ANCHORCLIP: ENHANCING CLIP ROBUSTNESS THROUGH INTERPRETABILITY

Anonymous authors
Paper under double-blind review

ABSTRACT

Pretrained Vision-language models such as CLIP show impressive zero-shot performance on clean images. However, they are extremely vulnerable to common image corruptions, such as Gaussian noise, Pixelation or Motion blur. Existing approaches to improve robustness focus mostly on adversarial fine tuning but they have crucial limitations. They are ineffective for severe image corruptions, involve fine tuning the entire CLIP encoder, and are largely black box. In this paper, we leverage CLIP's semantic interpretability to enforce robustness. We show that enforcing semantic consistency in concept space across corruptions significantly improves robustness. Our proposed method AnchorCLIP, uses a lightweight adapter to align the corrupted image embeddings to the interpretable latent space of clean CLIP embeddings, effectively "anchoring" the noisy representations to their semantic meanings. Extensive experiments indicate that AnchorCLIP outperforms existing adversarial robustness and interpretability based methods on widely used benchmark datasets, while maintaining performance on clean data. Distinct from previous work, AnchorCLIP is based on interpretability, is lightweight, and consistently outperforms strong baselines for severe image corruptions. By using interpretability as a mechanism for robustness instead of just post-hoc explanations, AnchorCLIP provides a scalable path toward vision-language systems that are both inherently robust and transparent.

1 INTRODUCTION

Vision-language foundation models, most notably CLIP (Radford et al., 2021), have achieved remarkable success in multi-modal representation learning, facilitating a wide array of downstream applications. These include zero-shot classification (Saha et al., 2024), open-vocabulary object detection (Minderer et al., 2024), video recognition (Lin et al., 2022), audio processing (Guzhov et al., 2022) and recently in text-guided image generation as a core component in models like DALL-E (Ramesh et al., 2021) and LLaVA (Liu et al., 2024). Due to this versatility, CLIP-based systems are increasingly integrated into safety-critical domains, including medical diagnosis (Kim et al., 2024; Liu et al., 2023) and autonomous driving (Xu et al., 2023). In these high-stakes applications, model reliability is paramount. However, several studies have shown that CLIP often exhibits unsatisfying interpretability (Li et al., 2022; Gong et al., 2025) and remains vulnerable to distribution shifts. In scenarios such as autonomous driving, where weather conditions or lens distortions can cause rapid shifts in image quality (Arnold et al., 2019), ensuring both robustness and transparency is critical for building calibrated trust.

Despite CLIP's unrivaled zero-shot performance on clean data, its accuracy degrades sharply when faced with real-world image corruptions. For instance, CLIP's zero-shot accuracy on CIFAR-10 drops from 94% to a mere 42% under Gaussian noise at severity level five. While existing literature has explored adversarial fine-tuning (Schlarmann et al., 2024; Zhang et al., 2025; Wang et al., 2024b) to bolster CLIP robustness, these methods are often optimized for norm-bounded attacks and may fail to generalize to unseen real-world corruptions (Chen et al., 2023).

Our objective is to bridge this gap by utilizing interpretability as a mechanism to enforce robustness against common image distortions like shot noise, gaussian noise, and pixelation. We introduce a new method, AnchorCLIP, that leverages the semantic interpretation of CLIP embeddings to guide the adaptation process, unlike traditional "black box" fine tuning. Recent work on CLIP

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

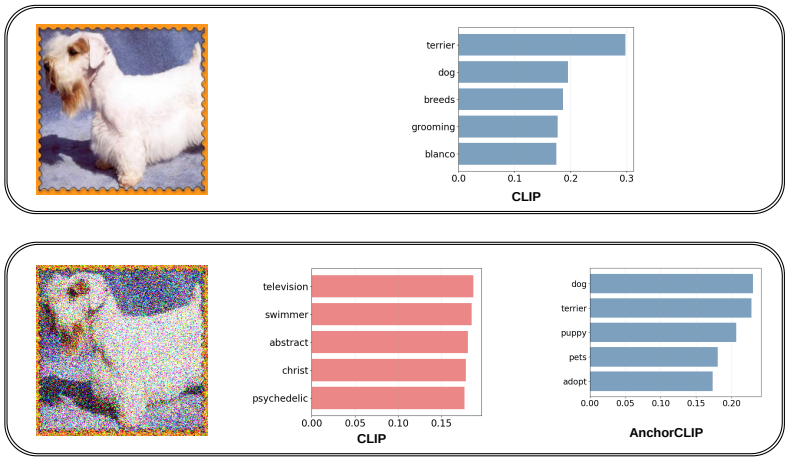


Figure 1: The top row shows a clean image, and the top 5 concepts with maximum cosine similarity with the embedding. The bottom row shows the corrupted version of the image, corrupted with Gaussian noise at severity 5. The original CLIP embedding is aligned with completely spurious concepts such as ”abstract” and ”psychedelic”. AnchorCLIP recovers dog related concepts which help in correctly classifying the image.

interpretability has proposed various methods for understanding its latent space, including label-free concept bottleneck models (Oikarinen et al., 2023), mechanistic head-based decomposition (Gandelsman et al., 2024a), and sparse linear concept embeddings such as SpLiCE (Bhalla et al., 2024). Many of these frameworks operate on the premise that CLIP embeddings can be decomposed into human-understandable concepts, with the residual vector encoding noise. While it is tempting to assume that such sparse representations inherently provide a degree of protection, we demonstrate that existing interpretable methods fail substantially even under mild corruption. By explicitly training a lightweight adapter to preserve the human interpretation of CLIP embeddings under noise, we provide a path toward a CLIP visual encoder that is both robust and inherently explainable.

Figure 1 shows an example, where we show the top five concepts with the largest cosine similarity with the CLIP visual embedding. We observe, in general, that while the CLIP embedding picks up concepts related to high frequency noise or textures (words such as ”abstract”, ”psychedelic”, ”texture”, ”glitter”), AnchorCLIP is able to recover the original meaning of the picture, (for example, emphasizing concepts corresponding to dogs in Figure 1). In Appendix D, we show more such examples.

2 Related Work

CLIP Robustness: While CLIP is known to be robust to natural distribution shifts in ImageNet (Radford et al., 2021), compared to models trained on ImageNet, it is well known that CLIP models are not robust against spurious features (Wang et al., 2024a), common noise patterns (Maharana et al., 2025), low level visual details (Wang et al., 2024c) or adversarial corruption (Schlarmann et al., 2024). Work on CLIP robustness has focused largely on building adversarial robustness of CLIP models, finetuning using supervised (Mao et al., 2023) and unsupervised (Schlarmann et al., 2024) loss functions. It has been observed that adversarial fine tuning improves CLIP's visual interpretability (Gong et al., 2025), as seen with salience maps. CLIPure (Zhang et al., 2025) suggests that purification in the latent space leads to better adversarial robustness. This is in line with our approach to use semantic interpretability to enforce robustness. COLA (Zhu et al., 2025) enforces adversarial robustness by projection of embeddings to a semantically defined subspace. However, unlike our work on common corruptions, these studies focus solely on adversarial robustness. Adversarial image embeddings are very close to the clean embeddings in latent space, and our results show that such adversarial methods (Mao et al., 2023; Schlarmann et al., 2024) do not work well under corruptions with large severity. Investigation into adapting CLIP for common corruptions

such as the ones defined in CIFAR10C, CIFAR100C and ImageNet-C are the most common in test time adaptation methods such as WATT (Osowiechi et al., 2024), TENT (Wang et al., 2020), ROTTA (Yuan et al., 2023), SAT (Mishra et al., 2024), RPL (Rusak et al., 2021), SAR (Niu et al., 2023) and BATCLIP (Maharana et al., 2025). However test-time adaptation adapts CLIP models to different corruption types and noises separately, with few shot unsupervised test examples. Unlike these methods, which require per-batch or per-domain optimization and costly backpropagation steps during inference, AnchorCLIP is trained once and performs inference with a single forward pass.

CLIP Interpretability: Recent work has focused on methods like SpLiCE (Bhalla et al., 2024) that interpret the CLIP image embedding as a sparse linear function of concept embeddings, or get concept scores (Hoang-Xuan et al.; Chattopadhyay et al., 2023; Yun et al., 2022) for related concepts. Mechanistic interpretability (Olah et al., 2017) approaches have interpreted CLIP's image embedding as a sum of contributions from each attention head (Gandelsman et al., 2024a), with its distinct semantic meaning. Concept Bottleneck Models(CBM)(Koh et al., 2020; Oikarinen et al., 2023; Panousis et al., 2023; Yuksekgonul et al., 2022) take the approach of creating a trainable hidden layer, whose neurons correspond to concepts. Attempts (Gandelsman et al., 2024b; Oikarinen & Weng, 2022) have also been made to interpret individual neurons in CLIP. However, evaluation of these models are limited to the case of clean images. As we discuss later, interpretable approaches do not neatly separate the semantics from the noise, and hence breakdown when faced with noisy images.

CLIP Adapters: While many of the adversarial robustness approaches rely on finetuning CLIP (Mao et al., 2023; Schlarmann et al., 2024), CLIP adapters have been used as a lightweight plug and play approach to improve CLIP models. CoOp (Zhou et al., 2022b), CoCoOp (Zhou et al., 2022a), KgCoOp (Yao et al., 2023) and ProGrad (Zhu et al., 2023) adapts CLIP by learning a prompt from few shot examples. CLIP-Adapter (Gao et al., 2024) uses a small learnable bottleneck linear layer with residual connections to adapt the CLIP embedding output. CLIP adapter's main application is in few shot image classification tasks. Similar few shot approaches include the training free TIP-Adapter (Zhang et al., 2022) and LP++(Huang et al., 2024), a strong linear probe. CLIP-CEIL (Yu et al., 2024) uses a lightweight adapter with transformer layers for domain generalization. Our adapter architecture is the closest to CLIPCEIL (Yu et al., 2024) in making use of intermediate transformer outputs.

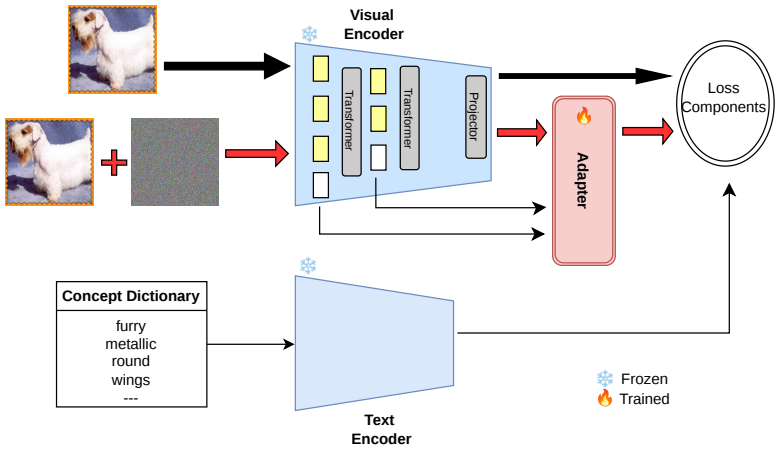


Figure 2: Overview of AnchorCLIP. The frozen CLIP image encoder processes both the clean image and its synthetically corrupted version during training. The corrupted image embedding is passed through an adapter. The adapter weights are trained using a combination of interpretability-based losses. Losses are calculated based on the adapter's output embedding (red arrow), and the embedding from the clean image (black arrow). The adapter takes the final embeddings, as well as outputs from intermediate transformer layers, as its inputs

3 METHOD

3.1 PROBLEM SETUP

Zero-Shot Classification: In this section, we look at how CLIP is used as a zero-shot classifier. CLIP consists of an image encoder $\phi : I \to \mathbb{R}^d$ and a text encoder $\psi : T \to \mathbb{R}^d$ which map an image and a text into a shared embedding space. For zero-shot classification, of an image **x** into **K** classes, we use the image encoder to get the image embedding $f_x^{\text{final}} = \phi(x)$. Then we create a text prompt for each class t_k = "A photo of <CLS>", find the embedding for each text prompt $z^k = \psi(t_k)$ and then we predict the correct class as the one whose prompt has the largest cosine similarity with the image embedding:

$$\arg\max_{k=1,\ldots,K} \cos(f_x^{\text{final}}, z^k).$$

When using corrupted images, the text encoder remains the same. Only the image encoder now encodes a corrupted image x_{corr} instead of x.

3.2 ANCHORCLIP

Figure 2 shows an overview of AnchorCLIP. AnchorCLIP passes a noisy image through the CLIP visual encoder, and then through an adapter, which helps recover the interpretable parts of the CLIP embedding. The main components of AnchorCLIP are the adapter architecture, and the novel loss components that depend on interpretability. Below we describe both these aspects in detail.

CLIP Image Encoder: While CLIP can be used with different image encoders, in this work, we focus on the variants that use ViT (Dosovitskiy et al., 2020) as the backbone. The ViT image encoder consists of L Transformer layers, each utilizing residual connections and containing a self-attention mechanism followed by an MLP block. This is followed by a final projection layer. The input image is divided into rectangular patches, and in each layer we have an image embedding corresponding to each patch. Additionally, there is a learned embedding $f_x^l \in \mathbb{R}^D$ which is called the class token. Thus, to be more precise, we can write the output of the image encoder as not just the final token, but as all the class tokens and the final token:

$$\{f_{\mathbf{x}}^l\}_{l=1}^L, f_{\mathbf{x}}^{\text{final}} = \phi(x) \tag{1}$$

In this paper, we use the OpenCLIP ViT-B/32 architecture with $L = 12$ transformer layers, the dimensionality of the joint embedding space is $d = 512$ and the dimensionality of the class tokens is $D = 768$.

Adapter: In this work, we aim to use a lightweight adapter to recover a meaningful embedding representation for severely corrupted images. Our adapter uses not only the final CLIP embedding, but also the class tokens from the intermediate layers as inputs. Figure 2 shows the adapter architecture. The class tokens $f_{\mathbf{x}}^l\}_{l=1}^L$ are used as inputs to a transformer layer, we use the last token of the attention layer as output. Then we concatenate the final embedding f_x^{final} with the last output token, and pass this through a two layer MLP, where the hidden layer is expanded. Finally we use a residual connection to do a weighted interpolation between the adapter output and the final embedding. Thus:

$$f_{\text{adapt}} = \text{MLP}([\text{Trans}(\{f_{\mathbf{x}}^l\}_{l=1}^L)_{-1} \parallel f_{\mathbf{x}}^{\text{final}}]) \tag{2}$$

where $\text{Trans}(\cdot)_{-1}$ denotes the last token of the transformer output sequence and $\parallel$ represents the concatenation operation.

$$f_{\mathbf{x}}^{\text{out}} = A_\theta(\{f_{\mathbf{x}}^l\}_{l=1}^L, f_{\mathbf{x}}^{\text{final}}) = \alpha \cdot f_{\text{adapt}} + (1 - \alpha) \cdot f_{\mathbf{x}}^{\text{final}} \tag{3}$$

The Ablation section contains a discussion detailing our motivation of choosing this architecture.

3.3 LOSS FUNCTIONS

Concept Dictionary: Interpretable CLIP models (Bhalla et al., 2024; Hoang-Xuan et al.; Gandelsman et al., 2024a; Oikarinen et al., 2023) use "concepts", which are one word or two word phrases in the english language, as atoms or semantic units, to explain the latent space. The affinity of

embeddings to these concepts define their human readable interpretation. We take the same approach, representing the latent space using a dictionary $D = \{c_1, c_2, \ldots, c_C\}$ of C concepts. We follow SpLiCE (Bhalla et al., 2024) which creates a set of 15000 unique concepts, comprising of 10,000 most frequent single-word and 5,000 most frequent two-word bigrams extracted from the LAION-400m (Schuhmann et al., 2021) dataset captions.

Embedding Standardization: Since it is known (Bhalla et al., 2024) that image and text embeddings are not aligned in the embedding space, we standardize these embeddings. We define $\sigma(\mathbf{x}) = \mathbf{x}/\|\mathbf{x}\|_2$ as the ℓ_2-normalization operation. For any embedding f, its standardized counterpart $\hat{f}$ is given by:

$$\hat{f} = \Phi(f) = \sigma\left(\sigma(f) - \mu_{\text{img}}\right) \tag{4}$$

where μ_{img} is the pre-computed image mean of all embeddings from MS-COCO. A similar treatment applies to text embeddings. This standardization technique was originally applied in SpLiCE (Bhalla et al., 2024).

Concept Loss: Our fundamental hypothesis in designing our losses is that robustness can be enforced through interpretability. Intuitively, the concept loss enforces that corrupted images should activate the same semantic concepts as clean images. Building upon the idea that affinity to embeddings are described by cosine similarities with concepts, we treat these similarity scores as logits for a discrete probability distribution, over the concept dictionary. Given a clean image $\mathbf{x}$, its CLIP image embedding is $f_{\mathbf{x}}^{\text{final}}$. We assign the probability mass at each concept each $c_i \in \mathbb{R}^d$ as:

$$l_i^{\text{clean}} = \frac{\cos(\hat{f}_{\mathbf{x}}^{\text{final}}, \hat{c}_i)}{\tau} \tag{5}$$

Transforming these logits, using a softmax function over the entire dictionary $\mathcal{D}$, we define the discrete probability distribution P:

$$P_i = \frac{\exp(l_i^{\text{clean}})}{\sum_{j=1}^{C} \exp(l_j^{\text{clean}})} \tag{6}$$

Equivalently, for a corrupted image $\mathbf{x}_{\text{corr}}$, $f_{\mathbf{x}_{\text{corr}}}^{\text{out}}$ is the embedding produced after passing the corrupted image through the clip encoder and then through the adapter. We derive similar logits and a distribution:

$$Q_i = \frac{\exp(l_i^{\text{corr}})}{\sum_{j=1}^{C} \exp(l_j^{\text{corr}})} \tag{7}$$

Finally, we define the concept loss as the KL divergence between these two discrete distributions:

$$\mathcal{L}_{\text{concept}} = D_{KL}(P\|Q) = \sum_{i=1}^{C} P_i \log\left(\frac{P_i}{Q_i}\right) \tag{8}$$

Semantic Loss: To further enforce interpretability, the semantic loss anchors corrupted embeddings to the interpretable reconstruction of the clean embedding. For this, we first identify the top K concepts in the concept dictionary whose cosine similarity is the largest with the clean image embedding. We then use linear regression (OLS) to find the best linear combination of these concept embeddings which reconstructs the clean image embedding. The method is similar to, and motivated by SpLiCE (Bhalla et al., 2024) which reconstructs the clean image embedding as a sparse linear combination of concept embeddings. However, our OLS reconstruction using top K concepts is much cheaper to compute, and hence scalable, compared to SpLiCE, which uses an iterative Lasso solver.

Forcing the corrupted image embedding to be close to this reconstruction enables it to only learn the interpretable parts of the image, ignoring the noise. Given the standardized clean embedding, we use OLS to find a set of weights $\mathbf{w}^*$ that reconstructs the original embedding. $\mathbf{w}^*$ is sparse by construction, since only elements corresponding to the top K concepts are non zero.

Given this set of weights, the reconstructed interpretable part of the clean image embedding looks like

$$f_{\text{recon}} = \sigma(\mathbf{C}\mathbf{w}^*) \tag{9}$$

Table 1: Zero-shot classification accuracy under image corruption for CIFAR-10C and ImageNet-C (severity=5)

Dataset	Corruption Type	CLIP	Linear Adapter	TeCoA	FARE	AnchorCLIP (Ours)
CIFAR-10C	Gaussian Noise	0.4172	0.4840	**0.5167**	0.4297	0.5096
	Shot Noise	0.4659	0.5230	0.5297	0.4337	**0.5419**
	Impulse Noise	0.4629	0.4670	0.3333	0.2980	**0.5537**
	Defocus Blur	0.7762	0.8040	0.3179	0.3627	**0.8492**
	Glass Blur	0.4498	0.5420	0.3896	0.3459	**0.6040**
	Motion Blur	0.7161	0.7860	0.3323	0.3853	**0.8264**
	Zoom Blur	0.7770	0.8240	0.3535	0.3903	**0.8585**
	Snow	0.8085	0.8330	0.5164	0.5302	**0.8458**
	Frost	0.8038	0.8290	0.5240	0.5416	**0.8376**
	Fog	0.7906	0.8210	0.1571	0.1689	**0.8430**
	Brightness	0.8991	0.9120	0.6310	0.6930	**0.9227**
	Contrast	0.7108	0.7670	0.1795	0.1574	**0.8159**
	Elastic Trans.	0.6468	0.7110	0.3665	0.3741	**0.7592**
	Pixelate	0.4995	0.5310	0.5517	**0.5644**	0.5368
	JPEG Comp.	0.6344	0.6580	0.5336	0.5601	**0.6860**
	Mean Accuracy	0.6572	0.6994	0.4155	0.4157	**0.7327**
ImageNet-C	Gaussian Noise	0.0732	0.0875	0.0517	0.0590	**0.1585**
	Shot Noise	0.0816	0.1007	0.0709	0.0744	**0.1742**
	Impulse Noise	0.0747	0.0854	0.0488	0.0577	**0.1602**
	Defocus Blur	0.2116	0.2622	0.0638	0.0549	**0.3145**
	Glass Blur	0.0869	0.1053	0.1163	0.0978	**0.1463**
	Motion Blur	0.1436	0.1833	0.0922	0.0677	**0.2633**
	Zoom Blur	0.1328	0.1752	0.1299	0.0912	**0.2254**
	Snow	0.1886	0.2472	0.1019	0.0929	**0.2759**
	Frost	0.1968	0.2524	0.0926	0.0866	**0.2741**
	Fog	0.2467	0.3190	0.0040	0.0025	**0.3485**
	Brightness	0.3992	0.5275	0.2417	0.2441	**0.5379**
	Contrast	0.1435	0.1723	0.0029	0.0022	**0.2379**
	Elastic Trans.	0.1422	0.1751	**0.2814**	0.2031	0.1976
	Pixelate	0.2614	0.3245	0.3237	0.2421	**0.3807**
	JPEG Comp.	0.2583	0.3263	**0.3972**	0.3395	0.3661
	Mean Accuracy	0.1761	0.2229	0.1346	0.1144	**0.2707**

Thus the semantic loss is defined as:

$$\mathcal{L}_{\text{sem}} = 1 - \langle f_{\text{recon}}, \hat{f}^{\text{out}}_{\mathbf{x}_{\text{corr}}} \rangle$$

In summary, in order to use interpretability to enforce robustness, our loss function enforces invariance in the concept distribution space (Concept loss) and semantic subspace (semantic loss). Based on our ablation studies we define the final loss as

$$\mathcal{L}_{\text{AnchorCLIP}} = 0.7\mathcal{L}_{\text{sem}} + 0.3\mathcal{L}_{\text{concept}} \tag{10}$$

3.4 Training

During Training, we only use the clean dataset. For each image, we randomly select one among the available synthetic transformations: Gaussian, Shot, Impulse, Contrast, Defocus Blur, Motion Blur, Brightness and Pixelate. This is only a small subset of the fifteen noises we test on. We also randomly choose a severity between 1 and 5. We then pass the clean image through the CLIP visual encoder, and the synthetically corrupted image through the visual encoder and adapter, and calculate the losses. We use a small pre-training phase, to calculate the mean value of the losses, in order to scale them. The CLIP encoder is frozen throughout the process, and we only train the adapter A_θ.

Table 2: Zero-shot classification accuracy for CIFAR-100C under image corruption (severity=5)

Dataset	Corruption Type	CLIP	Linear Adapter	TeCoA	FARE	AnchorCLIP (Ours)
CIFAR-100C	Gaussian Noise	0.1805	0.2240	**0.2566**	0.2174	0.2521
	Shot Noise	0.2026	0.2490	0.2563	0.2266	**0.2731**
	Impulse Noise	0.2018	0.2260	0.1087	0.1106	**0.2573**
	Defocus Blur	0.5145	0.5320	0.1880	0.1859	**0.5635**
	Glass Blur	0.1831	0.2160	0.2290	**0.2533**	0.2452
	Motion Blur	0.4786	0.5000	0.1739	0.1781	**0.5374**
	Zoom Blur	0.5448	0.5570	0.2172	0.2143	**0.5902**
	Snow	0.5427	0.5640	0.3063	0.3327	**0.5763**
	Frost	0.5249	0.5500	0.2704	0.2888	**0.5542**
	Fog	0.4748	0.5070	0.0521	0.0544	**0.5216**
	Brightness	0.6363	0.6500	0.3395	0.3728	**0.6750**
	Contrast	0.3913	0.4250	0.0206	0.0182	**0.4795**
	Elastic Trans.	0.3328	0.3870	0.2215	0.2407	**0.4017**
	Pixelate	0.2193	0.2540	**0.3325**	0.3237	0.2921
	JPEG Comp.	0.3590	0.3790	0.3337	0.3486	**0.4044**
	Mean Accuracy	0.3858	0.4146	0.2204	0.2244	**0.4416**

4 EXPERIMENTS

4.1 SETUP

Datasets: For all the experiments in the paper, we use the CIFAR10, CIFAR100 (Krizhevsky et al., 2009) and ImageNet (Deng et al., 2009) datasets as a source of clean images. The CIFAR-10C, CIFAR-100C and ImageNet-C (Hendrycks & Dietterich, 2019), which contain common corruptions of the corresponding clean datasets are used to test our algorithm against noisy images. These datasets include 15 common corruptions, such as Gaussian noise, shot noise, snow, pixelation etc. at five different levels of severities, increasing in level of degradation from 1 to 5. These datasets are part of the Robustbench benchmark (Croce et al., 2020) and are widely used in the robustness literature, including studying test time adaptation of CLIP models (Maharana et al., 2025). See Appendix A for more information on all the corruption types and visual examples.

Models: We use the OpenCLIP ViT-B/32 model (Ilharco et al., 2021) for all the experiments in our paper. We also show results for OpenCLIP ViT-B/16 in Appendix C.3.

4.2 ZERO-SHOT CLASSIFICATION

Tables 1 and 2 report zero-shot classification accuracy on CIFAR-10C, CIFAR-100C, and ImageNet-C. We compare our results to the original CLIP model, as well as multiple strong baselines, which include models designed for adversarial robustness, interpretability and task specific training protocol. We do not compare against test-time adaptation(TTA) methods such as BATCLIP (Maharana et al., 2025) as these methods operate under a fundamentally different and more favorable setting: they adapt model parameters at inference time using examples from each corruption type separately. In general, AnchorCLIP outperforms CLIP and all baselines across most corruption types. In some cases, like ImageNet-C Gaussian noise the improvement is dramatic ($\sim 2x$).

Adversarial Methods: We use two state of the art adversarial methods as baseline, TeCOA (Mao et al., 2023) and FARE (Schlarmann et al., 2024). The TeCoA and FARE models used are pretrained for adversarial robustness on ImageNet and released by the authors.TeCoA has the further advantage that, unlike AnchorCLIP, it is supervised. FARE is unsupervised.

While adversarial methods work well on the adversarial attacks, we observe they do not improve the accuracy in cases where the image is corrupted by noise of large magnitudes. TeCOA (Mao et al., 2023) performs well on Gaussian noise, but is not competitive across noises. AnchorCLIP on the other hand, performs well on all the 15 corruption types, across datasets.

Interpretable Methods: In Appendix C.4 (Table 11 and Table 12), we show two additional baselines for interpretability based methods SpLiCE (Bhalla et al., 2024) and COLA (Zhu et al., 2025). While SpLiCE (Bhalla et al., 2024) demonstrates that the CLIP image embedding is represented by an addition of underlying semantic concepts and COLA (Zhu et al., 2025) works by projecting the embeddings to an orthogonal subspace defined by the concept embeddings. AnchorCLIP beats both these baselines, indicating that the noise does not separate linearly from the interpretable part, nor does it live neatly in a subspace orthogonal to the interpretable part of the embeddings. This is our insight into why interpretable methods may not work in the presence of noise as a plug and play method.

Task Specific Baseline: We use a Linear Adapter trained with a standard MSE loss (see details in Appendix B.2), using the exact same training protocol (subset of noises and randomly chosen severities) as a strong task specific baseline. This helps us establish that the gains we see in AnchorCLIP are not simply due to training on a subset of common corruptions, and on the same dataset. We observe that interpretability based AnchorCLIP shows consistent and substantial gains over the Linear Adapter, thus showing the AnchorCLIP's gains are indeed due to its interpretability based losses.

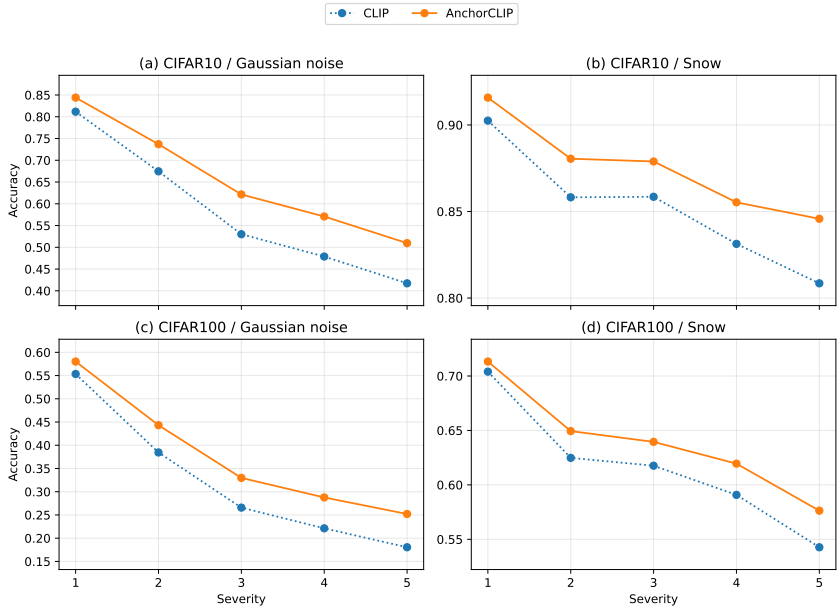


Figure 3: CLIP and AnchorCLIP accuracies for Gaussian and Snow noises, across a range of severities, from 1 to 5.

Generalization Capabilities: In multiple ways, AnchorCLIP shows that it can generalize beyond what it has seen in training. While AnchorCLIP is trained with only a subset of noise transforms, it improves performance on the set of noises it has never seen in training, namely zoom blur, snow, fog, elastic tranformation, JPEG compression, glass blur and frost in the majority of cases. Secondly, Table 4 shows that an AnchorCLIP model trained on a given dataset is competitive on another smaller dataset. Additionally, in Table 3, we show that AnchorCLIP trained on ImageNet is competitive across a range of datasets. This includes Caltech101 (Fei-Fei et al., 2006) and OxfordPets (Parkhi et al., 2012) for coarse and fine grained classification, and DTD (Cimpoi et al., 2014) for texture recognition. In the Appendix C.2, we show the results for each noise, for these datasets.

Additionally, we note that AnchorCLIP's performance is not limited to the maximum noise severity. It shows improvements over CLIP over the whole range of severities as shown in Figure 3.

Tradeoff on Clean Images: It is well known that adversarially robust methods often tradeoff robust accuracy for clean accuracy. For instance both TeCOA and FARE underperform CLIP (except TeCOA on ImageNet on which it is supervised)(Schlarmann et al., 2024). AnchorCLIP, despite being unsupervised, actually improves clean accuracy, in CIFAR10 and CIFAR100, and only shows a

Table 3: Zero-shot generalization to 3 held-out datasets, mean accuracy over 14 corruptions (severity=5). AnchorCLIP is trained on ImageNet only, transferred zero-shot.

Dataset	CLIP	TeCoA	FARE	AnchorCLIP (Ours)
Caltech101	0.6746	0.4674	0.4951	**0.6911**
DTD	0.2308	0.1376	0.1585	**0.2582**
OxfordPets	0.4486	0.3359	0.2595	**0.4542**

Table 4: Cross-dataset zero-shot transfer, mean accuracy over 15 corruptions (severity=5).

Train Dataset	Test Dataset	CLIP	TeCoA	FARE	AnchorCLIP (Ours)
ImageNet	CIFAR100	0.3858	0.2204	0.2244	**0.4007**
CIFAR100	CIFAR10	0.6572	0.4155	0.4157	**0.6653**

small drop in ImageNet. Table 5 shows the clean accuracies for CLIP and AnchorCLIP for the three datasets.

Table 5: Comparison of clean accuracies: CLIP vs. AnchorCLIP

Dataset	CLIP	AnchorCLIP
CIFAR10	0.9366	**0.9482**
CIFAR100	0.7569	**0.7613**
ImageNet	**0.6631**	0.6555

4.3 Ablation Studies

We perform ablation experiments to understand the effectiveness of important design decisions we made in AnchorCLIP, including the interpretable loss components, the adapter architecture, and the contribution of the CLIP encoder transformer layers. Due to space constraints, we present ablation results on loss components, adapter architecture, transformer layer inputs and interpolation ratio in Appendix C.5.

Adapter Architecture: Although the most commonly used Adapter for CLIP is the linear CLIP Adapter (Gao et al., 2024), our results show that architectures that use intermediate attention layers outperform linear adapters. Furthermore, we observe that the class tokens from the late layers are much more effective than those from the early transformer layers. These ablation studies motivate our design decision for the adapter architecture. Our observation on late transformer layers are in line with observations in previous interpretable CLIP decompositions (Gandelsman et al., 2024a) (see Figure 2 of this paper) which shows that only the last self attention layers have a meaningful effect on the zero-shot accuracy. While this previous observation was for clean images, our ablation study shows that for images severely corrupted with noise, the late transformer layers contain the interpretable semantic information, which is relevant to its zero-shot accuracy.

5 Discussion

Broader Impact: We see the broader impact of this work in safety critical applications such as autonomous driving. In these applications, it is critical that CLIP embeddings are meaningful under real world noises, and the interpretability of AnchorCLIP makes it easier to custom train it with use case specific concept dictionaries. Also, post hoc analysis can help understand failure modes in such real world situations, since AnchorCLIP is not a Black Box model.

Limitations: This work focuses on common corruptions, encoded in the CIFAR10-C, CIFAR100-C and ImageNet-C datasets. In future, we can also look at robustness against adversarial attacks, and domain shifts, through a similar interpretability driven approach. Further, we use an over complete dictionary of concepts, and future work might explore the effect of use case specific dictionaries.

AI USE STATEMENT

In this work, we used generative AI tools to clean and reformat dataset and implement methods. We have not used generative AI tools to propose or refine hypotheses, design or provide feedback on research methodology or experiments, support qualitative and thematic data analysis or assist with translation. The following use cases, in particular, generating synthetic data sets, help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs are not applicable to this work. Additionally, we used generative AI tools to create or modify scientific figures or images and create or edit software code. We have reviewed all AI-assisted work. AI generated code was limited to code completions or small functions. We have verified with test cases, and checked that results agree for base cases with data in literature. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

The paper does not use studies using any human subjects or release any new datasets. The experiments are conducted using publicly available data.

REPRODUCIBILITY STATEMENT

We have included the full implementation details for AnchorCLIP in Appendix B.1 and baseline methods in Appendix B.2. We have also included details on how to download the datasets used in the paper in Appendix A. We have also shown that the results are repeatable in Appendix C.1 and quantified the statistical errors. We will release the source code with the final version of the paper.

REFERENCES

Eduardo Arnold, Omar Y Al-Jarrah, Mehrdad Dianati, Saber Fallah, David Oxtoby, and Alex Mouzakitis. A survey on 3d object detection methods for autonomous driving applications. *IEEE Transactions on Intelligent Transportation Systems*, 20(10):3782–3795, 2019.

Usha Bhalla, Alex Oesterling, Suraj Srinivas, Flavio P Calmon, and Himabindu Lakkaraju. Interpreting CLIP with sparse linear concept embeddings (SpLiCE). In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Aditya Chattopadhyay, Ryan Pilgrim, and Rene Vidal. Information maximization perspective of orthogonal matching pursuit with applications to explainable ai. *Advances in Neural Information Processing Systems*, 36:2956–2990, 2023.

Chen Chen, Bowen Zhang, Liangliang Cao, Jiguang Shen, Tom Gunter, Albin Madappally Jose, Alexander Toshev, Jonathon Shlens, Ruoming Pang, and Yinfei Yang. STAIR: Learning sparse text and image representation in grounded tokens. *arXiv preprint arXiv:2301.13081*, 2023.

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*, pp. 3606–3613, 2014.

Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: a standardized adversarial robustness benchmark. *arXiv preprint arXiv:2010.09670*, 2020.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

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

Li Fei-Fei, Robert Fergus, and Pietro Perona. One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence*, 28(4):594–611, 2006.

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

Yossi Gandelsman, Alexei A Efros, and Jacob Steinhardt. Interpreting the second-order effects of neurons in clip. *arXiv preprint arXiv:2406.04341*, 2024b.

Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. *International journal of computer vision*, 132(2):581–595, 2024.

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.

Andrey Guzhov, Federico Raue, Jörn Hees, and Andreas Dengel. Audioclip: Extending clip to image, text and audio. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 976–980. IEEE, 2022.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

Nhat Hoang-Xuan, Xiyuan Wei, Wanli Xing, Tianbao Yang, and My T Thai. Advancing interpretability of clip representations with concept surrogate model. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Yunshi Huang, Fereshteh Shakeri, Jose Dolz, Malik Boudiaf, Houda Bahig, and Ismail Ben Ayed. Lp++: A surprisingly strong linear probe for few-shot clip. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23773–23782, 2024.

Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, et al. Openclip [software], 2021.

Chanwoo Kim, Soham U Gadgil, Alex J DeGrave, Jesutofunmi A Omiye, Zhuo Ran Cai, Roxana Daneshjou, and Su-In Lee. Transparent medical image AI via an image-text foundation model grounded in medical literature. *Nature Medicine*, pp. 1–12, 2024.

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*, pp. 5338–5348. PMLR, 2020.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Yi Li, Hualiang Wang, Yiqun Duan, Hang Xu, and Xiaomeng Li. Exploring visual interpretability for contrastive language-image pre-training. *arXiv preprint arXiv:2209.07046*, 2022.

Ziyi Lin, Shijie Geng, Renrui Zhang, Peng Gao, Gerard De Melo, Xiaogang Wang, Jifeng Dai, Yu Qiao, and Hongsheng Li. Frozen clip models are efficient video learners. In *European Conference on Computer Vision*, pp. 388–404. Springer, 2022.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2024.

Jie Liu, Yixiao Zhang, Jie-Neng Chen, Junfei Xiao, Yongyi Lu, Bennett A Landman, Yixuan Yuan, Alan Yuille, Yucheng Tang, and Zongwei Zhou. Clip-driven universal model for organ segmentation and tumor detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 21152–21164, 2023.

Sarthak Kumar Maharana, Baoming Zhang, Leonid Karlinsky, Rogerio Feris, and Yunhui Guo. BATCLIP: Bimodal online test-time adaptation for CLIP. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

Chengzhi Mao, Scott Geng, Junfeng Yang, Xin Wang, and Carl Vondrick. Understanding zero-shot adversarial robustness for large-scale models. In *International Conference on Learning Representations (ICLR)*, 2023.

Matthias Minderer, Alexey Gritsenko, and Neil Houlsby. Scaling open-vocabulary object detection. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2024.

Shambhavi Mishra, Julio Silva-Rodriguez, Ismail Ben Ayed, Marco Pedersoli, and Jose Dolz. Semantic anchor transport: Robust test-time adaptation for vision-language models. *arXiv preprint arXiv:2411.17002*, 2024.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Zhiquan Wen, Yaofo Chen, Peilin Zhao, and Mingkui Tan. Towards stable test-time adaptation in dynamic wild world. *arXiv preprint arXiv:2302.12400*, 2023.

Tuomas Oikarinen and Tsui-Wei Weng. Clip-dissect: Automatic description of neuron representations in deep vision networks. *arXiv preprint arXiv:2204.10965*, 2022.

Tuomas Oikarinen, Subhro Das, Lam M Nguyen, and Tsui-Wei Weng. Label-free concept bottleneck models. In *International Conference on Learning Representations (ICLR)*, 2023.

Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2(11):e7, 2017.

David Osowiechi, Mehrdad Noori, Gustavo A Hakim, Moslem Yazdanpanah, Ali Bahri, Milad Cheraghalikhani, Sahar Dastani, Farzad Beizaee, Ismail B Ayed, and Christian Desrosiers. Watt: Weight average test time adaptation of clip. *Advances in neural information processing systems*, 37:48015–48044, 2024.

Konstantinos Panagiotis Panousis, Dino Ienco, and Diego Marcos. Sparse linear concept discovery models. In *Proceedings of the ieee/cvf international conference on computer vision*, pp. 2767–2771, 2023.

Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pp. 3498–3505. IEEE, 2012.

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*, pp. 8748–8763. PmLR, 2021.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning (ICML)*, 2021.

Evgenia Rusak, Steffen Schneider, George Pachitariu, Luisa Eck, Peter Gehler, Oliver Bringmann, Wieland Brendel, and Matthias Bethge. If your data distribution shifts, use self-learning. *arXiv preprint arXiv:2104.12928*, 2021.

Oindrila Saha, Grant Van Horn, and Subhransu Maji. Improved zero-shot classification by adapting VLMs with text descriptions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Christian Schlarmann, Naman Deep Singh, Francesco Croce, and Matthias Hein. Robust CLIP: Unsupervised adversarial fine-tuning of vision embeddings for robust large vision-language models. In *International Conference on Machine Learning (ICML)*, 2024. Oral Presentation.

Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*, 2020.

Qizhou Wang, Yong Lin, Yongqiang Chen, Ludwig Schmidt, Bo Han, and Tong Zhang. A sober look at the robustness of clips to spurious features. *Advances in Neural Information Processing Systems*, 37:122484–122523, 2024a.

Sibo Wang, Jie Zhang, Zheng Yuan, and Shiguang Shan. Pre-trained model guided fine-tuning for zero-shot adversarial robustness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024b.

Wenxuan Wang, Quan Sun, Fan Zhang, Yepeng Tang, Jing Liu, and Xinlong Wang. Diffusion feedback helps clip see better. *arXiv preprint arXiv:2407.20171*, 2024c.

Zhenhua Xu, Yujia Zhang, Enze Xie, Zhen Zhao, Yong Guo, Kenneth KY Wong, Zhenguo Li, and Hengshuang Zhao. DriveGPT4: Interpretable end-to-end autonomous driving via large language model. *arXiv preprint arXiv:2310.01412*, 2023.

Hantao Yao, Rui Zhang, and Changsheng Xu. Visual-language prompt tuning with knowledge-guided context optimization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6757–6767, 2023.

Xi Yu, Shinjae Yoo, and Yuewei Lin. CLIPCEIL: Domain generalization through CLIP via channel refinement and image-text alignment. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Longhui Yuan, Binhui Xie, and Shuang Li. Robust test-time adaptation in dynamic scenarios. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15922–15932, 2023.

Mert Yuksekgonul, Maggie Wang, and James Zou. Post-hoc concept bottleneck models. *arXiv preprint arXiv:2205.15480*, 2022.

Tian Yun, Usha Bhalla, Ellie Pavlick, and Chen Sun. Do vision-language pretrained models learn composable primitive concepts? *arXiv preprint arXiv:2203.17271*, 2022.

Mingkun Zhang, Keping Bi, Wei Chen, Jiafeng Guo, and Xueqi Cheng. CLIPure: Purification in latent space via CLIP for adversarially robust zero-shot classification. In *International Conference on Learning Representations (ICLR)*, 2025.

Renrui Zhang, Wei Zhang, Rongyao Fang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free adaption of clip for few-shot classification. In *European conference on computer vision*, pp. 493–510. Springer, 2022.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16816–16825, 2022a.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International journal of computer vision*, 130(9):2337–2348, 2022b.

Beier Zhu, Yulei Niu, Yucheng Han, Yue Wu, and Hanwang Zhang. Prompt-aligned gradient for prompt tuning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 15659–15669, 2023.

Xingyu Zhu, Beier Zhu, Shuo Wang, Kesen Zhao, and Hanwang Zhang. Enhancing CLIP robustness via cross-modality alignment. In *Advances in Neural Information Processing Systems*, volume 38, 2025. Spotlight.

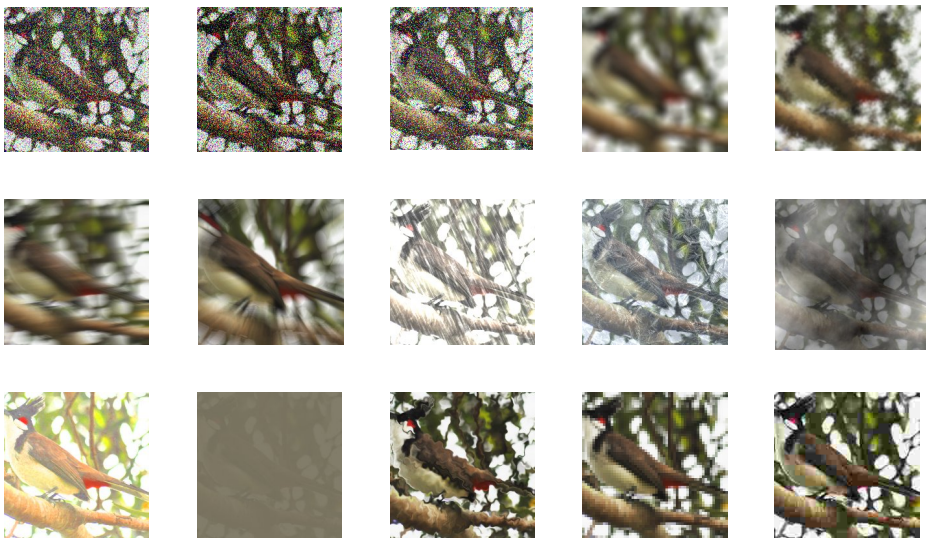


Figure 4: The figure shows 15 different noises, at severity 5, for the same image in ImageNet-C. The noises are Gaussian noise, Shot noise, Impulse noise, Defocus blur, Glass blur (top row), Motion blur, Zoom blur, Snow, Frost, Fog (middle row) and Brightness, Contrast, Elastic Transform, Pixelate and JPEG Compression (bottom row) from left to right.

A Datasets

A.1 Corrupt Datasets

To evaluate the robustness of our models, we utilize the CIFAR-10C, CIFAR-100C, and ImageNet-C benchmarks, originally established by Hendrycks and Dietterich (Hendrycks & Dietterich, 2019). These datasets serve as the standard for testing model performance under distribution shift and real-world image degradation. Each benchmark is constructed by applying 15 unique types of algorithmic corruptions to the original test sets of CIFAR-10, CIFAR-100 (Krizhevsky et al., 2009), and ImageNet (Deng et al., 2009).



Figure 5: The figure shows the same image from ImageNet-C, with Gaussian noise applied at 5 severities, from 1 to 5.

- **Severity Levels:** Each corruption type is implemented across five discrete levels of intensity. These range from *Level 1* (subtle degradation) to *Level 5* (extreme degradation), allowing for a granular analysis of model performance as image quality decreases.
- **Sample Size:** The number of test samples remains consistent across all 15 corruption tasks for each base dataset. The distribution is summarized in Table 6.

Table 7 shows the 15 corruption or noise types, and a rough category these noises can be grouped into. Figure 4 visualizes the fifteen noises applied at severity 5, on the same image in ImageNet-C. Figure 5 shows the same image at five different severities, ranging from 1 to 5. These figures provide us with a visual understanding of the distortions.

Table 6: Number of test samples per corruption type and severity combination.

Dataset	Samples per Noise/Severity
CIFAR-10C	10,000
CIFAR-100C	10,000
ImageNet-C	50,000

Table 7: Taxonomy of image corruptions categorized by type and description.

Category	Corruption Types	Description
Noise	Gaussian, Shot, Impulse	Simulates sensor interference and random pixel-level fluctuations.
Blur	Defocus, Glass, Motion, Zoom	Replicates optical distortions and camera movement artifacts.
Weather	Snow, Frost, Fog	Mimics atmospheric conditions that obscure visual clarity and contrast.
Digital	Brightness, Contrast, Elastic, Pixelate, JPEG	Represents post-processing artifacts and hardware-induced distortions.

The corrupted benchmarks can be accessed at their respective Zenodo repositories: CIFAR-10C (`https://zenodo.org/records/2535967`), CIFAR-100C (`https://zenodo.org/records/3555552`), and ImageNet-C (`https://zenodo.org/records/2235448`).

A.2 Additional Datasets

For the zero-shot generalization results in Table 3, we additionally evaluate on Caltech101 (Fei-Fei et al., 2006), DTD (Cimpoi et al., 2014), and OxfordPets (Parkhi et al., 2012). Unlike CIFAR and ImageNet, these datasets do not have a publicly released "-C" corrupted benchmark, so we construct one ourselves as follows.

Base data and splits. We download the clean test images automatically via `torchvision.datasets` (`Caltech101`, `DTD`, `OxfordIIITPet`), and use the train/val/test splits for consistent comparison with prior zero-shot transfer work on these datasets. Table 3 reports results on the test split only (although we never trained on this dataset, this splitting is just for making sure a future comparison which trains on this dataset is fair)

Corruption protocol. We use the corruption function source code from Hendrycks and Dietterich (Hendrycks & Dietterich, 2019) directly to each test image at inference time, at severity level 5, rather than relying on a pre-rendered archive. (Note: Frost is excluded because the source code referenced frost texture images, which we could not locate).

B Implementation

B.1 AnchorCLIP

Our method is implemented in Python using the PyTorch framework. We have used a T4 GPU. CIFAR10 and CIFAR100 training and inference (and ImageNet-C inference) can be done in a CPU as well but is more time consuming. For ImageNet training, it will be infeasible without a GPU. On a T4 GPU, CIFAR10/CIFAR100 training takes 2.9 iterations/second, and inference takes 5.7 iterations/sec.

We utilize an adapter-based approach integrated with intermediate layers of the vision backbone. For all datasets, we extract features from intermediate layers at indices 9, 10, and 11 as inputs to the adapter, in addition to the final embedding.

For CIFAR-100 and ImageNet, we set the learning rate to 1×10^{-4}. For CIFAR-10, we used a slightly lower learning rate of 5×10^{-5}. Across all datasets, we maintained a consistent training and evaluation batch size of 64, utilized 4 workers for data loading, and employed a random selection noise type and severity for each image during training.

B.2 BASELINES

We run both CLIP and SpLiCE (Bhalla et al., 2024) by forking and modifying the open source repository for SpLiCE, located in `https://github.com/AI4LIFE-GROUP/SpLiCE`. SpLiCE is run with L1 penalty 0.25 and the default skl solver. For all baseline runs, we use a batch size of 128. For TeCOA (Mao et al., 2023) and FARE (Schlarmann et al., 2024), we use the pre-trained CLIP models made available by authors of FARE (Schlarmann et al., 2024). These are located at `https://huggingface.co/chs20/tecoa4-clip` and `https://huggingface.co/chs20/fare4-clip` for TeCOA and FARE respectively. Both the models are trained on ImageNet. We use COLA (Zhu et al., 2025) by forking and modifying the source code released by the authors in `https://github.com/zhuhsingyuu/COLA`.

Our task-specific baseline is a linear adapter — a single bias-free 512×512 linear layer applied with a residual interpolation ($\alpha = 0.2$), trained with an MSE loss under an identical protocol, i.e. trained with the same subset of corruptions and randomly chosen severity and the same datasets.

C ADDITIONAL RESULTS

C.1 EXPERIMENT REPEATABILITY

We ran a set of 5 repeats with different random seeds for CIFAR-10 (different from Table 1). In Table 8, we report the mean and the standard deviations. These results also show that our improvements over the baselines are a lot more than 2 standard deviations of the mean for most corrupted types (except gaussian noise and pixelate)

Table 8: AnchorCLIP performance details: mean and standard deviation across corruption types(CIFAR-10-C, severity=5)

Corruption Type	**Mean**	**Std Dev**
Gaussian Noise	0.4925	0.0177
Shot Noise	0.5245	0.0178
Impulse Noise	0.5342	0.0134
Defocus Blur	0.8473	0.0032
Glass Blur	0.6064	0.0127
Motion Blur	0.8318	0.0019
Zoom Blur	0.8583	0.0018
Snow	0.8436	0.0021
Frost	0.8388	0.0019
Fog	0.8421	0.0020
Brightness	0.9223	0.0010
Contrast	0.8185	0.0034
Elastic Trans.	0.7556	0.0040
Pixelate	0.5310	0.0144
JPEG Comp.	0.6791	0.0040

C.2 ADDITIONAL DATASETS

In the main paper, we presented the results for the mean accuracies for additional datasets in Table 3. Here, we present the complete results for each noise in Table 9. The results establish that an AnchorCLIP model trained on Imagenet is competitive across a range of datasets and noises.

C.3 ADDITIONAL BACKBONE

In this section, we show additional results, showing the performance of AnchorCLIP on a OpenCLIP VIT-B/16 backbone in Table 10. Note that because pretrained TeCoA and FARE checkpoints are

Table 9: Zero-shot generalization to Caltech101, DTD, and OxfordPets under 14 corruptions (severity=5). AnchorCLIP is trained on ImageNet only, transferred zero-shot.

Dataset	Corruption Type	CLIP	TeCoA	FARE	AnchorCLIP (Ours)
Caltech101	Gaussian Noise	0.6223	0.4446	0.5371	**0.6438**
	Shot Noise	0.6073	0.4669	0.5582	**0.6458**
	Impulse Noise	0.5976	0.4410	0.5428	**0.6430**
	Defocus Blur	0.7675	0.3282	0.3501	**0.7830**
	Glass Blur	0.4661	0.4495	0.4702	**0.4787**
	Motion Blur	0.3282	0.2256	0.1866	**0.3984**
	Zoom Blur	0.5692	0.4536	0.4337	**0.5753**
	Snow	**0.7850**	0.5728	0.6256	0.7842
	Fog	**0.8568**	0.0819	0.1164	0.8479
	Brightness	0.8953	0.7286	0.7728	**0.8978**
	Contrast	0.8361	0.0446	0.0657	**0.8434**
	Elastic Trans.	0.4479	**0.6864**	0.6414	0.4576
	Pixelate	0.8130	0.7619	0.7513	**0.8231**
	JPEG Comp.	0.8519	0.8576	**0.8795**	0.8527
	Mean	0.6746	0.4674	0.4951	**0.6911**
DTD	Gaussian Noise	0.1501	0.1076	0.1631	**0.2187**
	Shot Noise	0.1602	0.1217	0.1643	**0.2181**
	Impulse Noise	0.1537	0.1011	0.1548	**0.2305**
	Defocus Blur	0.2145	0.0993	0.1111	**0.2364**
	Glass Blur	**0.1879**	0.1247	0.1413	0.1791
	Motion Blur	0.0780	0.0662	0.0467	**0.1342**
	Zoom Blur	0.1814	0.1537	0.1507	**0.2045**
	Snow	0.2719	0.1631	0.1749	**0.2861**
	Fog	0.3168	0.0656	0.0733	**0.3245**
	Brightness	0.4326	0.2116	0.2624	**0.4338**
	Contrast	0.2565	0.0201	0.0231	**0.3085**
	Elastic Trans.	**0.2654**	0.1862	0.2216	0.2476
	Pixelate	0.2447	0.2216	0.2234	**0.2683**
	JPEG Comp.	0.3180	0.2843	0.3085	**0.3245**
	Mean	0.2308	0.1376	0.1585	**0.2582**
OxfordPets	Gaussian Noise	0.2216	0.2385	0.1357	**0.2998**
	Shot Noise	0.2548	0.2456	0.1649	**0.3333**
	Impulse Noise	0.2289	0.2448	0.1303	**0.3053**
	Defocus Blur	0.5179	0.2339	0.1447	**0.5203**
	Glass Blur	0.3271	**0.3322**	0.2276	0.2720
	Motion Blur	0.1379	0.1202	0.0387	**0.1897**
	Zoom Blur	**0.3592**	0.3581	0.2295	0.3510
	Snow	**0.5874**	0.3451	0.3025	0.5590
	Fog	**0.6413**	0.0472	0.0273	0.6023
	Brightness	**0.7863**	0.5443	0.5255	0.7705
	Contrast	**0.5397**	0.0305	0.0273	0.5124
	Elastic Trans.	0.4824	**0.6323**	0.5198	0.4723
	Pixelate	0.6138	**0.6351**	0.5080	0.6154
	JPEG Comp.	0.5827	**0.6945**	0.6509	0.5557
	Mean	0.4486	0.3359	0.2595	**0.4542**

unavailable for OpenCLIP ViT-B/16, we compare against CLIP and a linear adapter trained using the same corruption training protocol. These results show that AnchorCLIP performs better than CLIP as well as a task specific linear adapter trained on the same common corruption subsets as AnchorCLIP, for this backbone as well.

Table 10: Zero-shot classification accuracy on CIFAR-100C (Severity=5) with a second backbone, OpenCLIP ViT-B-16.

Corruption Type	CLIP	Linear Adapter	AnchorCLIP (Ours)
Gaussian Noise	0.2207	0.2477	**0.2877**
Shot Noise	0.2484	0.2760	**0.3100**
Impulse Noise	0.2679	0.2983	**0.3238**
Defocus Blur	0.5347	0.5433	**0.5896**
Glass Blur	0.1615	0.1922	**0.2183**
Motion Blur	0.4843	0.5078	**0.5515**
Zoom Blur	0.5699	0.5809	**0.6126**
Snow	0.5694	0.5809	**0.5932**
Frost	0.5551	0.5716	**0.5853**
Fog	0.5184	0.5333	**0.5554**
Brightness	0.6519	0.6635	**0.6834**
Contrast	0.3978	0.4062	**0.4814**
Elastic Trans.	0.3274	0.3581	**0.3828**
Pixelate	0.2561	0.2700	**0.3513**
JPEG Comp.	0.3450	0.3600	**0.4098**
Mean	0.4072	0.4260	**0.4624**
Clean	0.7701	0.7637	**0.7735**

C.4 Interpretability Based Baselines

Here, we show two additional baselines. COLA (Zhu et al., 2025) and SpLiCE (Bhalla et al., 2024) for CIFAR-10-C and CIFAR-100-C in Table 11, and SpLiCE (Bhalla et al., 2024) for ImageNet-C in Table 12 (For ImageNet-C, we omit COLA because of scaling issues). COLA (Zhu et al., 2025) is particularly interesting to us, since it also depends on interpretability, enforcing robustness by projecting the image embeddings to a semantically defined space. However, as these results show, interpretability based methods designed for adversarial robustness are not well suited to common corruptions. SpLiCE (Bhalla et al., 2024) attempts to identify the interpretable part of CLIP embeddings using a sparse linear model to find the best linear decomposition of image embeddings as a weighted sum of concept embeddings. While SpLiCE (Bhalla et al., 2024) is good at zero shot classification for clean embeddings, it is not able to do a clean linear separation of the interpretable and noise components in a noisy image. As a whole, these results point to the weakness of current methods based on interpretability to classify commonly corrupted noisy images.

C.5 Full Ablation Results

Due to space constraints, we could not show all the ablation results in the main paper. Below, in Tables 13, 14, 15 and 16 , we show the ablation results for different interpolation ratios, loss types, adapter architecture, and transformer layers. Note that in Table 16 using all layers is also competitive. It is difficult to make choices that win for all noises. But since the late layers show similar performance for lesser number of parameters, we go only with the late layers, in the spirit of a smaller model.

Table 11: Comparison with SpLiCE and COLA on CIFAR-10C and CIFAR-100C (Severity=5).

Dataset	Corruption Type	CLIP	SpLiCE	COLA	AnchorCLIP (Ours)
CIFAR-10C	Gaussian Noise	0.4172	0.2910	0.3108	**0.5096**
	Shot Noise	0.4659	0.3267	0.3480	**0.5419**
	Impulse Noise	0.4629	0.3162	0.3834	**0.5537**
	Defocus Blur	0.7762	0.7011	0.6458	**0.8492**
	Glass Blur	0.4498	0.3248	0.3115	**0.6040**
	Motion Blur	0.7161	0.6713	0.6463	**0.8264**
	Zoom Blur	0.7770	0.7332	0.6747	**0.8585**
	Snow	0.8085	0.7429	0.6982	**0.8458**
	Frost	0.8038	0.7286	0.6464	**0.8376**
	Fog	0.7906	0.7011	0.6440	**0.8430**
	Brightness	0.8991	0.8547	0.8316	**0.9227**
	Contrast	0.7108	0.6295	0.5683	**0.8159**
	Elastic Trans.	0.6468	0.5636	0.5207	**0.7592**
	Pixelate	0.4995	0.3687	0.3855	**0.5368**
	JPEG Comp.	0.6344	0.5272	0.4560	**0.6860**
	Mean Accuracy	0.6572	0.5654	0.5381	**0.7327**
CIFAR-100C	Gaussian Noise	0.1805	0.1235	0.0946	**0.2521**
	Shot Noise	0.2026	0.1409	0.1073	**0.2731**
	Impulse Noise	0.2018	0.1019	0.1316	**0.2573**
	Defocus Blur	0.5145	0.4238	0.3559	**0.5635**
	Glass Blur	0.1831	0.1355	0.1231	**0.2452**
	Motion Blur	0.4786	0.4021	0.3347	**0.5374**
	Zoom Blur	0.5448	0.4593	0.3839	**0.5902**
	Snow	0.5427	0.4477	0.4311	**0.5763**
	Frost	0.5249	0.4401	0.3418	**0.5542**
	Fog	0.4748	0.4151	0.3121	**0.5216**
	Brightness	0.6363	0.5524	0.5326	**0.6750**
	Contrast	0.3913	0.3337	0.1994	**0.4795**
	Elastic Trans.	0.3328	0.2605	0.2496	**0.4017**
	Pixelate	0.2193	0.1717	0.1484	**0.2921**
	JPEG Comp.	0.3590	0.2816	0.2084	**0.4044**
	Mean Accuracy	0.3858	0.3127	0.2636	**0.4416**

Table 12: Comparison with SpLiCE on ImageNet-C (Severity=5).

Corruption Type	CLIP	SpLiCE	AnchorCLIP (Ours)
Gaussian Noise	0.0732	0.0487	**0.1585**
Shot Noise	0.0816	0.0529	**0.1742**
Impulse Noise	0.0747	0.0497	**0.1602**
Defocus Blur	0.2116	0.1217	**0.3145**
Glass Blur	0.0869	0.0503	**0.1463**
Motion Blur	0.1436	0.0806	**0.2633**
Zoom Blur	0.1328	0.0797	**0.2254**
Snow	0.1886	0.1071	**0.2759**
Frost	0.1968	0.1194	**0.2741**
Fog	0.2467	0.1480	**0.3485**
Brightness	0.3992	0.2215	**0.5379**
Contrast	0.1435	0.0854	**0.2379**
Elastic Trans.	0.1422	0.0775	**0.1976**
Pixelate	0.2614	0.1449	**0.3807**
JPEG Comp.	0.2583	0.1529	**0.3661**
Mean Accuracy	0.1761	0.1027	**0.2707**

Table 13: Sensitivity to α, the residual interpolation weight between the adapter output and the final CLIP embedding on CIFAR-100-C (severity=5). α=0 recovers the unadapted CLIP embedding; α=1 uses only the adapter output. **Bold** marks our locked-in choice, α=0.75.

α	Gaussian Noise	Defocus Blur	Brightness
0.00	0.1805	0.5145	0.6363
0.20	0.2394	0.5404	0.6607
0.50	0.2429	0.5623	0.6707
0.60	0.2393	0.5543	0.6682
0.70	0.2418	0.5623	0.6767
0.75 (Ours)	**0.2521**	**0.5635**	0.6750
1.00	0.2403	0.5453	0.6484

Table 14: Ablation study of the loss composition on CIFAR-100-C (Severity 5). Top: sweep over the $\mathcal{L}_{\text{sem}}$/$\mathcal{L}_{\text{concept}}$ weighting. Bottom: sweep over the top-k concept-selection solver's k, at the locked $0.7\,\mathcal{L}_{\text{sem}} + 0.3\,\mathcal{L}_{\text{concept}}$ split. Both blocks' **(Final)** row is the same primary checkpoint reported in Table 2. **Bold** indicates the best performance in each column within each block.

Objective	Brightness	Defocus Blur	Gaussian Noise	Snow
$\mathcal{L}_{\text{concept}}$ only	0.6596	0.5440	0.2229	0.5655
$\mathcal{L}_{\text{sem}}$ only	0.6333	0.5113	0.2177	0.5295
$0.3\,\mathcal{L}_{\text{sem}} + 0.7\,\mathcal{L}_{\text{concept}}$	0.6640	0.5532	0.2321	0.5673
$0.7\,\mathcal{L}_{\textbf{sem}} + 0.3\,\mathcal{L}_{\textbf{concept}}$ **(Final)**	**0.6750**	**0.5635**	**0.2521**	**0.5763**
$k = 5$	0.6671	0.5552	0.2385	0.5693
$k = 10$ **(Final)**	**0.6750**	**0.5635**	**0.2521**	**0.5763**
$k = 15$	0.6703	0.5568	0.2357	0.5703

Table 15: Ablation study of architectural design choices on CIFAR-100-C (Severity 5). We evaluate different adapter configurations, including bottleneck, expanded, and attentional variants, against our final **Concat Wide** design. **Bold** indicates the best performance in each column.

Architecture	Brightness	Defocus Blur	Gaussian Noise	Snow
Bottleneck	0.6443	0.5282	0.2083	0.5569
Expanded	0.6674	0.5519	0.2292	0.5679
Attentional	0.6624	0.5445	0.2329	0.5492
Concat Wide (Final)	**0.6750**	**0.5635**	**0.2521**	**0.5763**

Table 16: Ablation study of attention layer placement on CIFAR-100-C (Severity 5). We compare applying the AnchorCLIP adapter to different stages of the ViT backbone. **Bold** indicates the best performance in each column.

Layer Configuration	Brightness	Defocus Blur	Gaussian Noise	Snow
Early Layers	0.6636	0.5482	0.2272	0.5669
All Layers	0.6682	0.5608	0.2361	0.5720
Late Layers (Final)	**0.6750**	**0.5635**	**0.2521**	**0.5763**

D ADDITIONAL VISUALIZATIONS

We pick three random imageNet-C indices and in Figure 6, Figure 7 and Figure 8, show the concepts most aligned with the corrupted embeddings, and those most aligned with the AnchorCLIP embeddings. We also used SpLiCEBhalla et al. (2024) for this interpretation, but the results were identical to choosing the top 5 concepts in terms of cosine similarity.

The general observation is that corruption causes the embeddings to be aligned with keywords corresponding to high frequency noises and textures. We see "abstract", "psychedelic", "television" occur multiple times, in analyzing the concepts for corrupted images.

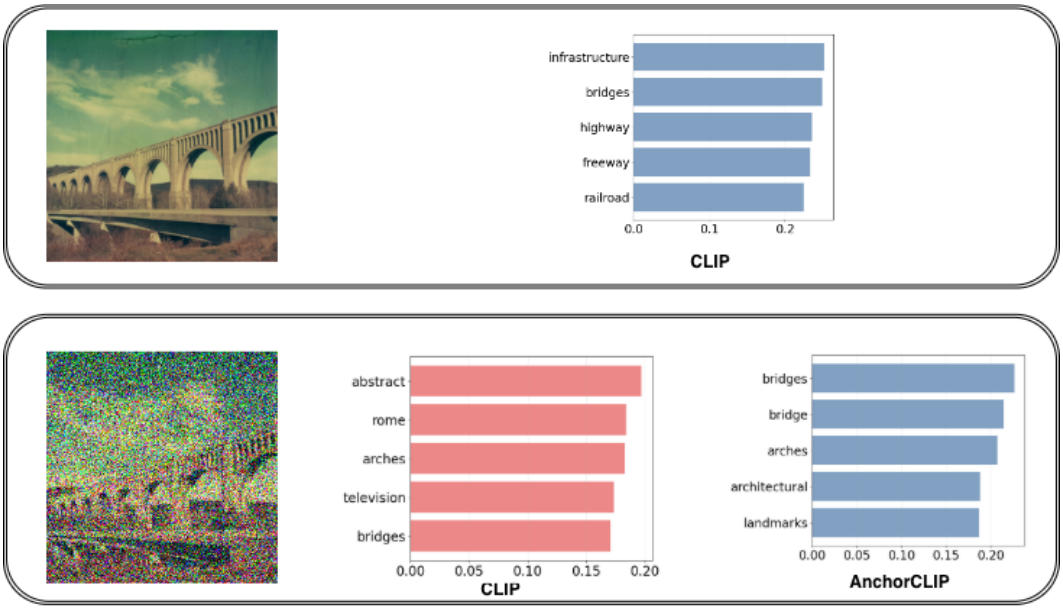


Figure 6: First randomly chosen example.

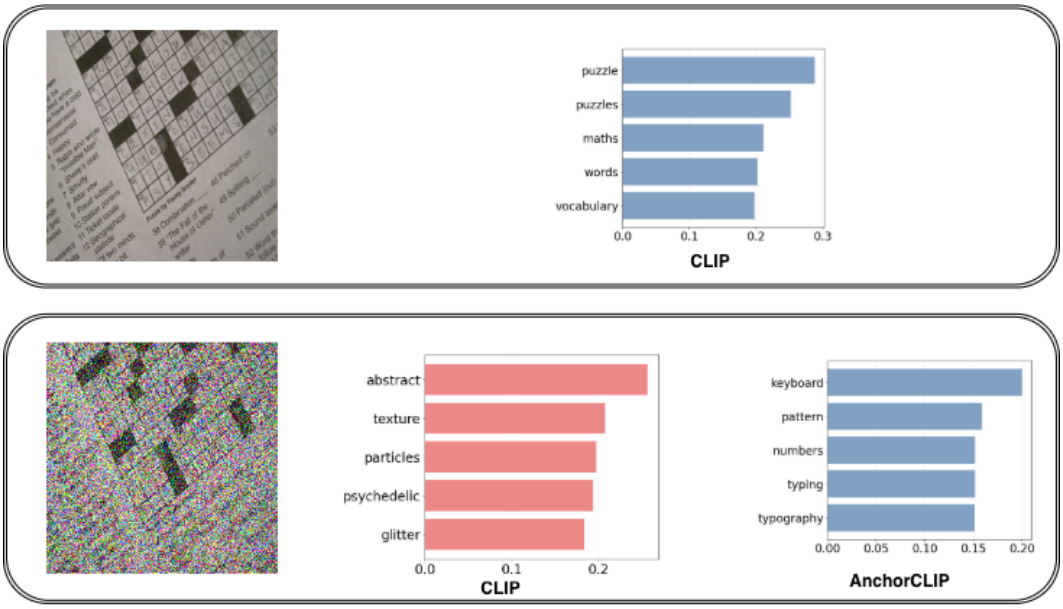


Figure 7: Second randomly chosen example.

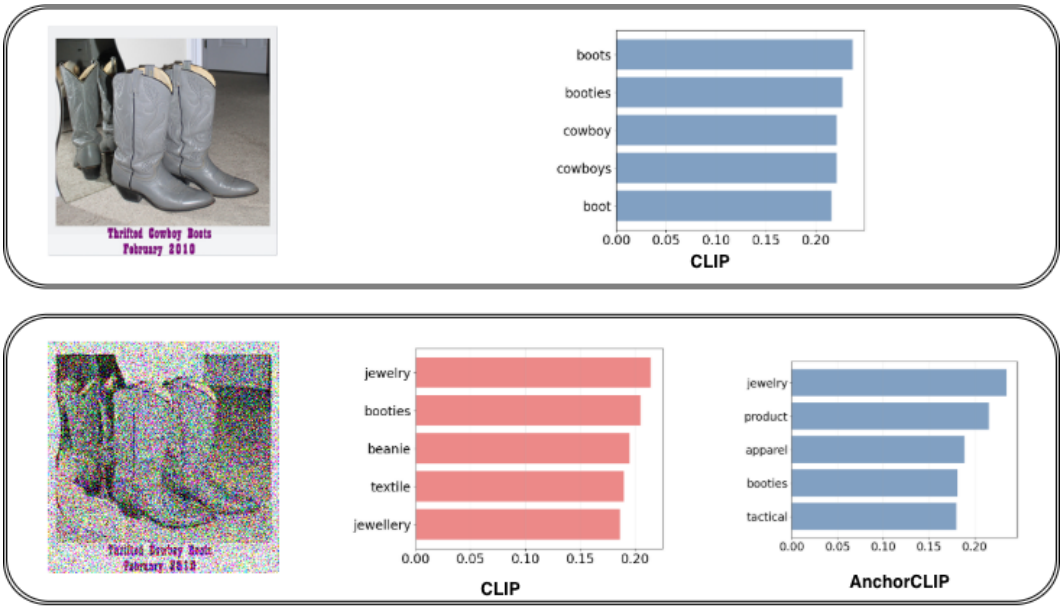


Figure 8: Third randomly chosen example.