Enhance Vision-Language Alignment with Noise

Sida Huang[1,2], **Hongyuan Zhang**[2, 3*], **Xuelong Li**[2*]

[1]Northwestern Polytechnical University
[2]Institute of Artificial Intelligence (TeleAI), China Telecom
[3]The University of Hong Kong
{sidahuang2001, hyzhang98}@gmail.com, xuelong_li@ieee.org

Abstract

With the advancement of pre-trained vision-language (VL) models, enhancing the alignment between visual and linguistic modalities in downstream tasks has emerged as a critical challenge. Different from existing fine-tuning methods that add extra modules to these two modalities, we investigate whether the frozen model can be fine-tuned by customized noise. Our approach is motivated by the scientific study of beneficial noise, namely Positive-incentive Noise (Pi-noise or π-noise) , which quantitatively analyzes the impact of noise. It therefore implies a new scheme to learn beneficial noise distribution that can be employed to fine-tune VL models. Focusing on few-shot classification tasks based on CLIP, we reformulate the inference process of CLIP and apply variational inference, demonstrating how to generate π-noise towards visual and linguistic modalities. Then, we propose **P**ositive-**i**ncentive **N**oise **I**njector (PiNI), which can fine-tune CLIP via injecting noise into both visual and text encoders. Since the proposed method can learn the distribution of beneficial noise, we can obtain more diverse embeddings of vision and language to better align these two modalities for specific downstream tasks within limited computational resources. We evaluate different noise incorporation approaches and network architectures of PiNI. The evaluation across 11 datasets demonstrates its effectiveness. Our code is available at: https://github.com/hyzhang98/PiNI.

1 Introduction

Vision and language are two crucial modalities in the real world. To build association and enable collaboration between these two modalities, vision-language (VL) models have emerged in recent years. Based on the classical dual-stream VL model CLIP (Radford et al. 2021), LLaVA (Liu et al. 2024b,a) has recently achieved remarkable performance. In these models, the embeddings of these two modalities are aligned during the pre-training phase. When applying CLIP models to downstream unseen data, *inherent dataset bias can lead to misalignment between visual and linguistic representations*. To realign these two modalities, fine-tuning all parameters requires extensive computational resources. Even with sufficient computational resources, full fine-tuning on limited downstream data also carries a risk of overfitting and forgetting the knowledge acquired during pre-training stage. Consequently, it is crucial to effectively and efficiently fine-tune VL models to achieve better alignment.

To address this challenge, existing methods such as prompt tuning (Lester, Al-Rfou, and Constant 2021) and adapter tuning (Houlsby et al. 2019) are proposed. Prompt tuning sets the word embeddings of prompts as learnable parameters and adapter tuning inserts a learnable shallow neural network to models. Different from these fine-tuning methods that add extra modules or concatenate extra tokens, we focus on an interesting question: ***Can a frozen model be tuned with customized noise?***

It is motivated by Positive-incentive Noise (Pi-noise or π-noise) (Li 2022; Zhang, Huang, and Li 2023), which quantitatively analyzes the impact of noise and investigates how to generate beneficial noise for a specific task. Different from traditional concepts that treat the randomness of noise as a harmful factor, the noise $\mathcal{E}$ can decrease the uncertainty of predictions and simplify the task if it satisfies

$$I(\mathcal{T}, \mathcal{E}) > 0 \Leftrightarrow H(\mathcal{T}) > H(\mathcal{T}|\mathcal{E}), \tag{1}$$

where $\mathcal{T}$ is the definition of a specific task from a probabilistic perspective, $I(\cdot, \cdot)$ denotes mutual information and H represents entropy. It should be clarified that our method is distinct from noise addition in data augmentation, as discussed in Section 2.2.

The advantages of noise to enhance alignment can be discussed from both vision and language perspectives. **From the perspective of vision**, the biases in different visual datasets (Liu and He 2024) can lead to performance degeneration in VL models. We hypothesize that the bias associated with each image follows a certain distribution, which can be mitigated by adding noise also from a specific distribution. As shown in Figure 4, the images with added noise exhibit more universal visual features. Additionally, the limited number of images makes it difficult to adequately sample from the continuous space of image embeddings. This dilemma can be alleviated by adding randomly sampled beneficial noise into the visual level, for which more diverse visual embeddings can be obtained. **From the perspective of language**, the templates of hand-crafted prompts and learnable prompts may be not effective enough, which limit the diversity of linguistic modality. Figure 1 shows two strategies and our proposed scheme for prompt construction. In

*Corresponding author.

the original CLIP (Radford et al. 2021), the hand-crafted prompt is used, e.g., a typical form "a photo of a [class]". Some prompt tuning works (Zhou et al. 2022b,a; Gao et al. 2024a) replace the hand-crafted prompt with trainable word embeddings to mine more precise description of images. These prompts are class-instanced. In other words, a class label corresponds to only one prompt. Nevertheless, a class should be related to various prompts (with different qualities). By sampling different beneficial noise from a customized distribution and adding it to hand-crafted prompts, new prompts can be easily obtained. It is thus easy to find that the prompts should also obey some probability distribution, ensuring the semantic richness of prompts.

Towards prompt-based image classification task, we propose PiNI based on π-noise theory which can enhance the alignment of VL models. Overall, the contributions can be summarized as follows:

- We propose **P**ositive-**i**ncentive **N**oise **I**njector (PiNI), a fine-tuning method for CLIP through injecting customized beneficial noise into it. To the best of our knowledge, this is the first noise-based approach to fine-tune CLIP.
- We reformulate the inference process of CLIP, by treating the prompt as a variable to analyze the effect of both images and prompts on the probability of category labels. It therefore offers a paradigm to refine CLIP from a probabilistic aspect.
- We apply variational inference and Monte Carlo method to converting the complex loss with noise distribution to a tractable one, which guides us in generating and injecting noise. Extensive experiments validate our idea.

2 Related Works

2.1 Fine-Tuning Methods for VL Models

It is crucial to fine-tune VL models for downstream tasks with limited computational resources. To address this challenge, some Parameter-Efficient Fine-Tuning (PEFT) methods have been proposed, primarily including parameter tuning, adapter tuning (Houlsby et al. 2019) and prompt tuning (Lester, Al-Rfou, and Constant 2021). The parameters of VL model are directly tuned in parameter tuning. LoRA (Hu et al. 2021) and Bitfit (Zaken, Goldberg, and Ravfogel 2022) focus on tuning the weights and biases of models, respectively. Adapter tuning inserts a trainable module named Adapter into a frozen model. CLIP-Adapter (Gao et al. 2024b) and VL-Adapter (Sung, Cho, and Bansal 2022) utilize Adapter in fine-tuning VL models. Prompt tuning was first proposed in NLP, which adds a learnable prompt template to pre-trained language models. Since CLIP model includes a text encoder, numerous works have attempted to introduce prompt tuning to CLIP (Zhou et al. 2022b,a; Gao et al. 2024a; Guo et al. 2023). Beyond text encoders, VPT (Jia et al. 2022) and MaPLe (Khattak et al. 2023) extend prompt tuning to visual encoders. Different from the above works, we focus on learning noise distribution to sample more visual and text embeddings without altering the architecture of VL models.

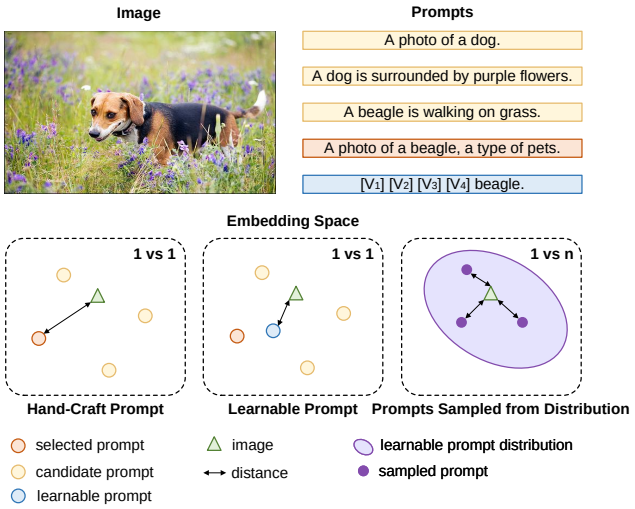

Figure 1: Different strategies for constructing prompts. A single image can be described by multiple prompts. The first and second strategies each correspond to one prompt per image. After fine-tuning, the learnable prompt becomes closer to the image in the embedding space. After learning the noise distribution in prompts, we can easily sample prompts with richer and more precise semantics as needed.

2.2 Difference with Data Augmentation

Adding noise during training as a data augmentation strategy (Cubuk Ekin et al. 2019) is common in computer vision tasks, such as adversarial training (Zhong et al. 2022; Zhang et al. 2023; Li et al. 2022). The data augmentation only participates in the training phase and is excluded during the inference phase. However, our proposed method injects noise for both the training and inference phases. For adversarial augmentation, it aims at adding perturbations to the input of networks, thereby enhancing robustness. Conversely, the noise in PiNI is employed to simplify tasks instead of creating difficulty for base models.

3 Method

When applying VL models to unseen downstream data, dataset bias (Liu and He 2024) can lead to misalignment in representations of vision and language. To mitigate this issue, we focus on achieving better vision-language alignment in the prompt-based image recognition task $\mathcal{T}$. Inspired by π-noise (Li 2022; Zhang et al. 2024) , we propose a new scheme for learning beneficial noise distribution, namely **P**ositive-**i**ncentive **N**oise **I**njector (PiNI).

In this section, we first reformulate the inference process of CLIP in Section 3.1. Section 3.2 introduces how to define the task entropy on the specific dataset $\mathcal{X}$, which is vital to calculate the optimization objective $I(\mathcal{T}, \mathcal{E})$ when generating π-noise. To optimize this objective, the variational approximation is applied to getting its upper bound in Section 3.3. In the final approximate loss, we adopt a learnable function to approximate noise and determine the noise incorporation approaches, which are discussed in Section 3.4.

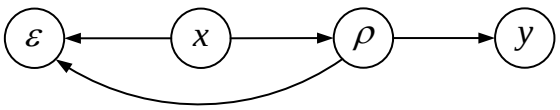

Figure 2: The probabilistic graphical model of PiNI.

3.1 Reformulation of CLIP Inference

Different from classical image classification models that only require images as input, CLIP additionally needs a prompt set to obtain the position of each category in the embedding space. Obviously, the inference process of CLIP relies on the prompts. However, in traditional models, the label probability is usually modeled as $p(y|x)$, which ignores the role of prompts.

It is rational to introduce a separate variable for prompts. Given an image x and a set of prompts $\rho_i \in \mathcal{P}, i \in 1, 2, ..., N$, the probability of class $y \in 1, 2, ..., N$ can be reformulated as

$$p(y|x, \mathcal{P}) = \frac{\exp(\text{sim}(V(x), T(\rho_y))/\tau)}{\sum_{i=1}^{N} \exp(\text{sim}(V(x), T(\rho_i))/\tau)}, \quad (2)$$

where V and T represent the visual and text encoders, $\text{sim}(\cdot, \cdot)$ is the similarity between two vectors, and τ is a temperature parameter. Note that $\mathcal{P}$ is a variable, and there exists a one-to-one correspondence between prompt ρ_i and label y. In CLIP model, $\text{sim}(\cdot, \cdot)$ is the cosine similarity.

3.2 π-noise Regarding Prompts

Primarily, it is crucial to evaluate the complexity of a classification task on an arbitrary dataset $\mathcal{X}$ and prompt set $\mathcal{P}$. Inspired by (Li 2022), task entropy can be used to formulate the complexity as

$$\begin{aligned} H(\mathcal{T}) &= H(y|x, \mathcal{P}) \\ &= \mathbb{E}_{x\sim\mathcal{D}_\mathcal{X}} \mathbb{E}_{y\sim p(y|x,\mathcal{P})}[-\log p(y|x, \mathcal{P})] \\ &= \mathbb{E}_{p(x,y|\mathcal{P})}[-\log p(y|x, \mathcal{P})], \end{aligned} \quad (3)$$

where $\mathcal{D}_\mathcal{X}$ is the data distribution over $\mathcal{X}$. This equation measures complexity by computing the uncertainty of labels. For a given image, it may resemble multiple categories, which increases the uncertainty of the label y. The noise $\mathcal{E}$ can decrease this uncertainty if it satisfies Eq. (1). The definition of $I(\mathcal{T}, \mathcal{E})$ is

$$I(\mathcal{T}, \mathcal{E}) = H(\mathcal{T}) - H(\mathcal{T}|\mathcal{E}). \quad (4)$$

Similar to $H(\mathcal{T})$, $H(\mathcal{T}|\mathcal{E})$ is defined as

$$H(\mathcal{T}|\mathcal{E}) = \mathbb{E}_{p(x,y,\varepsilon|\mathcal{P})}[-\log p(y|x, \varepsilon, \mathcal{P})]. \quad (5)$$

As $H(\mathcal{T})$ is a constant term for fixed CLIP model, maximizing $I(\mathcal{T}, \mathcal{E})$ is equivalent to minimizing $H(\mathcal{T}|\mathcal{E})$.

3.3 Variational Inference for the Intractable Problem

Since $p(y|x, \varepsilon, \mathcal{P})$ is difficult to calculate precisely due to the integral, we apply the variational inference technique (Blei, Kucukelbir, and McAuliffe 2017) to Eq. (5) and obtain a variational upper bound of $H(\mathcal{T}|\mathcal{E})$ as

$$\mathcal{L} = \mathbb{E}_{p(x,y,\varepsilon|\mathcal{P})}[-\log q(y|x, \varepsilon, \mathcal{P})] \geq H(\mathcal{T}|\mathcal{E}), \quad (6)$$

where $q(y|x, \varepsilon, \mathcal{P})$ is a tractable approximation of $p(y|x, \varepsilon, \mathcal{P})$. The above inequality is derived from the non-negative property of KL-Divergence as follows,

$$KL(p||q) \geq 0 \Leftrightarrow \mathbb{E}_{p(x)}[\log p(x)] \geq \mathbb{E}_{p(x)}[\log q(x)]. \quad (7)$$

The current problem is how to effectively compute this variational bound. Monte Carlo estimation can be applied to data distribution $\mathcal{D}_\mathcal{X}$ to obtain image-label pairs (x_i, y_i). So the variatonal bound is further approximated as

$$\mathcal{L} \approx \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{p(\varepsilon|x_i,y_i,\mathcal{P})}[-\log q(y_i|x_i, \varepsilon, \mathcal{P})]. \quad (8)$$

For further calculation, $p(\varepsilon|x_i, y_i, \mathcal{P})$ needs to be simplified as there are three given variables. In Figure 2, the probabilistic graphical model illustrates the dependencies among variables. The noise and the label are conditionally independent given the prompt: $\varepsilon \perp y \mid \rho$. Accordingly, we can obtain the following result from the conditional independence,

$$p(\varepsilon|x, y, \rho) = p(\varepsilon|x, \rho). \quad (9)$$

The above derivations indicate that the generated noise should be controlled by the set of prompts so that it can exclude interference from information irrelevant to prompts. Therefore, the variational bound is further transformed to

$$\mathcal{L} \approx \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{p(\varepsilon|x_i,\mathcal{P})}[-\log q(y_i|x_i, \varepsilon, \mathcal{P})]. \quad (10)$$

We assume that $p(\varepsilon|x, \mathcal{P})$ follows a classical Gaussian distribution, whose distribution parameters μ and Σ is approximated using a learnable function $f_\theta(x_i, \mathcal{P})$:

$$(\mu, \Sigma) = f_\theta(x_i, \mathcal{P}), \quad (11)$$

where θ represents the parameters of the function. However, calculating the integral of the continuous variable ε is difficult. The Monte Carlo method is applied again to address this problem. To ensure the back propagation of gradients in $f_\theta(x_i, \mathcal{P})$, the reparameterization trick (Kingma and Welling 2014) introduces an auxiliary variable $\epsilon \sim N(0, I)$ satisfying $p(\varepsilon|x_i, \mathcal{P})d\varepsilon = p(\epsilon)d\epsilon$. Subsequently, a learnable function G_θ is established to generate noise

$$\varepsilon = G_\theta(\epsilon, x_i, \mathcal{P}) = \Sigma_\theta(x_i, \mathcal{P}) \cdot \epsilon + \mu_\theta(x_i, \mathcal{P}). \quad (12)$$

We randomly sample ϵ m times for each x_i to get the final loss function for training

$$\begin{aligned} \mathcal{L} &\approx \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{p(\epsilon)}[-\log q(y_i|x_i, G_\theta(\epsilon, x_i, \mathcal{P}), \mathcal{P})] \\ &\approx \frac{1}{n \cdot m} \sum_{i=1}^{n} \sum_{j=1}^{m} [-\log q(y_i|x_i, G_\theta(\epsilon_{ij}, x_i, \mathcal{P}), \mathcal{P})]. \end{aligned} \quad (13)$$

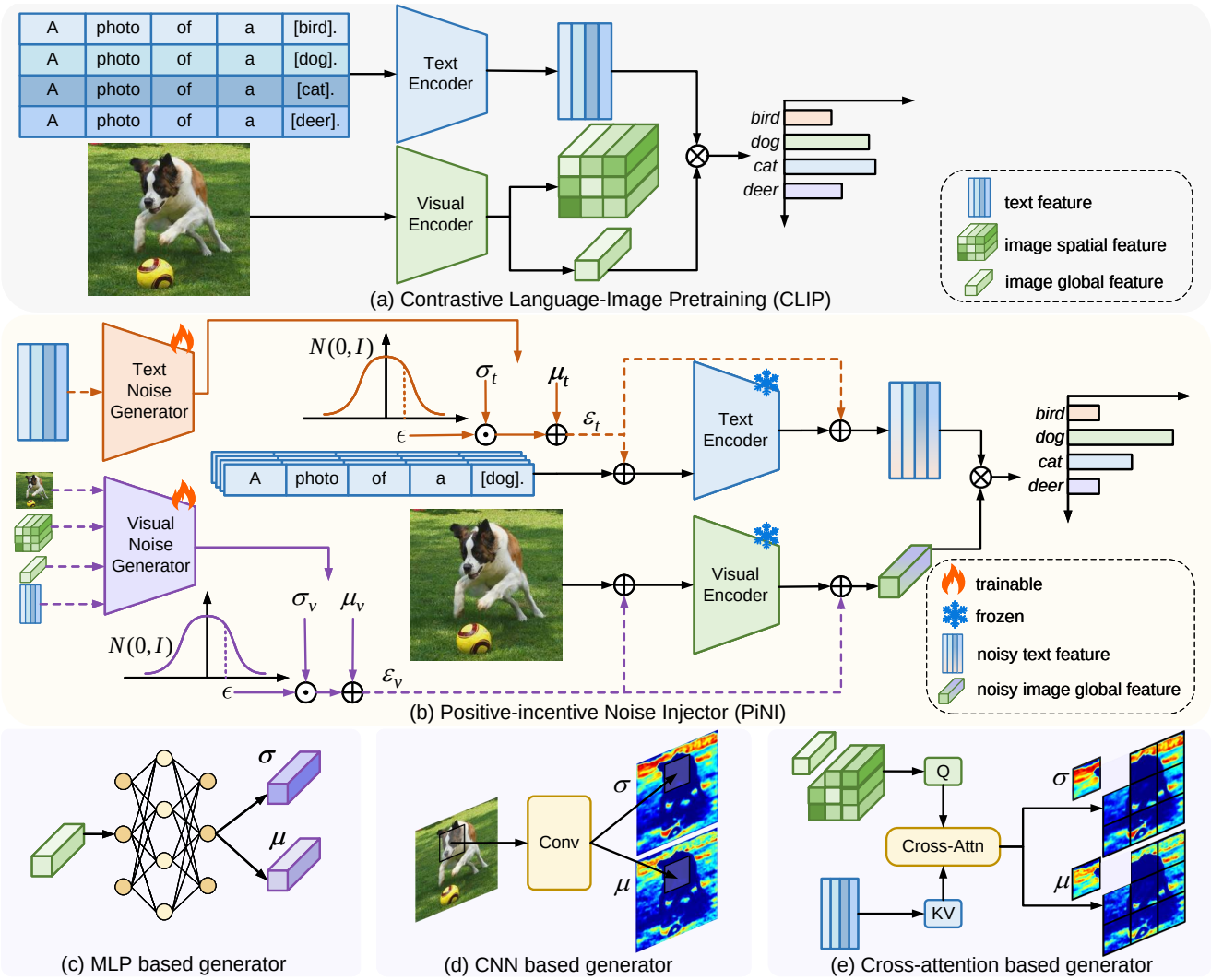

Figure 3: **(a)** The inference process of CLIP, whose output features are utilized in PINI. **(b)** The framework of PiNI, which includes two procedures for adding noise. There are multiple options for the input of noise generators and locations of noise injection, whose potential transmission paths are represented with dashed lines. In the figure, $\odot$ denotes the Hadamard product, $\oplus$ denotes matrix or vector addition, and $\otimes$ denotes matrix multiplication. **(c)(d)(e)** are three architectures of noise generators for learning distribution parameters.

3.4 Fine-Tuning with π-noise

In the derived final loss Eq. (13), there are two important components: $q(y|x,\varepsilon,\mathcal{P})$ and $\varepsilon = G_\theta(\epsilon,x,\mathcal{P})$. The former predicts the labels with the aid of noise. The specific approaches for incorporating noise will be discussed below. The latter $G_\theta(\epsilon,x,\mathcal{P})$ defined in Eq. (12) consists of two phases: (1) Sampling: $\epsilon \sim N(0,I)$ and (2) Distribution parameter estimation: $(\mu,\Sigma) = f_\theta(x,\mathcal{P})$. $f_\theta(x,\mathcal{P})$ can be implemented using various neural network architectures, which will be introduced in the following discussion. The detailed framework of PiNI is illustrated in Figure 3.

Noise Incorporation Approach When approximating $q(y|x,\varepsilon,\mathcal{P})$ to predict labels, we avoid altering the original inference process of CLIP as formulated in Eq. (2). It will remain the architecture of CLIP. A simple approach is to inject noise ε into the two encoders. To simplify the discussion, we employ the classical factorization trick to *decompose the noise into two components* $\varepsilon = \{\varepsilon_v, \varepsilon_t\}$, where ε_v and ε_t are injected into the visual and text encoders, respectively. Note that similar to the classical Naive Bayes model (Bishop 2006), a hypothesis that ε_v and ε_t are conditional independent is established. According to this hypothesis, ε_v and ε_t can be generated and utilized separately.

As shown in Figure 3(b), the visual noise ε_v can be injected into different locations of CLIP, including raw image or visual feature. For the text noise ε_t, it can be injected into word embeddings of prompts or text features. (Zhang, Zhu, and Li 2024) proposes to train multiple modules simultaneously in GNN. We will compare the effects of different

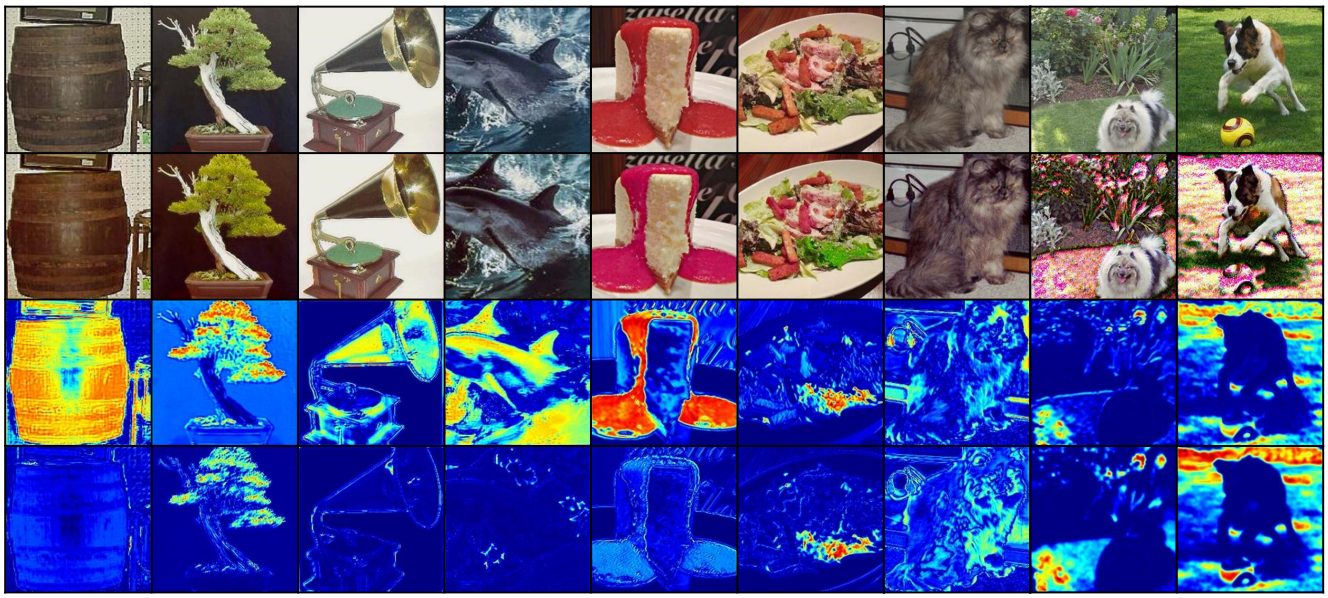

Figure 4: Visualization of generated noise injected into raw images. The first row shows the raw images. The second row displays the noise-injected images. The third and fourth rows present the heatmaps of the mean μ and variance σ for each pixel, respectively. In the first column, the noise deepens the color of an old barrel, making it look new again and thereby reducing the bias between datasets. The vegetation and ball in the last column are disturbed by noise, simplifying the task of recognizing.

injection locations in Section 4.2. Note that the noise has the same shape as the input at the injection location. Additionally, additive noise and multiplicative noise correspond to different methods of fusing input and noise. Unless specifically stated, we use additive noise by default.

Architecture of Noise Generator Now, we discuss different implementations of $f_\theta(x, \mathcal{P})$ in Eq. (11) to estimate the distribution parameters. To maintain the computational efficiency, we simply assume that ε follows uncorrelated multivariate Gaussian distribution, i.e., Σ is a diagonal matrix. The learnable variance parameters are represented as $\sigma = \mathrm{Diag}(\Sigma)$. Based on the above hypothesis, we separately generate ε_v and ε_t using $f_v(x, \mathcal{P})$ and $f_t(x, \mathcal{P})$. Subsequently, Eq. (11) and Eq. (12) are formulated as

$$\begin{aligned} \varepsilon_v &= \sigma_v \odot \epsilon_v + \mu_v, \quad \mu_v, \sigma_v = f_v(x, \mathcal{P}); \\ \varepsilon_t &= \sigma_t \odot \epsilon_t + \mu_t, \quad \mu_t, \sigma_t = f_t(x, \mathcal{P}). \end{aligned} \tag{14}$$

Details of Visual Noise Generator f_v As shown in Figure 3(b), the input to the visual noise generator can be not only raw image x and prompts $\mathcal{P}$ but also their features encoded by CLIP, including text feature, image spatial feature, and image global feature. In Figures 3(c)-(e), we introduce three architectures for f_v based on MLP, CNN and Cross Attention (Vaswani et al. 2017), which can estimate μ and σ at either the raw image level or the feature level.

Details of Text Noise Generator f_t When estimating μ_t and σ_t from $f_t(x, \mathcal{P})$, each image x corresponds to a noise distribution over prompts. Thus, the prompt set with added noise should be image-instanced. Encoding these image-instanced prompts will cause a considerable computational burden. Conversely, for the classical CLIP based classification in Figure 3(a), the prompt set is shared across all images. Consequently, the text noise generator can only take text features as input, ignoring the image x. MLP can serve as this function $f_t(\mathcal{P})$.

In another implementation, we set μ_t, σ_t as learnable embeddings. Since μ_t and σ_t have no functional relationship with x and $\mathcal{P}$, $f_t(x, \mathcal{P})$ is effectively a constant function.

4 Experiments

4.1 Benchmark Settings

Datasets To evaluate the performance of PiNI, 11 datasets covering a wide range of visual concepts are selected. They include two generic object datasets, ImageNet (Deng et al. 2009) and Caltech101 (Fei-Fei, Fergus, and Perona 2004); five fine-grained datasets, OxfordPets (Parkhi et al. 2012), StanfordCars (Krause et al. 2013), Flowers102 (Nilsback and Zisserman 2008), Food101 (Bossard, Guillaumin, and Van Gool 2014) and FGVCAircraft (Maji et al. 2013), which contain fine-grained categories of pets, cars, flowers, food and aircraft, respectively. The other datasets are scene recognition dataset SUN397 (Xiao et al. 2010), action recognition dataset UCF101 (Soomro, Zamir, and Shah 2012), describable textures dataset DTD (Cimpoi et al. 2014) and EuroSAT (Helber et al. 2019) which contains satellite images. These datasets are abbreviated as Net, Caltech, Pets, Cars, Flowers, Food, Air, SUN, UCF, DTD and SAT.

Baseline Methods PiNI is compared with four baseline methods: Zero-shot CLIP (ZS) (Radford et al. 2021), Linear Probe (LP), CoOp (Zhou et al. 2022b), and CLIP-Adapter

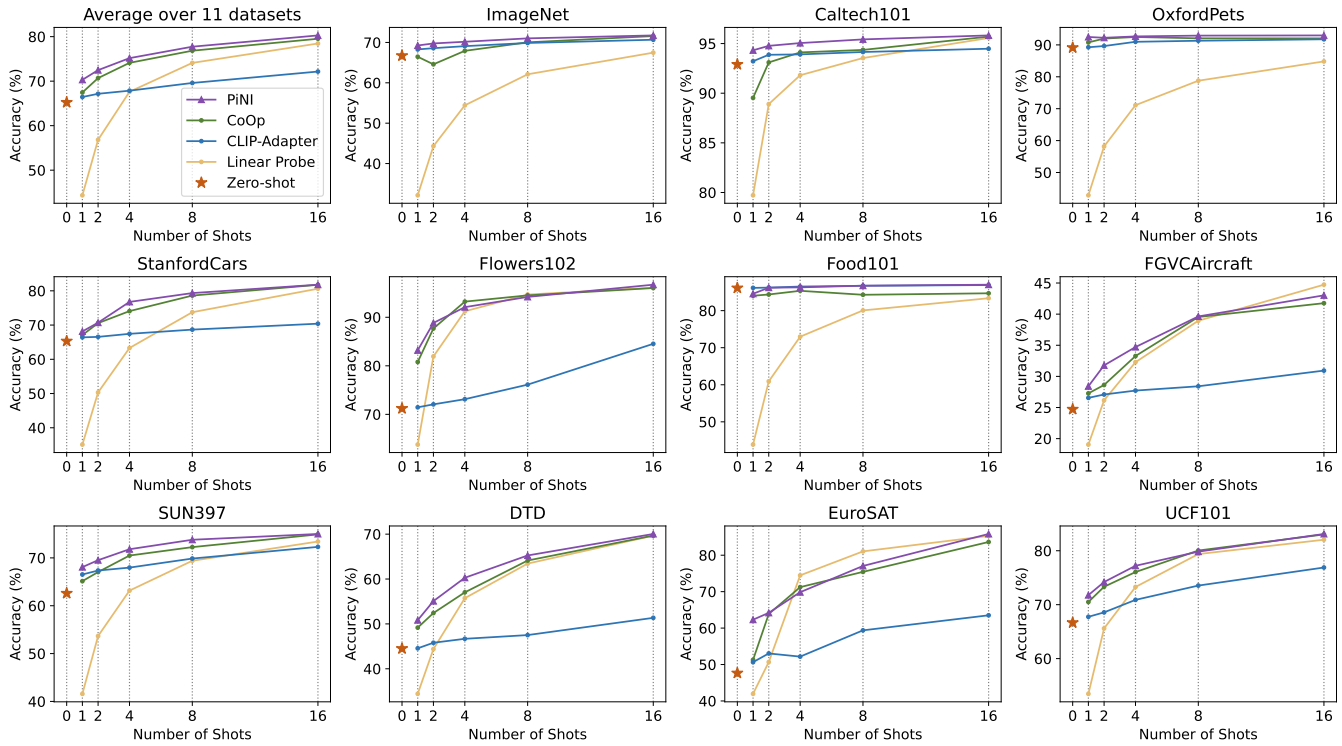

Figure 5: Performance of few-shot learning across 11 datasets. In the top-left subplot, the results are averaged over 11 datasets. PiNI shows better performance compared to baselines, especially under conditions with fewer shots.

Loc.	Net.	ImageNet	Caltech	Pets	Food
Image	MLP	67.43	94.52	90.92	85.39
	CNN	67.54	93.67	89.51	86.03
	CA	67.19	94.36	90.48	85.12
Visual Feature	MLP	69.51	95.41	**92.25**	85.91
	CA	**70.93**	**95.82**	91.99	**86.12**
PE	LE	**71.64**	**95.61**	**92.69**	86.98
Text Feature	LE	70.74	95.45	92.25	**87.01**
	MLP	71.14	95.13	91.44	85.29

Table 1: Performance of PiNI adopting different injection locations and network architectures. The number of shots is 16. PE is short for prompt embedding; CA is short for Cross-attention; LE is short for learnable embedding. Visual Feature + CA and PE + LE show the best performance at the visual and text ends, respectively.

Shot	Linear Probe	CoOp (IJCV'22)	CLIP-Adapter (IJCV'24)	PiNI (Ours)
0	65.23	65.23	65.23	65.23
1	44.36	67.46	66.44	**70.30**
2	56.82	70.69	67.16	**72.47**
4	67.61	74.11	67.86	**75.17**
8	74.10	76.84	69.59	**77.74**
16	78.46	79.53	72.16	**80.28**

Table 2: The performance averaged across 11 datasets under different shot settings.

(CAda) (Gao et al. 2024b). For Linear Probe, logistic regression is applied to classify the visual features of images. CoOp substitutes the word embeddings of prompt template with learnable parameters. CLIP-Adapter adds an extra module at the end of CLIP. The default parameter configurations are used for these baselines.

Training Details In the few-shot learning experiments, the train dataset is randomly sampled with 1, 2, 4, 8, and 16 shots per category. The model is tested on all data in the test dataset. To ensure the fairness of the experiment, the best-performing ViT-B/16 is selected as the visual encoder unless otherwise noted. The noise sample number m in Eq. (13) is set to 1. More details can be found in Appendix A.

4.2 Exploring the Framework of PiNI

Based on the discussion in Section 3.4, we explore different noise incorporation approaches and network architectures for noise distribution parameter estimation. We employ a 10-layer convolutional network with residual connections as the CNN architecture. For the cross-attention architecture, we input the text feature as the key and value. The query is the image's spatial feature when injecting into raw images, as shown in Figure 3(e). When visual features are injected,

Method	Source	Target				Avg.
	ImageNet	-V2	-S	-A	-R	
ZS	66.72	60.81	46.12	47.70	74.01	59.07
CoOp	71.58	63.76	46.34	48.21	73.18	60.61
CAda	70.67	63.35	47.42	**48.82**	73.94	60.84
PiNI	**71.74**	**64.40**	**48.25**	48.57	**74.39**	**61.47**

Table 3: Comparison of PiNI with other methods on robustness to distribution shift.

Dataset	Method	RN50	RN101	ViT-B/32	ViT-B/16
ImageNet	ZS	58.19	61.26	62.04	66.72
	CoOp	63.06	71.68	66.62	71.58
	CAda	63.20	70.65	65.81	70.66
	PiNI	**64.13**	**72.01**	**66.84**	**71.74**
Caltech	ZS	85.92	89.65	91.11	92.90
	CoOp	92.37	95.70	95.13	95.70
	CAda	90.50	94.48	93.55	94.48
	PiNI	**93.02**	**95.86**	**95.42**	**95.82**

Table 4: Performance of PiNI in different visual backbones.

the query is the feature itself.

Table 1 presents the performance of different combinations. For visual noise, injecting into visual feature with the cross-attention architecture yields the best performance. Regarding text noise, it is more effective to inject into the prompt embeddings using learnable embeddings as the generator. In the following experiments, we use these two combinations as visual and text noise generators as the default.

Visualization To understand how noise works, we visualize the noise injected into raw images, as shown in Figure 4. On the one hand, the injected noise can alleviate dataset bias, making objects in images resemble more common categories, such as the barrel in the first column and the gramophone in the third column. On the other hand, the noise can interfere with regions unrelated to the category, thereby simplifying the classification task. For instance, the vegetation and ball around the dog in the last column. Since the given prompt contains only a single category name, masking irrelevant regions in the image can enhance the alignment between the image and prompt embeddings.

4.3 Few-Shot Learning

The performance of different methods across 11 datasets is shown in Figure 5. From the figure, it is clear that PiNI has significant performance improvement. The results averaged across 11 datasets are provided in Table 2. It can be observed that the performance improvement is more pronounced with fewer shots.

The bar chart shown in Figure 6 illustrates the performance improvement of PiNI compared to Zero-shot CLIP. The maximum performance improvement is 38.18% on EuroSAT. On ImageNet, Caltech101, OxfordPets, and Food101, the performance improvements are relatively small as these categories are extensively contained in the corpus during pre-training stage.

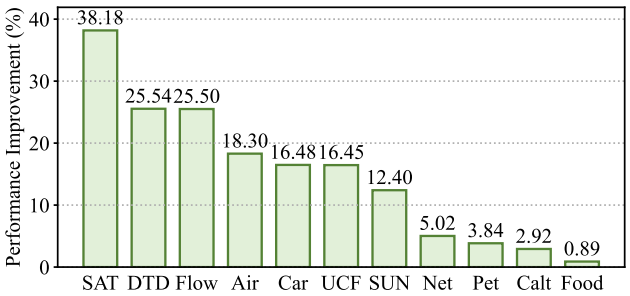

Figure 6: Performance improvement of PiNI compared to Zero-shot CLIP. The number of shots is 16.

Compared to Linear Probe, PiNI performs worse on FGVCAircraft. The aircraft names in FGVCAircraft are rarely included in the corpus of the pre-training phase, such as "Boeing 737". These specialized terms make prompt embeddings inaccurate in representing the fine-grained categories of aircraft. However, Linear Probe only utilizes the visual encoder for classification, which eliminates the interference of inaccurate prompt embeddings.

4.4 Domain Generalization

In domain generalization experiments, the model is trained on the source dataset ImageNet, and tested on target datasets. The target datasets are four variants of ImageNet, namely, ImageNetV2 (Recht et al. 2019), ImageNet-Sketch (Wang et al. 2019), ImageNet-A (Hendrycks et al. 2021b) and ImageNet-R (Hendrycks et al. 2021a). Detailed results are shown in Table 3. PiNI outperforms other methods except on ImageNet-A, exhibiting strong robustness against distribution shift.

4.5 Ablation on Visual Backbones

To investigate the impact of visual backbones in CLIP, we conduct 16-shot experiments, as shown in Table 4. These visual backbones include RN-50, RN-101, ViT-B/32, and ViT-B/16. On the two generic object datasets, ImageNet and Caltech, PiNI consistently demonstrates superior performance across different visual backbones.

5 Conclusion

In this work, we propose PiNI, a noise-based fine-tuning method towards vision-language alignment. It implies a new scheme to learn beneficial noise distribution. In our experiments, we demonstrate that PiNI outperforms existing methods. Our work can be further extended by refining the task entropy in Eq. (3). In this work, we select prompt-based image classification as the primary task. However, CLIP has a wide range of applications, such as Visual Question Answering (VQA), object detection, and image generation. Through defining task entropy on these tasks, the beneficial noise can be generated to simplify their complexity.

References

Bishop, C. M. 2006. Pattern recognition and machine learning. *Springer google schola*, 2: 645–678.

Blei, D. M.; Kucukelbir, A.; and McAuliffe, J. D. 2017. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518): 859–877.

Bossard, L.; Guillaumin, M.; and Van Gool, L. 2014. Food-101–mining discriminative components with random forests. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI 13*, 446–461. Springer.

Cimpoi, M.; Maji, S.; Kokkinos, I.; Mohamed, S.; and Vedaldi, A. 2014. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3606–3613.

Cubuk Ekin, D.; Barret, Z.; Dandelion, M.; Vijay, V.; and Le Quoc, V. A. 2019. Learning augmentation strategies from data. In *30th IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

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

Fei-Fei, L.; Fergus, R.; and Perona, P. 2004. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *2004 conference on computer vision and pattern recognition workshop*, 178–178. IEEE.

Gao, J.; Ruan, J.; Xiang, S.; Yu, Z.; Ji, K.; Xie, M.; Liu, T.; and Fu, Y. 2024a. Lamm: Label alignment for multi-modal prompt learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 1815–1823.

Gao, P.; Geng, S.; Zhang, R.; Ma, T.; Fang, R.; Zhang, Y.; Li, H.; and Qiao, Y. 2024b. Clip-adapter: Better vision-language models with feature adapters. *International Journal of Computer Vision*, 132(2): 581–595.

Guo, Z.; Zhang, R.; Qiu, L.; Ma, X.; Miao, X.; He, X.; and Cui, B. 2023. Calip: Zero-shot enhancement of clip with parameter-free attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 746–754.

Helber, P.; Bischke, B.; Dengel, A.; and Borth, D. 2019. 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.

Hendrycks, D.; Basart, S.; Mu, N.; Kadavath, S.; Wang, F.; Dorundo, E.; Desai, R.; Zhu, T.; Parajuli, S.; Guo, M.; et al. 2021a. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF international conference on computer vision*, 8340–8349.

Hendrycks, D.; Zhao, K.; Basart, S.; Steinhardt, J.; and Song, D. 2021b. Natural adversarial examples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 15262–15271.

Houlsby, N.; Giurgiu, A.; Jastrzebski, S.; Morrone, B.; De Laroussilhe, Q.; Gesmundo, A.; Attariyan, M.; and Gelly, S. 2019. adapterParameter-efficient transfer learning for NLP. In *International conference on machine learning*, 2790–2799. PMLR.

Hu, E. J.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W.; et al. 2021. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*.

Jia, M.; Tang, L.; Chen, B.-C.; Cardie, C.; Belongie, S.; Hariharan, B.; and Lim, S.-N. 2022. Visual prompt tuning. In *European Conference on Computer Vision*, 709–727. Springer.

Khattak, M. U.; Rasheed, H.; Maaz, M.; Khan, S.; and Khan, F. S. 2023. Maple: Multi-modal prompt learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 19113–19122.

Kingma, D. P.; and Welling, M. 2014. Auto-Encoding Variational Bayes. *stat*, 1050: 1.

Krause, J.; Stark, M.; Deng, J.; and Fei-Fei, L. 2013. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, 554–561.

Lester, B.; Al-Rfou, R.; and Constant, N. 2021. The Power of Scale for Parameter-Efficient Prompt Tuning. In Moens, M.-F.; Huang, X.; Specia, L.; and Yih, S. W.-t., eds., *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 3045–3059. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics.

Li, X. 2022. Positive-Incentive Noise. *IEEE Transactions on Neural Networks and Learning Systems*, 1–7.

Li, X.; Dai, Y.; Ge, Y.; Liu, J.; Shan, Y.; and Duan, L.-Y. 2022. Uncertainty modeling for out-of-distribution generalization. *arXiv preprint arXiv:2202.03958*.

Liu, H.; Li, C.; Li, Y.; and Lee, Y. J. 2024a. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 26296–26306.

Liu, H.; Li, C.; Wu, Q.; and Lee, Y. J. 2024b. Visual instruction tuning. *Advances in neural information processing systems*, 36.

Liu, Z.; and He, K. 2024. A Decade's Battle on Dataset Bias: Are We There Yet? *arXiv preprint arXiv:2403.08632*.

Maji, S.; Rahtu, E.; Kannala, J.; Blaschko, M.; and Vedaldi, A. 2013. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*.

Nilsback, M.-E.; and Zisserman, A. 2008. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*, 722–729. IEEE.

Parkhi, O. M.; Vedaldi, A.; Zisserman, A.; and Jawahar, C. 2012. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, 3498–3505. IEEE.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.;

et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, 8748–8763. PMLR.

Recht, B.; Roelofs, R.; Schmidt, L.; and Shankar, V. 2019. Do imagenet classifiers generalize to imagenet? In *International conference on machine learning*, 5389–5400. PMLR.

Soomro, K.; Zamir, A. R.; and Shah, M. 2012. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*.

Sung, Y.-L.; Cho, J.; and Bansal, M. 2022. Vl-adapter: Parameter-efficient transfer learning for vision-and-language tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5227–5237.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, H.; Ge, S.; Lipton, Z.; and Xing, E. P. 2019. Learning robust global representations by penalizing local predictive power. *Advances in Neural Information Processing Systems*, 32.

Xiao, J.; Hays, J.; Ehinger, K. A.; Oliva, A.; and Torralba, A. 2010. SUN database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 3485–3492. IEEE Computer Society.

Zaken, E. B.; Goldberg, Y.; and Ravfogel, S. 2022. BitFit: Simple Parameter-efficient Fine-tuning for Transformer-based Masked Language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 1–9.

Zhang, H.; Huang, S.; and Li, X. 2023. Variational positive-incentive noise: How noise benefits models. *arXiv preprint arXiv:2306.07651*.

Zhang, H.; Xu, Y.; Huang, S.; and Li, X. 2024. Data Augmentation of Contrastive Learning is Estimating Positive-incentive Noise. *arXiv preprint arXiv:2408.09929*.

Zhang, H.; Zhu, Y.; and Li, X. 2024. Decouple Graph Neural Networks: Train Multiple Simple GNNs Simultaneously Instead of One. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Zhang, Y.; Deng, B.; Li, R.; Jia, K.; and Zhang, L. 2023. Adversarial style augmentation for domain generalization. *arXiv preprint arXiv:2301.12643*.

Zhong, Z.; Zhao, Y.; Lee, G. H.; and Sebe, N. 2022. Adversarial style augmentation for domain generalized urban-scene segmentation. *Advances in neural information processing systems*, 35: 338–350.

Zhou, K.; Yang, J.; Loy, C. C.; and Liu, Z. 2022a. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 16816–16825.

Zhou, K.; Yang, J.; Loy, C. C.; and Liu, Z. 2022b. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9): 2337–2348.

A Additional Implementation Details

A.1 Datasets

We use 11 datasets for few-shot learning experiments and four datasets for domain generation experiments. In the domain generation experiments, we evaluate the model only on the four datasets with distribution shift, thus ignoring the training sets. Generally, we follow the same dataset processing procedures as those in CoOp (Zhou et al. 2022b). Details of these datasets are provided in Table 6.

A.2 Prompts

We use the default prompts from (Radford et al. 2021) for our experiments. For datasets of generic objects or scenes, the typical form "A photo of a [class]." is adopted. For other datasets in specific domains, additional context is added to the prompts. For instance, the prompt in OxfordFlowers is "A photo of a [class], a type of flower." The corresponding prompts are shown in Table 6.

The prompt that we select for PiNI differs slightly from hand-crafted prompts in CLIP (Radford et al. 2021) and CLIP-Adapter (Gao et al. 2024b). The difference lies in placing all the category names at the end of the template to avoid splitting the prompt template into two parts. This approach simplifies adding noise into the hand-crafted prompts. Additionally, we insert random words at the beginning of the prompt to ensure the same number of learnable word embeddings as CoOp.

A.3 Training

In all experiments, a batch size of 32 is used. For ImageNet containing 1000 categories, the number of epochs is set to 100, while for the other datasets, it is set to 200. The SGD optimizer with a Cosine Annealing Learning Rate is employed. For ImageNet, the initial learning rate is 0.001. For other datasets, the learning rate is between 0.001 to 0.003. We also apply a constant warm-up phase with a learning rate of 1e-5, which lasts for 1 epoch. All experiments are conducted using a single NVIDIA 4090 GPU.

B Additional Results

B.1 Few-Shot Learning

We provide detailed few-shot learning results for 11 datasets in Table 7. In this table, PiNI is compared with Zero-shot CLIP, Linear Probe, CoOp, and CLIP-Adapter in the 16-shot learning. In Table 5, we apply ViT-L-14@336px to PiNI and other baselines. PiNI also enhances the performance of strong CLIP models, particularly when there are very few training samples.

B.2 Visualization of Noise

We visualize the noise injected into raw images from Caltech101, Food101, and OxfordPets in Figures 7, 8, and 9, respectively. The first row shows the raw images. The second row displays the noise-injected images. The third and fourth rows present heatmaps of the mean μ and variance σ for each pixel, respectively.

Dataset	Method	shot=2	shot=8	shot=16
Caltech	LP	91.52	95.05	96.31
	CoOp	95.82	96.43	97.12
	CAda	94.68	96.10	96.63
	PiNI	**96.27**	**96.87**	**97.28**
DTD	LP	42.85	65.13	72.1
	CoOp	58.03	**70.74**	**73.75**
	CAda	56.5	59.87	64.36
	PiNI	**59.39**	70.56	73.58
OxfordPets	LP	66.53	84.08	89.56
	CoOp	93.67	94.19	94.35
	CAda	94.03	94.33	94.52
	PiNI	**94.41**	**95.01**	**95.12**
SUN397	LP	56.02	71.34	74.92
	CoOp	69.44	76.35	**78.40**
	CAda	71.79	74.85	77.01
	PiNI	**73.22**	**76.74**	78.16

Table 5: Comparison of PiNI with other methods using the visual backbone of ViT-L-14@336px.

B.3 Interpretation of the Learned Prompt Distribution

In our proposed method PiNI, random noise is injected into the word embeddings of the prompt template, so the template itself also follows a probability distribution. However, interpreting the learned distribution of word embeddings is challenging, because they exist in continuous space, whereas the embeddings of real words are in discrete space. We address this issue by finding the nearest embeddings of real words in the vocabulary. Specifically, we sample the noise from the distribution 100 times and add it to the prompt template. For each sample, we record five words nearest to the word embedding at each position of the prompt template. Table 8 shows the most frequently occurring words at each position of the prompt template. Note that the vocabulary includes many subwords, e.g., "pic" is a subword of "picture". By adding noise to the prompt, we find that the breadth of semantics expressed by the words in the prompt template has increased. For example, the words "doing", "do", and "does" express the same meaning in different grammatical contexts. Additionally, "photo", "photos", "pic" and "picture" also have similar meanings. In conclusion, these words with similar meanings demonstrate that the obtained prompt distribution has richer semantics, confirming the effectiveness of our proposed PiNI.

Dataset	Classes	Train Size	Test Size	Prompt
ImageNet	1,000	1,281,167	50,000	"a photo of a [class]"
Caltech101	100	4,128	2,465	"a photo of a [class]"
OxfordPets	37	2,944	3,669	"a type of pet, a photo of a [class]"
StanfordCars	196	6,509	8,041	"a photo of a [class]"
Flowers102	102	4,093	2,463	"a type of flower, a photo of a [class]"
Food101	101	50,500	30,300	"a type of food, a photo of [class]"
FGVCAircraft	100	3,334	3,333	"a type of aircraft, a photo of a [class]"
SUN397	397	15,880	19,850	"a photo of a [class]"
DTD	47	2,820	1,692	"texture [class]"
EuroSAT	10	13,500	8,100	"a centered satellite photo of [class]"
UCF101	101	7,639	3,783	"a photo of a person doing [class]"
ImageNetV2	1,000	–	10,000	"a photo of a [class]"
ImageNet-Sketch	1,000	–	50,889	"a photo of a [class]"
ImageNet-A	200	–	7,500	"a photo of a [class]"
ImageNet-R	200	–	30,000	"a photo of a [class]"

Table 6: The details of the datasets and their corresponding prompts.

Method	ImageNet	Caltech	Pets	Cars	Flowers	Food	Aircraft	SUN	DTD	SAT	UCF
Zero-shot	66.72	92.90	89.13	65.31	71.21	86.11	24.72	62.60	44.50	47.65	66.66
Linear Probe	67.47	95.56	84.82	80.62	96.10	83.35	**44.73**	73.42	69.66	85.30	82.04
CoOp	71.58	95.70	92.04	81.76	96.02	84.68	41.76	74.91	69.68	83.63	83.02
CLIP-Adapter	70.67	94.48	91.76	70.40	84.53	86.90	30.93	72.30	51.35	63.51	76.89
PiNI	**71.74**	**95.82**	**92.97**	**81.79**	**96.71**	**87.00**	43.02	**75.00**	**70.04**	**85.83**	**83.11**

Table 7: Comparison of PiNI with other methods in 16-shot learning. The names of these datasets are abbreviated.

Dataset	Position	1	2	3	4	5
OxfordPets	1	a(100)	an (99)	the (87)	my (54)	sundaywithmarsha (46)
	2	type(100)	0 (83)	types (79)	1 (51)	pational (30)
	3	of(100)	in (65)	to (63)	on (52)	0 (51)
	4	pet (100)	0 (99)	pets (75)	1 (45)	2 (39)
	5	, (100)	., (60)	), (57)	. (30)	! (30)
	6	a (100)	an (94)	flyeagles (50)	the (50)	0 (34)
	7	photo (100)	0 (84)	1 (65)	2 (45)	coscino (44)
	8	of (100)	to (100)	by (49)	in (49)	with (40)
	9	a (100)	an (79)	sundaywithmarsha (51)	0 (47)	1 (28)
UCF101	1	a (100)	an (84)	0 (81)	the (56)	1 (55)
	2	photo (100)	photos (96)	pic (75)	0 (74)	picture (39)
	3	of (100)	to (86)	for (55)	, (52)	at (28)
	4	a (100)	an (58)	the (58)	instaweather (53)	0 (49)
	5	person (100)	0 (75)	people (57)	1 (53)	pational (35)
	6	doing (100)	0 (69)	do (63)	does (39)	1 (39)

Table 8: The most frequently occurring words at each position in the prompt templates in OxfordPets and UCF101. The frequency of occurrence for each word is provided in parentheses.

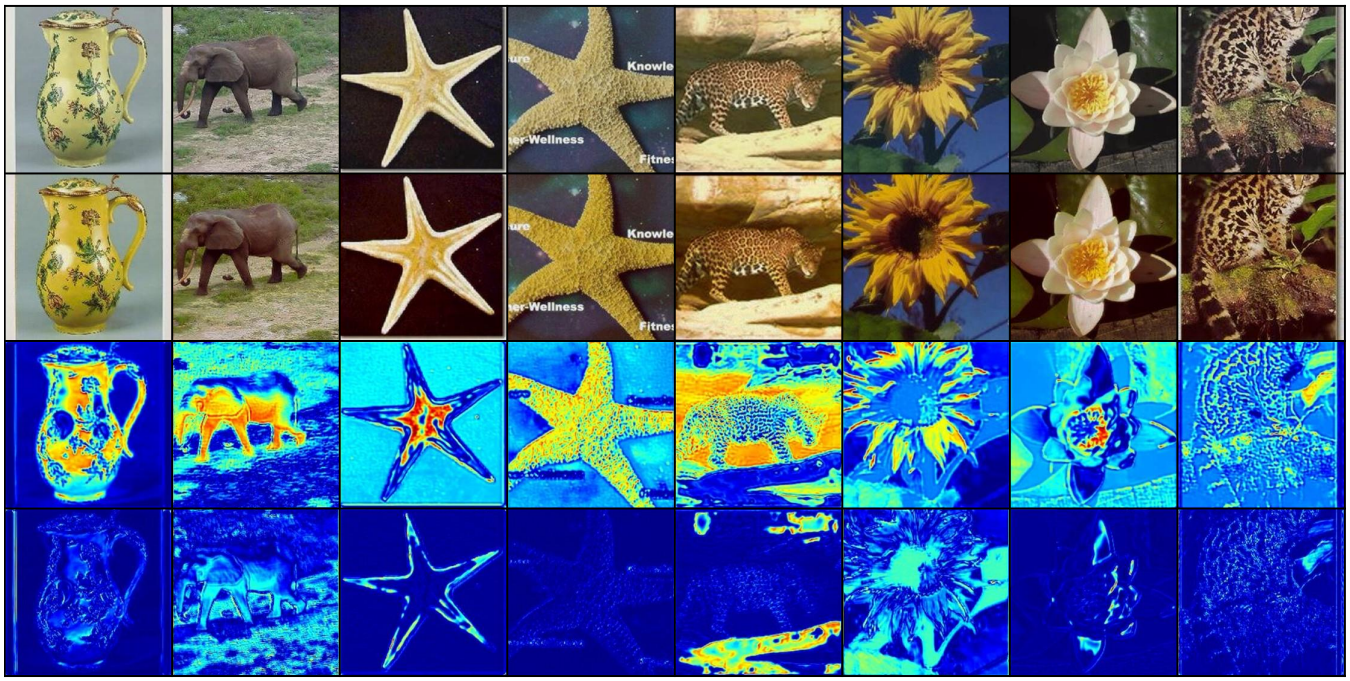

Figure 7: Visualization of the noise injected into raw images from Caltech101.

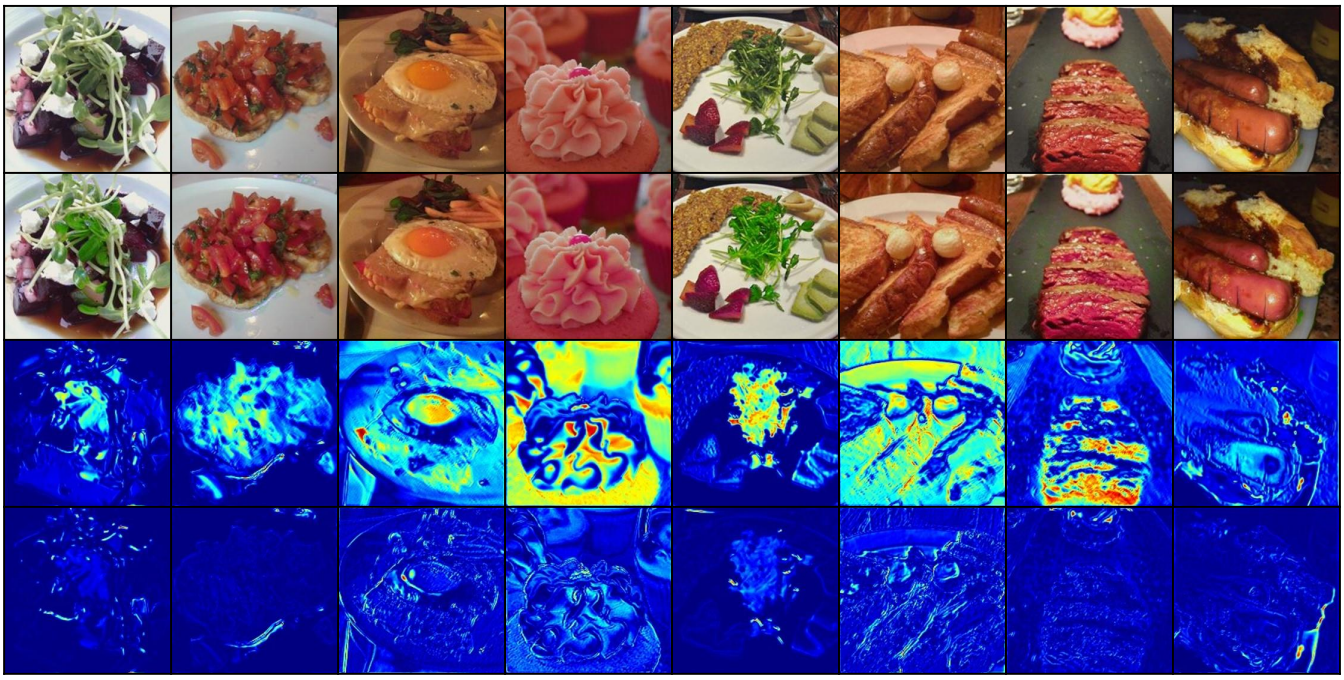

Figure 8: Visualization of the noise injected into raw images from Food101.

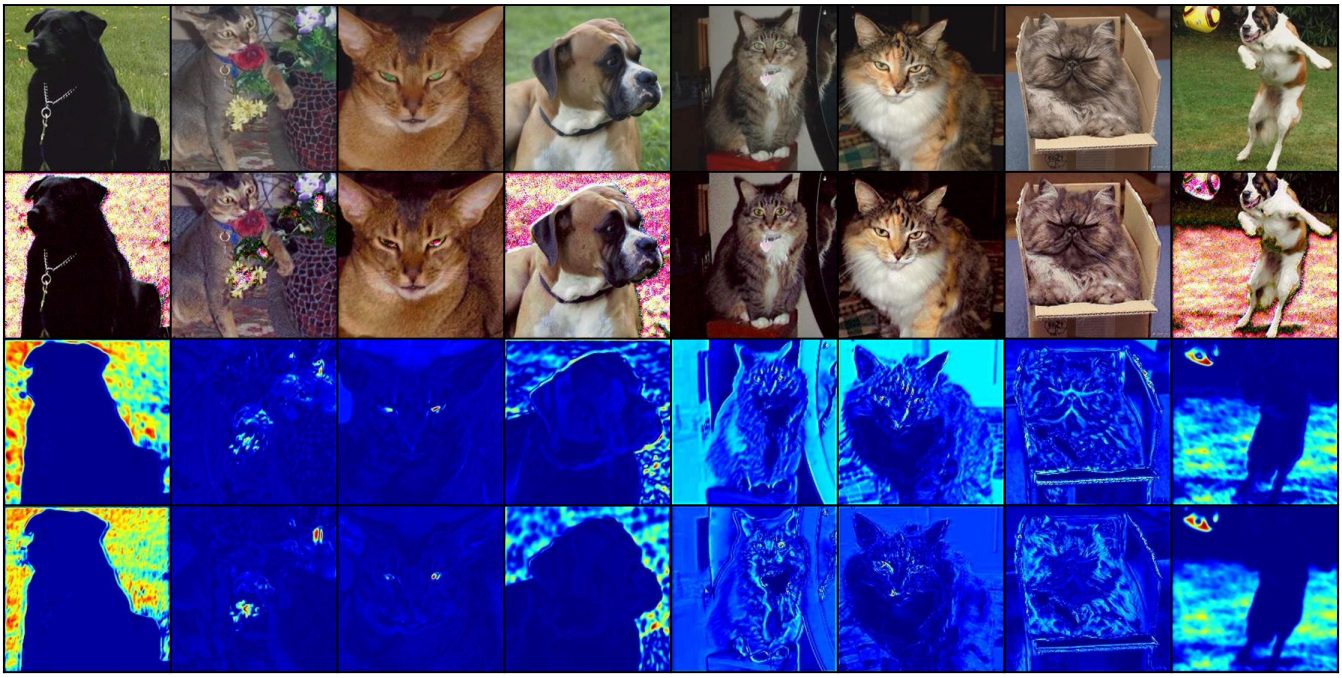

Figure 9: Visualization of the noise injected into raw images from OxfordPets.