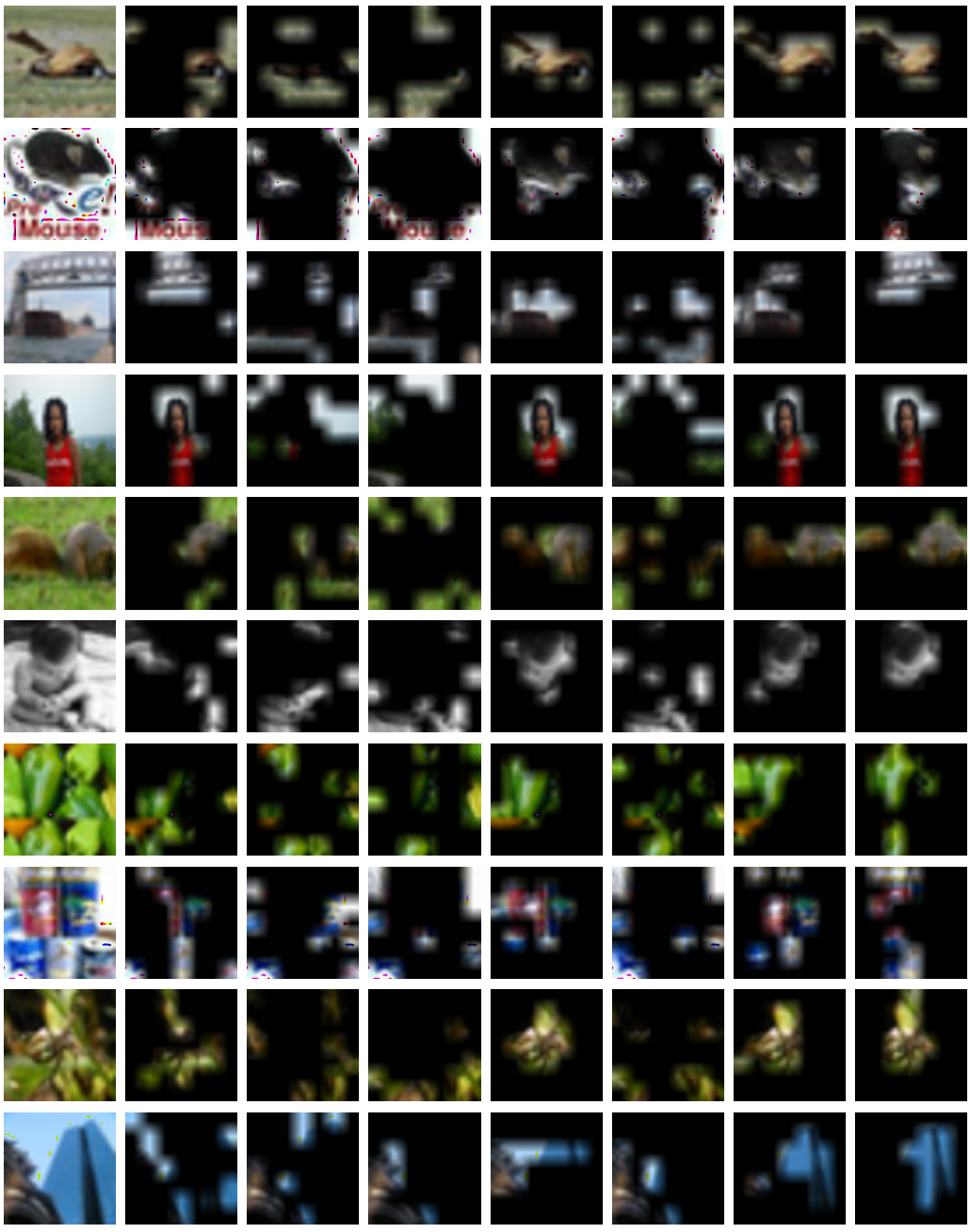
(a) Input (b) Ground-truth (c) Zero-shot (d) Finetuned on ImageNet (e) Finetuned on ImageNet+X (f) Finetuned on target (g) Proposed (h) Finetuned on target+X

Figure 18: Randomly selected visualization examples for SVHN. Visual artifacts in the input images are due to the denormalization and upscaling process.



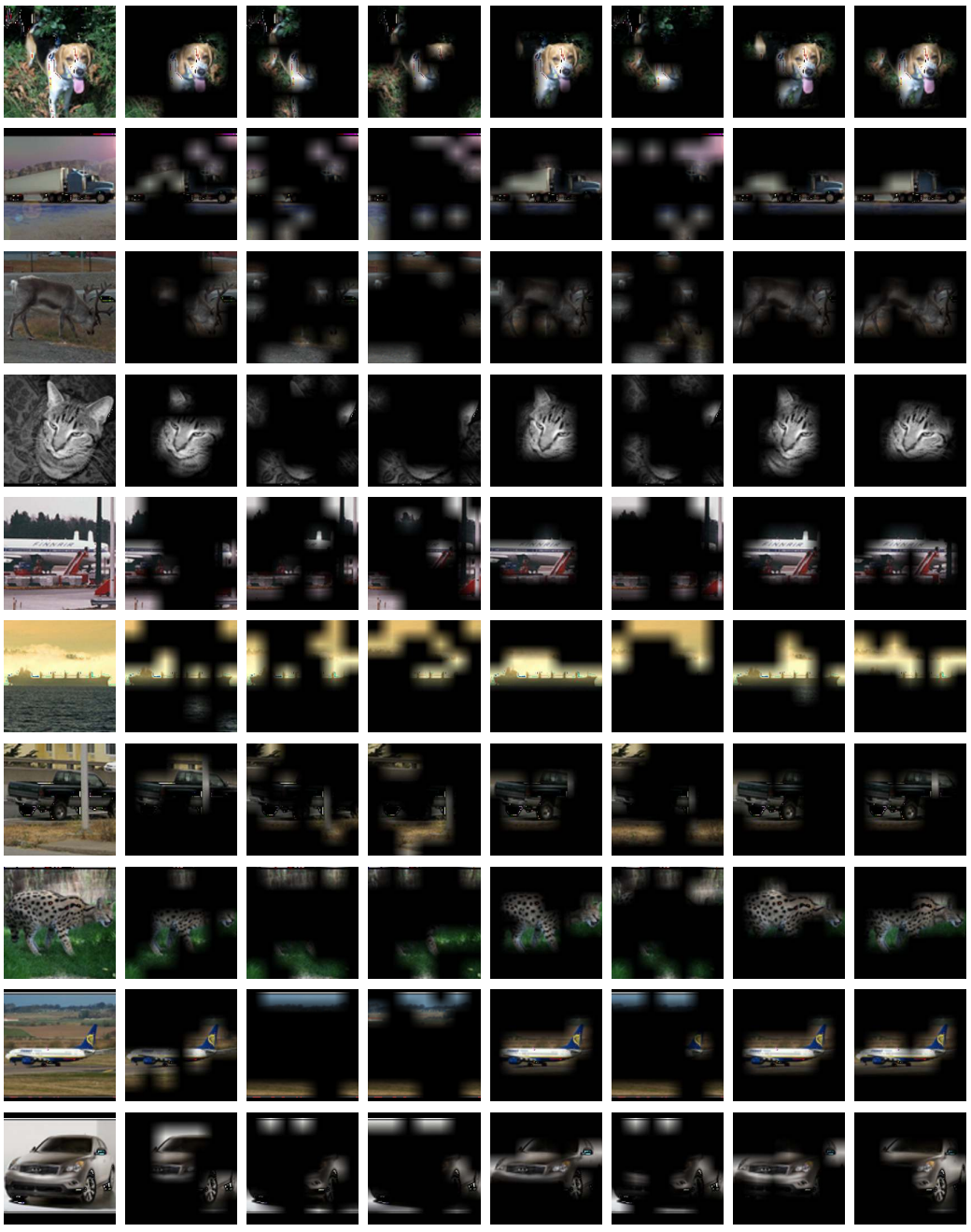
(a) Input (b) Ground-truth (c) Zero-shot (d) Finetuned on ImageNet (e) Finetuned on ImageNet+X (f) Finetuned on target (g) Proposed (h) Finetuned on target+X

Figure 19: Randomly selected visualization examples for MNIST. Visual artifacts in the input images are due to the denormalization and upscaling process.



(a) Input (b) Ground-truth (c) Zero-shot (d) Finetuned on ImageNet (e) Finetuned on ImageNet+X (f) Finetuned on target (g) Proposed (h) Finetuned on target+X

Figure 20: Randomly selected visualization examples for CIFAR100.



(a) Input (b) Ground-truth (c) Zero-shot (d) Finetuned on ImageNet (e) Finetuned on ImageNet+X (f) Finetuned on target (g) Proposed (h) Finetuned on target+X

Figure 21: Randomly selected visualization examples for STL10.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Section 1 states three contributions. (i) The task-arithmetic-based transfer method without target-side explanation supervision is introduced in Section 4.3. (ii) The consistent transfer of a single explainability vector learned on ImageNet-1k to 10 diverse target datasets, together with its faithfulness and specificity, is shown in Section 5.1. (iii) The predictability of transfer success is examined in Sections 4.4 and 5.2. The scope is stated explicitly. The method targets prediction-only, zero-shot-capable VLM-based ViTs, the faithfulness and control-vector evidence is reported in Section 5.1.5, and the predictability claim is restricted to a fixed source with held-out targets (Section 5.2 and Appendix F.2).

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Appendix O discusses five limitations. (i) The method is tailored to ViT-based vision-language models such as CLIP and would require architecture-specific redesign for CNNs or ViTs trained with other strategies. (ii) The framework targets patch-level attributions and does not directly cover concept- or prototype-based explanations. (iii) The evaluation scope is limited, with GT agreement as the primary metric, faithfulness evaluated within the transferred model only, part of the transferred effect being a content-free attribution prior, and the two-task experiment using composited images. (iv) The method modifies the target model and is not appropriate when the deployed model must remain unmodified. (v) The transfer-success predictor is unreliable on a subset of metrics (IoU@1 on ViT-B/32, and IoU@5 and IoU@10 on ViT-L/14 under the oracle-normalized target) and does not generalize to a held-out source under leave-one-source-out evaluation.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.

- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Appendix G derives two theoretical results step by step. (i) The explainability vector $\boldsymbol{\tau}_\star^{\mathrm{S}}$ is the optimal perturbation under a local quadratic approximation of the training objective in (4). (ii) Under a local quadratic approximation around $\boldsymbol{\theta}_{\mathrm{ft}}^{\mathrm{T}}$, adding $\lambda_2 \boldsymbol{\tau}_\star^{\mathrm{S}}$ increases the target classification loss by $\frac{\lambda_2^2}{2} R(\boldsymbol{\tau}_\star^{\mathrm{S}}; \boldsymbol{H}^{\mathrm{T}})$ with $R(\boldsymbol{\tau}_\star^{\mathrm{S}}; \boldsymbol{H}^{\mathrm{T}}) = (\boldsymbol{\tau}_\star^{\mathrm{S}})^\top \boldsymbol{H}^{\mathrm{T}} \boldsymbol{\tau}_\star^{\mathrm{S}}$. The underlying assumptions (near-stationary points such as $\nabla \mathcal{L}_{\mathrm{cls}}(\boldsymbol{\theta}_{\mathrm{ft}}^{\mathrm{S}}) \approx \mathbf{0}$ and small-perturbation Taylor expansions) are stated explicitly there.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The common experimental protocol (CLIP ViT-B/32 backbone, 10 publicly available target datasets, evaluation metrics, and ground-truth attribution generation via Kernel SHAP with 500 perturbations) is specified in Sections 5 and 5.1. The method-specific hyperparameters (α, λ_1, λ_2) and the reported choices are stated in Sections 4.3 and 5. The predictor pool construction ($N = 10$ ImageNet-source pairs with leave-one-target-out evaluation, six model-internal features, and the Lasso/Ridge training protocol) is documented in Section 5.2 together with Appendix C. The complete computing environment (hardware, libraries, and versions) is reported in Appendix D, and the ViT-L/14 setup is given in Appendix E.1. The protocols of the additional experiments (within-model faithfulness metrics and evaluation subsets, control-vector construction, the amortized and TVE baselines including the probe protocol, the leave-one-dataset-out selection of λ_2, the leave-one-source-out evaluation with bootstrap intervals, the two-task experiment, and the cost measurements) are specified in Appendices J–N, F.2, and I, including seeds and subset-construction rules.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.

- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The code is publicly available at the repository URL given in the abstract, and its README gives instructions for reproducing the main experimental results and the additional experiments of the appendix. All datasets are publicly available benchmarks, and the ground-truth attributions are generated by the released code.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 5 fixes the backbone (CLIP ViT-B/32 with the DataComp-1B pretrain), the input resolution (224×224), the patch size (32×32), and the number of patches ($M = 49$). It also states $\lambda_1 = 1.0$, the chosen values $\alpha = 0.8$ and $\lambda_2 = 0.7$, and the sweep ranges $\alpha \in \{0.1, \ldots, 0.9\}$ and $\lambda_2 \in \{0.4, \ldots, 1.2\}$. Section 5.1 describes the ground-truth attribution generation (Kernel SHAP with 500 perturbations on the classifier finetuned on each dataset). The ViT-L/14 counterparts (224×224 input, 16×16 patches of size 14×14, $M = 256$, and Grad-CAM ground truth) are given in Appendix E.1. The remaining training-loop details (optimizer, learning rate, batch size, and number of epochs) are provided as configuration files in the public code repository. The settings of the additional experiments (evaluation subsets and metrics, control vectors, baselines, and the construction of the two-task images) are given in Appendices J–N.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Table 1 reports the mean and standard deviation across the 10 target datasets and marks, per metric, the methods with no statistically significant difference from the best (paired t-test, $p < 0.05$), as stated in the table caption. The same convention is used for the ViT-L/14 results in Appendix E.2 and for the shared- vs. separated-head comparison in Appendix A. Figure 2 reports 95% bootstrap intervals of the mean over the nine α values, and the Kernel SHAP comparison figure in Appendix H reports 95% confidence intervals. The transfer-success predictor results in Section 5.2 and Appendix C report permutation-test p-values for the held-out Pearson correlations, and Appendix F.2 adds bootstrap 95% intervals (pair-level, and source-level cluster intervals for the leave-one-source-out evaluation). The two-task experiment in Appendix N reports paired bootstrap 95% intervals. The faithfulness and control-vector results in Appendices J and K are reported per target as win counts and per-target tables rather than as aggregate significance tests. The circuit-overlap analysis in Appendix B.2 reports paired Wilcoxon p-values.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.

- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix D reports the compute environment. The main experiments use Rocky Linux machines with two Intel Xeon Platinum 8558 2.1GHz (48-core) CPUs, 2,048 GB of RAM, and eight NVIDIA H200 SXM GPUs with 141 GB of memory, with the code implemented in Python 3.9.21 and PyTorch 2.6. The additional experiments of the appendix were run on NVIDIA RTX 6000 Ada GPUs. Appendix I reports the one-time source-side cost (100.4 GPU-hours for the Kernel SHAP supervision and source training) and the per-image inference cost of each compared method.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work uses only publicly available image classification benchmarks (ImageNet-1k, the 10 target datasets listed in Section 5.1, and FashionMNIST for the two-task images), does not involve human subjects, and does not collect or release any new personal data. The authors have reviewed the NeurIPS Code of Ethics, and the research conforms to it.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Appendix P discusses both sides. On the positive side, the explanation supervision is needed only once on the source, which can lower the barrier to adopting explainable AI in resource-constrained scenarios and may help users inspect the decisions of AI systems. On the negative side, the transferred explanation capability is not necessarily optimized for the target domain and may produce lower-quality or misleading explanations on which users could base incorrect decisions. Plausible-looking explanations can also raise

trust beyond what is warranted, since part of the transferred effect is a content-free attribution prior. As mitigations, the section recommends checking the transferred explanations with faithfulness checks or a small set of explanation labels and, when necessary, finetuning with a small amount of labeled explanation data.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release any high-risk artifact. No generative model, no language model, and no scraped dataset is introduced. The only released artifact is a code repository that finetunes a publicly available CLIP backbone on standard image classification benchmarks and computes patch-level attributions, which does not pose a misuse risk that would require additional safeguards.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

Justification: All existing assets used in the paper are credited via citations to their original publications: the CLIP backbone with its DataComp-1B pretrain [49, 19, 27], ImageNet-1k [17] as the source dataset, the 10 target datasets [36, 14, 25, 55, 12, 63, 45, 38, 37, 15], the `pytorch-grad-cam` [21] library used to generate the ViT-L/14 ground-truth attributions, FashionMNIST [62] used in the two-task experiment of Appendix N, and the public TVE checkpoint [57] used as a baseline in Appendix L. All of these are publicly available research assets used in accordance with their stated terms of use.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The only new asset introduced by the paper is the public code repository. It covers source and target finetuning, explainability-vector transfer via task arithmetic, evaluation on the 10 target datasets, the additional experiments of the appendix, and the script that composites the two-task images from MNIST and FashionMNIST. The repository is accompanied by a README describing dependencies, license, and entry-point commands.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing or research with human subjects. All experiments are conducted on publicly available image classification benchmarks.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve research with human subjects. All experiments are conducted on publicly available image classification benchmarks, so no IRB approval is required.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core methodology is the explainability transfer between CLIP-based ViT image classifiers via task arithmetic. LLMs are not used as an important, original, or non-standard component of this method. LLM-based tools were used only to assist with writing, editing, and figure scripting, which does not require declaration under the policy.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.