Improving Zero-Shot Generalization for CLIP with Synthesized Prompts

Zhengbo Wang[1,2], Jian Liang[2,3*], Ran He[2,3], Nan Xu[5], Zilei Wang[1], and Tieniu Tan[2,3,4]
[1] University of Science and Technology of China
[2] CRIPAC & MAIS, Institute of Automation, Chinese Academy of Sciences
[3] University of Chinese Academy of Sciences [4] Nanjing University [5] Beijing Wenge Group
zhengbowang@mail.ustc.edu.cn, liangjian92@gmail.com

Abstract

*With the growing interest in pretrained vision-language models like CLIP, recent research has focused on adapting these models to downstream tasks. Despite achieving promising results, most existing methods require labeled data for all classes, which may not hold in real-world applications due to the long tail and Zipf's law. For example, some classes may lack labeled data entirely, such as emerging concepts. To address this problem, we propose a plug-and-play generative approach called **SyntHesIzed Prompts (SHIP)** to improve existing fine-tuning methods. Specifically, we follow variational autoencoders to introduce a generator that reconstructs the visual features by inputting the synthesized prompts and the corresponding class names to the textual encoder of CLIP. In this manner, we easily obtain the synthesized features for the remaining label-only classes. Thereafter, we fine-tune CLIP with off-the-shelf methods by combining labeled and synthesized features. Extensive experiments on base-to-new generalization, cross-dataset transfer learning, and generalized zero-shot learning demonstrate the superiority of our approach. The code is available at https://github.com/mrflogs/SHIP.*

1. Introduction

In recent years, language-supervised vision pretrained models have garnered much attention. By establishing a link between images and natural language, these models exhibit impressive zero-shot capabilities and remarkable transfer ability [35, 18, 1, 26, 4, 42], demonstrating potential in learning open-world concepts. One of the most successful large-scale pretrained vision-language models is CLIP [35]. By leveraging a massive dataset of 400 million image-text pairs, it learns to align visual and textual representations from a vision encoder and a language encoder, respectively. After pretraining, CLIP [35] can perform zero-shot recognition by merely providing the class names. The classification weights are generated by the language encoder through prompting [27]. For instance, we can adopt a prompt template like "a photo of a {class}" as the input of the text encoder, and then the weights for classification can be synthesized by substituting in the "{class}" with the actual class name. And the resulting classification score is the cosine similarity between the image features and the weights.

To further enhance the performance of CLIP, several previous works have proposed the use of learnable prompts [53, 52, 10, 29] or adapters [51, 14] to fine-tune the pretrained CLIP to specific downstream tasks. These methods have achieved significant improvements with only a small amount of labeled data from downstream datasets, which clearly demonstrates their superiority in terms of data efficiency. However, a significant limitation of these methods is their reliance on having data available for all classes, which can be impractical in real-world applications. The issue arises due to Zipf's law and the long tail phenomenon, which make it challenging to collect data for rare categories, such as new species or emerging concepts. As a result, many categories may be devoid of any relevant data, rendering previous methods either invalid [41, 51] for such scenarios or leading to a significant drop in performance on the label-only classes [53], compared to zero-shot CLIP. To address this limitation, our goal is to develop a fine-tuning approach that can effectively recognize both categories with and without available data while maintaining the superior data efficiency of previous methods.

In this paper, we propose a plug-and-play generative approach called **SyntHesIzed Prompts** (**SHIP**) to improve existing fine-tuning methods. The main objective is to train a generative model that can synthesize features by providing class names, which enables us to generate features for categories without data. And we proceed to fine-tune CLIP using both the original labeled and the newly synthesized features with off-the-shelf methods. However,

*To whom correspondence should be addressed.

a major obstacle is that generative models typically require a substantial amount of data to train, which contradicts our goal of data efficiency. We propose to utilize variational autoencoder [23] (VAE) as the framework, which is easier to train and more effective in low-data scenarios compared to models that require adversarial training [2, 15]. Additionally, inspired by previous prompt learning methods [53, 52, 10, 29], we train the generator to produce prompts instead of visual features. We then feed these prompts and corresponding class names into the frozen CLIP language encoder to obtain synthesized features. Since CLIP has been pretrained on a large-scale dataset and has aligned visual and language representations, we believe that the pretrained language encoder aids in generating more realistic features.

In summary, this paper aims to address the issue of downstream tasks where some classes have no relevant data while maintaining the superior data efficiency of previous methods. To achieve this goal, we propose a novel generative approach named SHIP, which can synthesize features for categories without data based solely on their class names. Notably, our proposed generative method is orthogonal to CLIP fine-tuning methods and can enhance their performance by utilizing synthesized data. We conduct comprehensive experiments on base-to-new generalization, cross-dataset transfer learning, and generalized zero-shot learning, resulting in state-of-the-art performance.

2. Related Work

Vision-Language Pretraining. Vision-language pretraining models (VLMs) investigate the relationship between vision and language modalities. Various methods have been proposed to establish this connection through self-supervised learning, such as masked language model [22, 28], masked region prediction [39, 38] and image-text matching [39, 22]. Recently, contrastive learning-based VLMs have shown remarkable performance by utilizing large-scale noisy image-text pairs. These methods, including CLIP [35] and ALIGN [18], learn aligned representations of images and text via the contrastive loss, which pulls the representations of matching image-text pairs together and pushes those of mismatching pairs apart. Based on natural language supervision, these VLMs acquire transferable visual representations and exhibit impressive zero-shot performance on various image classification tasks.

Fine-tuning for VLMs. Inspired by the prior work in NLP, recent researches focus on developing efficient fine-tuning methods for VLMs on downstream tasks. One type of such method is prompt tuning, which has been explored in several recent works [53, 29, 5]. CoOp [53] proposes a prompt learning method that optimizes a class-agnostic prompt template in the continuous token embedding space through back-propagation on few-shot datasets. ProDA [29] attempts to learn a collection of continuous prompts to capture the variational visual representation. PLOT [5] proposes to apply optimal transport to match the learnable prompts with different areas of the images. Another type of fine-tuning method is adapters [14, 51]. CLIP-Adapter [14] proposes to add a lightweight MLP following the last vision layer and mix the output feature with the original zero-shot feature via a residual connection. Tip-Adapter [51] further improves CLIP-Adapter [14] by replacing the lightweight MLP with a linear layer, whose weights are comprised of the labeled visual embeddings, acting as visual prototypes of the concepts. This not only inherits the training-free advantage of zero-shot CLIP [35] but also performs comparably to those training-required approaches.

While these methods have achieved significant improvements on downstream datasets, they require data for all classes when fine-tuning. When dealing with new unseen classes, they either become invalid [51] or their performance drops dramatically [53]. However, some classes are difficult to collect data for because of their rareness, such as new species or concepts. As a result, many categories may be devoid of any relevant data. To address this, previous methods have attempted to learn more robust prompts. CoCoOp [52] improves new class performance by learning an instance-specific continuous prompt conditioned on the input image. With image information, the prompts are easily transferred to recognize new class samples. VPT [10] proposes to learn the distribution of instance-specific prompts via variational inference. During inference, VPT ensembles several prompts sampled from the distribution for the classification. In contrast to the previous methods [52, 10], we propose to synthesize features for those unseen categories. With features for all classes, we can utilize off-the-shelf methods to fine-tune CLIP.

Generalized Zero-Shot Learning. Generalized zero-shot learning (GZSL) is a relevant research field with similar objectives to our work. Specifically, GZSL focuses on training a classifier that can recognize both seen and unseen object classes, where the latter is absent from the training set. To accomplish this, GZSL leverages auxiliary semantic information such as expert annotated attributes or text descriptions [31] for both seen and unseen classes. Embedding-based GZSL methods aim to learn a visual-to-semantic mapping for visual-semantic interaction by mapping visual features into the semantic space [49, 48]. However, a major drawback of these methods is their bias towards seen classes, as they only learn from seen data. As a solution, generative-based GZSL methods have been introduced to learn semantic-to-visual mapping to generate visual features of unseen classes [45, 46, 25, 32] for data augmentation. Currently, the generative methods are typically based on variational autoencoders (VAEs) [23, 46], generative adversarial networks (GANs) [45, 46, 25, 13], and gen-

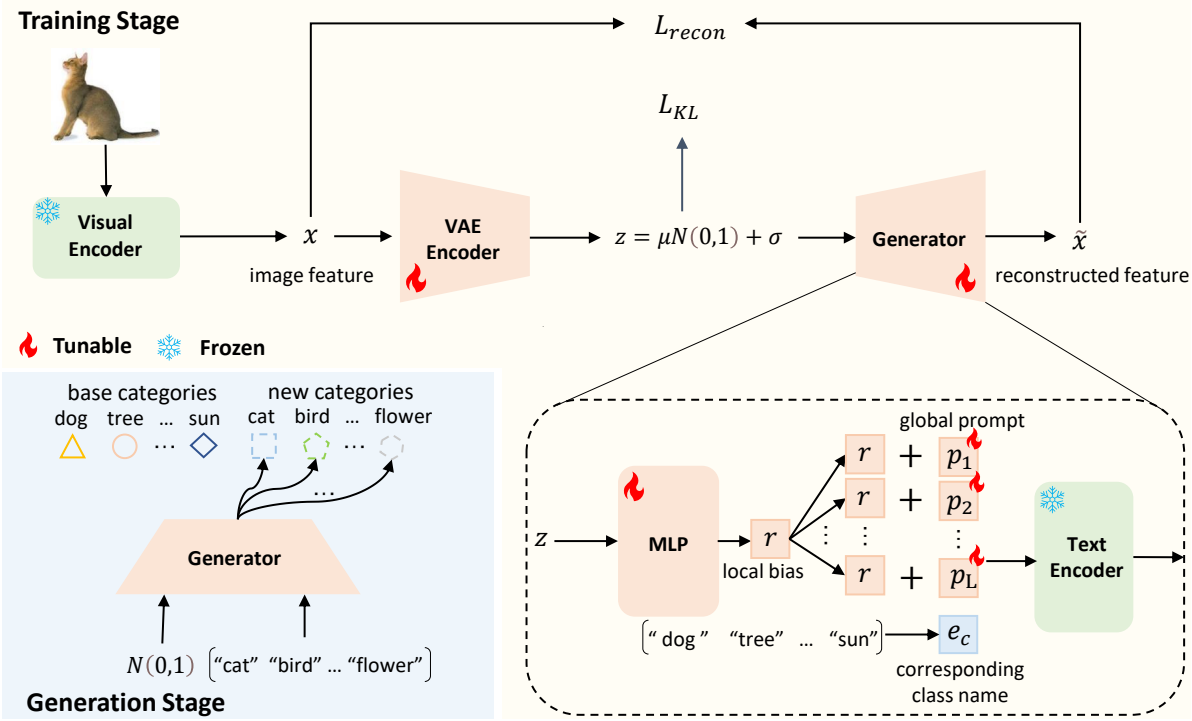


Figure 1. The proposed model architecture is built upon the VAE framework, comprising a VAE encoder and a generator. In the training stage, we extract the image feature with CLIP visual encoder and the VAE encoder encodes it into a latent code z, which is subsequently confined to a prior distribution. Following this, the generator reconstructs the input feature by utilizing the encoded information. Notably, a novel CLIP-based generator is introduced, which comprises two subnetworks: a lightweight MLP and a frozen CLIP text encoder. The MLP transforms the latent code z into a local bias, which is subsequently added to global learnable prompt vectors to construct the final prompts. The prompts, together with the class names, are then input into the frozen text encoder to obtain the reconstructed feature. During the generating stage, we sample the latent code from the prior distribution and then use it with the new class name to synthesize the corresponding features. Finally, we fine-tune CLIP using off-the-shelf methods with the base class and synthetic new class features.

erative flows [36]. Despite their promising results, these generative-based methods require training on a large seen dataset to learn semantic-visual mapping and expertly annotated attributes or text descriptions for all classes, which can be labor-intensive. In our work, we aim to imitate GZSL by learning to synthesize samples for new classes. However, with limited labeled data in the training set and coarse semantic vectors for each class through prompting like "a photo of a {class}", these GZSL generative methods fail to synthesize valid new samples for new classes.

3. Method

3.1. Background

Contrastive Language-Image Pretraining, known as CLIP [35], is a method developed for aligning the representations of images and their corresponding captions, which has gained considerable attention in recent years. CLIP consists of two encoder modules: a visual encoder $\mathcal{I}(\boldsymbol{x})$ and a language encoder $\mathcal{T}(t)$, which encode images and text descriptions, respectively, into a shared d-dimensional space. The visual encoder can be ViT [11] or ResNet [16], while the language encoder is a Transformer [40]. Both encoders are trained jointly using a contrastive loss applied to a large dataset of paired images and captions. Once trained, CLIP can be used for zero-shot classification of downstream tasks. To perform C-class image classification, category descriptions $\{t_c\}_{c=1}^{C}$ are generated through prompting, such as "a photo of a {class}". Then, the classification probability of the input image $\boldsymbol{x}$ is computed as follows:

$$p(y|\boldsymbol{x}) = \frac{\exp(\cos(\mathcal{I}(\boldsymbol{x}), \mathcal{T}(t_y))/\tau)}{\sum_{c=1}^{C} \exp(\cos(\mathcal{I}(\boldsymbol{x}), \mathcal{T}(t_c))/\tau)}, \quad (1)$$

where τ denotes the temperature, $\cos(\cdot,\cdot)$ is the cosine similarity function, and y is the target class.

3.2. Synthesized Prompts

In this paper, we aim to improve the performance of CLIP on both base and new categories, i.e., categories with and without available data, while maintaining data efficiency as previous methods. To achieve this goal, a novel generative approach named **Synt**H**es**I**z**ed **P**rompts (**SHIP**) is proposed, which involves three stages. First, we follow

variational autoencoders to introduce a generator that reconstructs the visual features by inputting the synthesized prompts and the corresponding class names to the language encoder of CLIP. Subsequently, we obtain the synthesized features for new categories by providing the class names. Finally, we combine the labeled base class features with the synthesized new class features and employ existing fine-tuning methods, such as CoOp [53] and Tip-Adapter [51], to fine-tune CLIP, which thus enhances its performance on both base and new classes.

The architecture of the generative model is illustrated in Figure 1. To maintain the data efficiency, we opt to employ the Variational Autoencoder (VAE) [23] for training our generator instead of Generative Adversarial Networks (GANs) [15]. The reason is that it is difficult to train an effective discriminator for GANs with limited labeled data [21]. As depicted in Figure 1, the VAE architecture comprises an encoder $E(x)$ and a generator $G(z, c)$. First, we leverage the fixed CLIP visual encoder to extract the feature of the input image, i.e., $x = \mathcal{I}(img)$. Subsequently, the VAE encoder $E(x)$ encodes the feature x into a latent code z, and the generator $G(z, c)$ reconstructs the feature x using the latent code z and the corresponding class name c. The optimization of both E and G is achieved via the evidence-lower bound given by the equation as follows:

$$\begin{aligned} L &= L_{recon} + L_{KL} \\ &= \mathbb{E}_{\tilde{x}\sim p(G(z,c))}[-\log p(\tilde{x})] + \mathbb{E}_{x\sim p(x)}[D_{KL}(p(x)\|p(z|c))], \end{aligned} \tag{2}$$

where D_{KL} represents the Kullback-Leibler divergence, $p(z|c)$ is a prior distribution that is assumed to be $\mathcal{N}(0, 1)$, and $-\log p(\tilde{x})$ denotes the reconstruction loss.

To further utilize the pretrained knowledge of CLIP, we propose a CLIP-based generator. Notably, the pretrained CLIP has learned aligned vision and language representations, allowing us to reconstruct input features from the language encoder $\mathcal{T}$. Since having been trained on a large-scale dataset, the reconstructed features obtained from the pretrained language model $\mathcal{T}$ are expected to be of higher quality than those generated by a new generator trained from scratch on the few-shot base dataset. Drawing inspiration from previous prompt learning methods [53, 52, 29], we generate instance-specific prompts, instead of generating the features directly. Specifically, given the latent code z, we generate instance-specific prompts as follows:

$$\boldsymbol{p}(z) = [\boldsymbol{p}_1 + \boldsymbol{r}, \boldsymbol{p}_2 + \boldsymbol{r}, ..., \boldsymbol{p}_L + \boldsymbol{r}], \tag{3}$$

where the local bias $\boldsymbol{r}$ is obtained through a two-layer fully-connected network, i.e., $\boldsymbol{r} = MLP(z)$, that embeds the latent code z into the token embedding space, and L is the length of prompts. As in Eq. (3), our prompts consist of two components: a global fixed set of learnable prompts $\{\boldsymbol{p}_i, i = 1, 2, ..., L\}$, which are randomly initialized, capturing the global information of the input features and a local bias $\boldsymbol{r}$ that encodes the instance-specific information of the input feature into the prompts. By combining the prompts and the token embedding of the corresponding class name, we obtain the reconstructed features as follows:

$$\tilde{\boldsymbol{x}} = \mathcal{T}(t), \quad t = \{\boldsymbol{p}(z), \boldsymbol{e_c}\}, \tag{4}$$

where $\mathcal{T}$ is the frozen language encoder, and $\boldsymbol{e_c}$ is the token embedding of the corresponding class names.

During the training stage, we maintain the CLIP frozen and only optimize the encoder E, the lightweight MLP, and the global prompts $\boldsymbol{p} = [\boldsymbol{p_1}, \boldsymbol{p_2}, ..., \boldsymbol{p_L}]$.

3.3. Fine-tuning CLIP

Following the training stage, the generator is employed to synthesize features for new classes. Specifically, given the class name c of a new class and the noise z sampled from the prior distribution, the generator $G(z, c)$ is utilized to generate the corresponding features. This process is repeated for each new class, resulting in a new synthetic dataset. When combined with the labeled base dataset, a complete dataset for all classes is obtained. Consequently, off-the-shelf methods [53, 14, 51, 5] can be employed to fine-tune CLIP, which is expected to perform better on new classes in comparison to its previous counterparts.

4. Experiments

4.1. Setup

We evaluate our method for three different tasks: base-to-new generalization, cross-dataset transfer, and generalized zero-shot classification. For the base-to-new generalization and cross-dataset transfer tasks, we follow the same experimental setting as CoCoOp [52]. It uses a total of 11 diverse image classification datasets, i.e., ImageNet [9] and Caltech101 [12] for generic object recognition, OxfordPets [34], StanfordCars [24], Flowers102 [33], Food101 [3] and FGVCAircraft [30] for fine-grained image recognition, EuroSAT [17] for satellite image classification, UCF101 [37] for action classification, DTD [8] for texture classification, and SUN397 [47] for scene recognition. For generalized zero-shot classification tasks, we follow the same setting as [44], and we conduct the experiments on three standard zero-shot recognition datasets: Caltech-UCSD-Birds [43] (CUB), Oxford Flowers [33] (FLO), and Animals with Attributes2 [44] (AWA2), containing 200, 102, and 50 categories, respectively. For a fair comparison, we use the same data splits and evaluation protocols as proposed in [44].

Implementation details. Our proposed method is comprised of three sub-networks: a VAE encoder, a lightweight MLP, and a pretrained CLIP. The VAE encoder and the MLP are implemented as two-layer fully-connected networks with 4,096 hidden units and ReLU activation. And

Table 1. **Base-to-new generalization.** Our proposed model is trained on a few-shot training set (base) and then evaluated on both base and new classes. +SHIP denotes we add our method to previous off-the-shelf methods. The result of Tip-Adapter-F [51] is not included in the table due to its inability to test on new classes. The average accuracy of the base and new classes is represented by the terms **Base** and **New**, respectively, while their harmonic mean is denoted as **H**. The best results are presented in bold.

	Average			ImageNet [9]			Caltech101 [12]			OxfordPets [34]		
	Base	**New**	**H**	**Base**	**New**	**H**	**Base**	**New**	**H**	**Base**	**New**	**H**
CLIP [35]	69.34	74.22	71.70	72.43	68.14	70.22	96.84	94.00	95.40	91.17	97.26	94.12
CoOp [53]	82.69	63.22	71.66	76.47	67.88	71.92	98.00	89.81	93.73	93.67	95.29	94.47
CoCoOp [52]	80.47	71.69	75.83	75.98	70.43	73.10	97.96	93.81	95.84	95.20	97.69	96.43
ProDA [29]	81.56	72.30	76.65	75.40	70.23	72.72	98.27	93.23	95.68	**95.43**	97.83	**96.62**
CLIP-Adapter [29]	83.05	65.20	73.05	75.74	68.21	71.78	98.13	92.19	95.39	91.55	90.10	90.82
CoOp + VPT [10]	71.98	74.76	73.34	74.73	**70.60**	72.60	95.47	93.80	94.62	90.77	97.83	94.16
CoOp + SHIP	80.03	73.69	76.73	75.87	69.95	72.79	97.55	**95.20**	96.36	95.37	**97.87**	96.61
CLIP-Adapter + SHIP	83.14	67.77	74.67	76.00	69.32	72.51	97.68	95.09	**96.37**	92.19	93.85	93.01
Tip-Adapter-F + SHIP	**83.80**	**76.42**	**79.94**	**77.53**	70.26	**73.71**	**98.32**	94.43	96.34	94.95	97.09	96.01

	StanfordCars [24]			Flowers102 [33]			Food101 [3]			FGVCAircraft [30]		
	Base	**New**	**H**	**Base**	**New**	**H**	**Base**	**New**	**H**	**Base**	**New**	**H**
CLIP [35]	63.37	74.89	68.65	72.08	77.80	74.83	90.10	91.22	90.66	27.19	**36.29**	31.09
CoOp [53]	78.12	60.40	68.13	97.60	59.67	74.06	88.33	82.26	85.19	40.44	22.30	28.75
CoCoOp [52]	70.49	73.59	72.01	94.87	71.75	81.71	**90.70**	91.29	90.99	33.41	23.71	27.74
ProDA [29]	74.70	71.20	72.91	97.70	68.68	80.66	90.30	88.57	89.43	36.90	34.13	35.46
CLIP-Adapter [14]	79.16	59.49	67.93	**98.29**	64.68	78.02	88.24	88.33	88.29	42.14	25.67	31.91
CoOp + VPT [10]	65.27	**75.97**	70.21	72.97	75.90	74.40	90.37	**91.67**	91.01	29.57	33.80	31.54
CoOp + SHIP	68.57	73.90	71.14	94.02	74.40	83.06	90.54	91.03	90.78	34.27	32.33	33.28
CLIP-Adapter + SHIP	78.51	62.52	69.61	98.20	65.89	78.86	88.63	87.07	87.84	42.26	30.05	35.13
Tip-Adapter-F + SHIP	**79.91**	74.62	**77.18**	95.35	**77.87**	**85.73**	90.63	91.51	**91.07**	**42.62**	35.93	**38.99**

	SUN397 [47]			DTD [8]			EuroSAT [17]			UCF101 [37]		
	Base	**New**	**H**	**Base**	**New**	**H**	**Base**	**New**	**H**	**Base**	**New**	**H**
CLIP [35]	69.36	75.35	72.23	53.24	59.90	56.37	56.48	64.05	60.03	70.53	77.50	73.85
CoOp [53]	80.60	65.89	72.51	79.44	41.18	54.24	92.19	54.74	68.69	84.69	56.05	67.46
CoCoOp [52]	79.74	76.86	78.27	77.01	56.00	64.85	87.49	60.04	71.21	82.33	73.45	77.64
ProDA [29]	78.67	76.93	77.79	80.67	56.48	66.44	83.90	66.00	73.88	85.23	71.97	78.04
CLIP-Adapter [14]	79.44	66.81	72.58	**81.94**	39.49	53.30	**93.45**	54.41	68.78	85.42	67.77	75.58
CoOp + VPT [10]	73.77	**77.90**	75.77	57.67	58.70	58.18	67.97	71.63	69.75	73.23	74.63	73.92
CoOp + SHIP	79.54	75.27	77.35	74.88	56.88	64.65	88.62	66.87	76.22	81.08	76.85	78.91
CLIP-Adapter + SHIP	79.86	66.52	72.58	81.60	46.38	59.14	93.05	57.15	70.81	**86.61**	71.61	78.40
Tip-Adapter-F + SHIP	**81.32**	77.64	**79.43**	81.83	**61.47**	**70.21**	93.38	**81.67**	**87.13**	85.99	**78.10**	**81.85**

we employ ViT-B/16 [11] and transformer [40] as the vision and language encoders of CLIP, which are initialized with CLIP's pretrained weights and kept frozen during training. The dimensions of the latent code z are set to be equal to the dimension of token embedding. We fix the length of the learnable global context vectors to 4 and initialize them with Gaussian noise. The features are normalized to a unit sphere, as proposed in CLIP [35]. And we utilize MSE as the reconstruction loss of the VAE. All the networks are trained using the AdamW optimizer with a learning rate of 0.001. During the fine-tuning of CLIP, since we utilize off-the-shelf methods, we follow the same settings as those proposed in their papers [52, 53, 51, 14]. We randomly synthesize a batch of new class features and combine them with the original batch to form a new batch during training. We conduct all experiments on a single NVIDIA GeForce RTX 3090, except for the ImageNet dataset, which is conducted on an NVIDIA A100.

4.2. Results

4.2.1 Base-to-new generalization

Setup. Following CoCoOp [52], we partition each dataset into two equal non-overlapping subsets: the base classes and the new classes. Subsequently, we randomly extract a few-shot training set from base classes, while preserving the original test set for evaluation purposes. Specifically, we perform training on the base classes with a mere 16 samples per class and evaluate the trained model on both the base and new classes. To evaluate the model's performance, we compute the average accuracy of both the base and new classes, as well as their harmonic mean [52] ($H = 2 \times base \times new/(base + new)$).

Results. We choose CLIP [35], CoOp [53], CoCoOp [52], CLIP-Adapter [14], Tip-Adapter-F [51], VPT [10], and ProDA [29] as our baseline. The result of Tip-Adapter-F [51] is not included in the table due to its inability to test on new classes. Results from Table 1 show that the previous

Table 2. **Cross dataset transfer learning.** The methods are trained on a source dataset (ImageNet) and subsequently evaluated on target datasets. We report the average accuracy of the target datasets. To quantify the performance gains of our method, we compute the difference between the results obtained using our approach (CoOp + SHIP) and the baseline approach (CoOp).

Method	*Caltech101 [12]*	*OxfordPets [34]*	*StanfordCars [24]*	*Flowers102 [33]*	*Food101 [3]*	*FGVC [30]*	*SUN397 [47]*	*DTD [8]*	*EuroSAT [17]*	*UCF101 [37]*	*Average*
CLIP [35]	92.94	89.21	65.32	**71.34**	86.06	**24.72**	62.50	44.39	47.60	66.75	65.08
CoOp [53]	93.70	89.14	64.51	68.71	85.30	18.47	64.15	41.92	46.39	66.55	63.88
CoOp + SHIP	**94.04**	**90.38**	**65.55**	69.67	**86.40**	21.90	**66.62**	**45.69**	**48.17**	**68.52**	**65.69**
Δ	+0.34	+1.24	+1.04	+0.96	+1.10	+3.43	+2.47	+3.77	+1.78	+1.97	+1.81

fine-tuning methods significantly degrade the performance of CLIP on new classes. Specifically, CoOp [53] reduces the accuracy of new classes by an average of 11% across 11 datasets. Tip-Adapter-F [51] is even worse as it fails to recognize new categories outside the training set. It is noteworthy that all previous methods, except VPT [10], harm the CLIP performance on new classes. However, VPT [10] achieves this by reducing the base class accuracy by 10.7%.

As shown in Table 1, we add our generative prompt tuning method to three baseline methods: CoOp [53], CLIP-Adapter [14], and Tip-Adapter-F [51]. By adding our method, CoOp + SHIP outperforms CoOp [53] by 10.47% and 5.07% on the new classes and harmonic mean, respectively, while only sacrificing 2.66% on the base classes. The incorporation of generative prompt tuning into CLIP-Adapter [14] results in a 2.57% and 1.62% improvement in performance on the new classes and harmonic mean, respectively, without affecting the performance of the base classes. Notably, augmenting Tip-Adapter-F [51] with our proposed generative prompt tuning method not only expands its recognition ability to new classes but also achieves almost the best results compared to all the baseline methods. Specifically, Tip-Adapter-F + SHIP achieves a 14.46% improvement on the base classes, 2.20% on the new classes, and 8.24% on the harmonic mean on average across all datasets compared to zero-shot CLIP. Moreover, it obtains the highest harmonic mean on nine of the eleven datasets, except for Caltech101 [12] and OxfordPets [34], where the performance has already reached a high level ($> 95\%$), thus limiting the potential for improvement.

4.2.2 Cross-dataset transfer learning

Setup. Following CoCoOp [52], we present an evaluation of our method's cross-dataset transfer performance. Specifically, we examine the effectiveness of our approach on ten different target datasets following training on the source dataset (ImageNet [9]). To simulate more realistic scenarios, we train our generative model and CoOp [53] on 16-shot ImageNet, utilizing all 1,000 available classes. Subsequently, using the generative model, we generate features for all classes in the target dataset and fine-tune CoOp [53] with the synthesized data. We report the average accuracy of these datasets for a fair comparison.

Results. We report the performance of the proposed CoOp + SHIP compared to the CoOp [53] and CLIP [35] in ten target datasets. The results are shown in Table 2, indicating an improvement range of 0.34% to 3.77%, with an average improvement of 1.81%. Notably, the CoOp + SHIP outperformed the baselines in eight out of ten datasets, with exceptions in Flowers102 [33] and FGVCAircraft [30] datasets. The reason for this observation is that Flowers102 [33] and FGVCAircraft [30] are fine-grained datasets that pose a challenge for the generator to synthesize in-distribution and non-trivial features.

4.2.3 Generalized zero-shot learning

Setup. We follow the same data split and evaluation metrics as in [44]. To ensure fairness in comparison, the model is trained on the complete training set of seen classes instead of 16 shots per class. In this case, we extract the image feature from CLIP visual encoder and obtain the corresponding class attribute from the prompt template "a photo of a {class}". As in [44], we report the average per-class top-1 accuracy on seen and unseen classes. Furthermore, the harmonic mean is also reported to provide a balance between seen and unseen accuracy.

Results. The results of generalized zero-shot learning are shown in Table 3. Experiments are conducted on three standard benchmarks for zero-shot classification: CUB [43], AWA2 [44], and FLO [33]. We choose f-CLSWGAN [45], Cycle-WGAN [13], LisGAN [25], TCN [20], f-VAEGAN [46], TF-VAEGAN [32], GCM-CF [50], HSVA [7], DUET [19], and MSDN [6] as our baseline methods. These methods extract the average-pooled feature instances of size 2,048 from the ImageNet-1K [9] pretrained ResNet-101 [16]. And they use expert annotated

Table 3. **Generalized zero-shot learning.** Models are trained on seen class data and evaluated on the mixture of seen and unseen test datasets. We evaluate on three datasets: CUB [43], AWA2 [44], and FLO [33]. Results are reported in terms of average *top-1* accuracy of unseen and seen classes, together with their harmonic mean (H).

	Method	CUB [43]			AWA2 [44]			FLO [33]		
		Unseen	**Seen**	**H**	**Unseen**	**Seen**	**H**	**Unseen**	**Seen**	**H**
Resnet-101	f-CLSWGAN [45]	43.7	57.7	49.7	57.9	61.4	59.6	59.0	73.8	65.6
	Cycle-WGAN [13]	47.9	59.3	53.0	59.6	63.4	59.8	61.6	69.2	65.2
	LisGAN [25]	46.5	57.9	51.6	52.6	76.3	62.3	57.7	83.8	68.3
	TCN [20]	52.6	52.0	52.3	61.2	65.8	63.4	-	-	-
	f-VAEGAN [46]	48.4	60.1	53.6	57.6	70.6	63.5	56.8	74.9	64.6
	TF-VAEGAN [32]	52.8	64.7	58.1	59.8	75.1	66.6	62.5	84.1	71.7
	GCM-CF [50]	61.0	59.7	60.3	60.4	75.1	67.0	-	-	-
	HSVA [7]	52.7	58.3	55.3	56.7	79.8	66.3	-	-	-
	DUET [19]	39.7	80.1	53.1	48.2	90.2	63.4	-	-	-
	MSDN [6]	**68.7**	67.5	**68.1**	62.0	74.5	67.7	-	-	-
CLIP	CLIP [35]	55.2	54.8	55.0	**88.3**	93.1	**90.6**	65.6	67.9	66.7
	CoOp [53]	49.2	63.8	55.6	72.7	95.3	82.5	52.2	85.8	64.9
	TF-VAEGAN [32]	21.1	**84.4**	34.0	43.7	**96.3**	60.1	37.4	97.2	54.0
	f-VAEGAN [46]	22.5	82.2	35.3	61.2	95.9	74.7	11.1	**97.6**	20.0
	CoOp + SHIP	55.3	58.9	57.1	84.1	94.4	89.0	**69.0**	76.3	**72.4**

attributes or text descriptions [31] as auxiliary information of classes, which requires additional human labor.

The results reported in Table 3 indicate that CoOp [53] yields a substantial improvement in the performance of seen classes. Specifically, the method leads to a 9.0%, 2.2%, and 17.9% performance increase on CUB, AWA2, and FLO datasets, respectively. However, the performance of CoOp on unseen classes is comparatively lower, as evidenced by a decline of 6.0%, 15.6%, and 13.4% on CUB, AWA2, and FLO datasets, respectively, compared to CLIP [35]. This observation suggests that CoOp may suffer from severe overfitting on the seen classes. In this regard, our proposed method, CoOp + SHIP, leverages generative prompt tuning to enhance the performance of unseen classes. Our experimental results demonstrate that CoOp + SHIP leads to significant gains of +6.1%, +11.4%, and +16.8% on unseen classes compared to CoOp [53]. Furthermore, the performance of CoOp + SHIP is comparable or superior to previous zero-shot learning methods.

To ensure a fair comparison, we have implemented the TF-VAEGAN [32] and f-VAEGAN [46] using CLIP extracted features, with the attribute of each class generated through the prompt template "a photo of a {class}". The results presented in the table indicate that while these models achieve the highest performance on seen classes, their performance on unseen classes is significantly lower, suggesting that these models suffer from severe overfitting to the seen classes. We presume that the use of a coarse prompt template such as "a photo of a {class}" may not provide sufficient transferability compared to expert-annotated attributes used in previous methods.

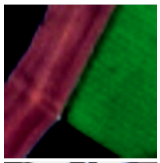
4.3. Ablation Study

Different generative models. We conducted a series of experiments to investigate the effectiveness of the generative framework and the CLIP-based generator. For this, we implemented four distinct types of generators, with two types of frameworks and two types of generators. Table 4 presents the experimental results. In the table, G denotes the use of GAN [2] as the framework, while V denotes the use of VAE [23] as the framework. S represents the training of a three-layer MLP as a generator from scratch, while T denotes the utilization of the CLIP-based generator discussed in Section 3. Notably, V + T is equivalent to our model. We incorporate the generative models into CoOp [53] and evaluate their performance on the 11 datasets mentioned above. In Table 4, the results indicate that VAE-based models outperform GAN-based models, supporting our claims that GANs [15] are difficult to

Table 4. We conducted an ablation analysis to evaluate the effectiveness of the generative frameworks and generators. We add them to CoOp and the results are average on the 11 datasets. The last row is CoOp's results. G: use a GAN-based framework. V: use a VAE-based framework. S: train the generator from scratch. T: using the CLIP-based generator.

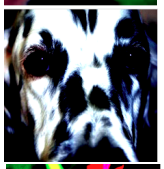
framework	**generator**	base	new	H
G	S	79.96	59.35	67.30
V	S	79.05	69.32	73.41
G	T	77.69	67.32	71.68
V	T	80.03	**73.69**	**76.73**
-	-	**82.69**	63.22	71.66

(a) "aerial - aerial - aerial - aerial." (EuroSAT)



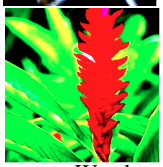
(b) "swift-footed - swift-footed - swift-footed - swift-footed." (OxfordPets)

(c) "fierce - fierce - polydactyl - polydactyl." (OxfordPets)



(d) "outstretched - odd-pinnate - odd-pinnate - three-lobed." (Flowers102)

(e) "outstretched - outstretched - outstretched - short-stem." (Flowers102)



(f) "vibrational - odd-pinnate - odd-pinnate - odd-pinnate." (Flowers102)

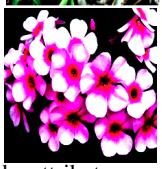


Figure 2. Interpretation of prompts in the latent space. We observe that some words are capable of characterizing the attributes present in the images. However, since we utilize the identical bias in the prompts, some words are the same.

train with the few-shot base dataset, leading to a suboptimal performance on new classes. Additionally, we find that utilizing the CLIP-based generator yields superior results to straightforwardly training the generator from scratch, highlighting the effectiveness of our CLIP-based generator and the efficient utilization of pretrained knowledge of CLIP. Furthermore, the table reveals that the combination of CoOp with G + S yields inferior performance compared to vanilla CoOp [53]. This indicates that not arbitrary data generation for new classes can improve model performance. Based on these results, we select VAE [23] as our generative architecture and choose to utilize the CLIP-based generator.

Table 5. We evaluate different forms of prompts. Results are average on the 11 datasets. **global** denotes whether using global prompts. And **sequential** denotes whether the local bias is sequential or identical.

global	**sequential**	base	new	H
✗	✗	80.70	71.60	75.37
✓	✗	80.03	**73.69**	**76.73**
✗	✓	**80.77**	71.95	75.73
✓	✓	80.59	70.75	74.89

Different forms of generative prompts. The prompts used in our method comprise a fixed set of global prompts and a local instance-specific bias, as described in Section 3. More specifically, the prompts are represented as an addition of the global prompts and the local bias, i.e., $\boldsymbol{p} = [\boldsymbol{p_1} + \boldsymbol{r}, \boldsymbol{p_2} + \boldsymbol{r}, ...\boldsymbol{p_L} + \boldsymbol{r}]$. We investigate the impact of different forms of prompts on performance. We use the term **global** to denote the use of global prompts, $[\boldsymbol{p_1}, \boldsymbol{p_2}, ...\boldsymbol{p_L}]$, and the term **sequential** to refer to the use of identical or sequential local bias, i.e., $[\boldsymbol{r}, \boldsymbol{r}, ..., \boldsymbol{r}]$ or $[\boldsymbol{r}_1, \boldsymbol{r}_2, ..., \boldsymbol{r}_L]$. The results, presented in Table 5, indicate that utilizing global prompts along with identical local bias yields the best performance. And using local prompts alone (whether sequential or not) results in a negative impact on the new class performance, underscoring the importance of global prompts in capturing vital information. Notably, using both **global** and **sequential** prompts results in the worst performance. This may be attributed to the instability of training since they both learn sequential prompts.

Different lengths of prompts. As described in Section 3, our proposed approach generates instance-specific prompts to produce corresponding features, which consist of a global prompt and a local bias. Specifically, the prompts are computed as follows: $\boldsymbol{p} = [\boldsymbol{p}_1 + \boldsymbol{r}, \boldsymbol{p}_2 + \boldsymbol{r}, ..., \boldsymbol{p}_L + \boldsymbol{r}]$, where L is the length of prompts. To examine the influence of prompt length on our method's performance, we conduct an ablation study, the results of which are presented in Table 6. Specifically, we set the prompt lengths in our approach to 1, 2, 4, and 8 and integrate our method into CoOp [53] to evaluate its performance on base-to-new generalization. Our experimental results indicate that our proposed approach performs best when the prompt length is set to $L = 4$. Therefore, we set the default prompt length as $L = 4$.

Table 6. We evaluate different lengths of prompts. Results are average on the 11 datasets.

	Base	New	H
CoOp + SHIP (L=1)	80.61	71.69	75.53
CoOp + SHIP (L=2)	**80.61**	70.28	74.56
CoOp + SHIP (L=4)	80.03	**73.69**	**76.73**
CoOp + SHIP (L=8)	80.54	72.05	75.70

Interpretation of prompts. One benefit of our CLIP-based generative model is that we can provide interpretive prompts. The model learns the mapping from visual features to token embedding space via the VAE process. By utilizing this mapping, we can obtain instance-specific

prompts for the input image. The next step involves selecting the nearest natural words from the vocabulary based on their Euclidean distance to the prompts in latent space. However, the approach maps continuous vectors into discrete codes of words, which can result in generated sentences that may not necessarily be semantically coherent, as noted in prior research [53].

The interpretation of the prompts reveals several noteworthy observations. As depicted in Figure 2 (a), the model has learned to associate the term 'aerial" with images captured from an aerial perspective in EuroSAT. Furthermore, the model has accurately identified some characteristics of dogs, as exemplified in Figure 2 (b)-(c), where the terms "swift-footed" and "fierce" can be used to describe animals. Additionally, the model has demonstrated an understanding of floral morphology, as demonstrated in Figure 2 (d)-(f), where the terms "odd-pinnate," "three-lobed," and "shot-stem" are employed to describe characteristics of flowers. Since we utilize the identical local bias in the prompts, some words are the same in the interpretation sentence.

Although the interpretation may not be entirely precise, it provides valuable insights into the images. We hope the results inform future studies on interpretable vision-language inference and yield further insights.

5. Conclusion

In this paper, we provide a generative approach, SHIP, to handle the scenario where some classes have no data. By training a data-efficient generator to bridge the data gap in new classes, we improve CLIP performance on various tasks using off-the-shelf methods, including base-to-new generalization, cross-data transfer learning, and generalized zero-shot classification. Although achieving remarkable results, it requires additional training costs, which we aim to mitigate in future research. Additionally, future work will explore the applicability of SHIP in dense prediction tasks.

Acknowledgment

This work was partially funded by National Natural Science Foundation of China under Grants 62276256 and Beijing Nova Program under Grant Z211100002121108.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *NeurIPS*, 2022. 1

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *ICML*, 2017. 2, 7

[3] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *ECCV*, 2014. 4, 5, 6, 11, 12

[4] Fei-Long Chen, Du-Zhen Zhang, Ming-Lun Han, Xiu-Yi Chen, Jing Shi, Shuang Xu, and Bo Xu. Vlp: A survey on vision-language pre-training. *Machine Intelligence Research*, 20(1):38–56, 2023. 1

[5] Guangyi Chen, Weiran Yao, Xiangchen Song, Xinyue Li, Yongming Rao, and Kun Zhang. Prompt learning with optimal transport for vision-language models. In *ICLR*, 2023. 2, 4

[6] Shiming Chen, Ziming Hong, Guo-Sen Xie, Wenhan Yang, Qinmu Peng, Kai Wang, Jian Zhao, and Xinge You. Msdn: Mutually semantic distillation network for zero-shot learning. In *CVPR*, 2022. 6, 7

[7] Shiming Chen, Guosen Xie, Yang Liu, Qinmu Peng, Baigui Sun, Hao Li, Xinge You, and Ling Shao. Hsva: Hierarchical semantic-visual adaptation for zero-shot learning. In *NeurIPS*, 2021. 6, 7

[8] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *CVPR*, 2014. 4, 5, 6, 11, 12

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 4, 5, 6, 11, 12

[10] Mohammad Mahdi Derakhshani, Enrique Sanchez, Adrian Bulat, Victor Guilherme Turrisi da Costa, Cees GM Snoek, Georgios Tzimiropoulos, and Brais Martinez. Variational prompt tuning improves generalization of vision-language models. *arXiv preprint arXiv:2210.02390*, 2022. 1, 2, 5, 6

[11] 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. In *ICLR*, 2021. 3, 5

[12] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *CVPRW*, 2004. 4, 5, 6, 11, 12

[13] Rafael Felix, Ian Reid, Gustavo Carneiro, et al. Multi-modal cycle-consistent generalized zero-shot learning. In *ECCV*, 2018. 2, 6, 7

[14] 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. *arXiv preprint arXiv:2110.04544*, 2021. 1, 2, 4, 5, 6, 11, 12

[15] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. In *NeurIPS*, 2014. 2, 4, 7

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3, 6

[17] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *J-STARS*, 2019. 4, 5, 6, 11, 12

[18] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, 2021. 1, 2

[19] Zhen Jia, Zhang Zhang, Liang Wang, Caifeng Shan, and Tieniu Tan. Deep unbiased embedding transfer for zero-shot learning. *IEEE Transactions on Image Processing*, 29:1958–1971, 2019. 6, 7

[20] Huajie Jiang, Ruiping Wang, Shiguang Shan, and Xilin Chen. Transferable contrastive network for generalized zero-shot learning. In *ICCV*, 2019. 6, 7

[21] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. In *NeurIPS*, 2020. 4

[22] Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision. In *ICML*, 2021. 2

[23] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014. 2, 4, 7, 8

[24] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *ICCV*, 2013. 4, 5, 6, 11, 12

[25] Jingjing Li, Mengmeng Jing, Ke Lu, Zhengming Ding, Lei Zhu, and Zi Huang. Leveraging the invariant side of generative zero-shot learning. In *CVPR*, 2019. 2, 6, 7

[26] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, 2022. 1

[27] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35, 2023. 1

[28] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *NeurIPS*, 2019. 2

[29] Yuning Lu, Jianzhuang Liu, Yonggang Zhang, Yajing Liu, and Xinmei Tian. Prompt distribution learning. In *CVPR*, 2022. 1, 2, 4, 5

[30] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 4, 5, 6, 11, 12

[31] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *NeurIPS*, 2013. 2, 7

[32] Sanath Narayan, Akshita Gupta, Fahad Shahbaz Khan, Cees GM Snoek, and Ling Shao. Latent embedding feedback and discriminative features for zero-shot classification. In *ECCV*, 2020. 2, 6, 7

[33] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *ICVGIP*, 2008. 4, 5, 6, 7, 11, 12

[34] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *CVPR*, 2012. 4, 5, 6, 11, 12

[35] 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 *ICML*, 2021. 1, 2, 3, 5, 6, 7, 12

[36] Yuming Shen, Jie Qin, Lei Huang, Li Liu, Fan Zhu, and Ling Shao. Invertible zero-shot recognition flows. In *ECCV*, 2020. 3

[37] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 4, 5, 6, 11, 12

[38] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. In *ICLR*, 2019. 2

[39] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. In *EMNLP-IJCNLP*, 2019. 2

[40] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 3, 5

[41] Feng Wang, Manling Li, Xudong Lin, Hairong Lv, Alexander G Schwing, and Heng Ji. Learning to decompose visual features with latent textual prompts. In *ICLR*, 2023. 1

[42] Xiao Wang, Guangyao Chen, Guangwu Qian, Pengcheng Gao, Xiao-Yong Wei, Yaowei Wang, Yonghong Tian, and Wen Gao. Large-scale multi-modal pre-trained models: A comprehensive survey. *Machine Intelligence Research*, pages 1–36, 2023. 1

[43] Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. Caltech-ucsd birds 200. *Technical Report*, 2010. 4, 6, 7, 11

[44] Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *IEEE transactions on pattern analysis and machine intelligence*, 41(9):2251–2265, 2018. 4, 6, 7, 11

[45] Yongqin Xian, Tobias Lorenz, Bernt Schiele, and Zeynep Akata. Feature generating networks for zero-shot learning. In *CVPR*, 2018. 2, 6, 7

[46] Yongqin Xian, Saurabh Sharma, Bernt Schiele, and Zeynep Akata. f-vaegan-d2: A feature generating framework for any-shot learning. In *CVPR*, 2019. 2, 6, 7

[47] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010. 4, 5, 6, 11, 12

[48] Guo-Sen Xie, Li Liu, Xiaobo Jin, Fan Zhu, Zheng Zhang, Jie Qin, Yazhou Yao, and Ling Shao. Attentive region embedding network for zero-shot learning. In *CVPR*, 2019. 2

[49] Wenjia Xu, Yongqin Xian, Jiuniu Wang, Bernt Schiele, and Zeynep Akata. Attribute prototype network for zero-shot learning. In *NeurIPS*, 2020. 2

[50] Zhongqi Yue, Tan Wang, Qianru Sun, Xian-Sheng Hua, and Hanwang Zhang. Counterfactual zero-shot and open-set visual recognition. In *CVPR*, 2021. 6, 7

[51] 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 *ECCV*, 2022. 1, 2, 4, 5, 6

[52] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *CVPR*, 2022. 1, 2, 4, 5, 6

[53] 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, 2022. 1, 2, 4, 5, 6, 7, 8, 9, 11, 12

A. Appendix

A.1. Datasets details

The details of the 11 datasets used in base-to-new generalization and cross-dataset transfer learning are shown in Table 7. In addition, the statistic of datasets used in generalized zero-shot learning is summarized in Table 8.

Table 7. Datasets statistic of 11 datasets for base-to-new generalization and cross-dataset transfer learning.

Dataset	classes	train	val	test
ImageNet [9]	1,000	1.28M	N/A	50,000
Caltech101 [12]	100	4,128	1,649	2,465
OxfordPets [34]	37	2,944	736	3,669
StanfordCars [24]	196	6,509	1,635	8,041
Flowers102 [33]	102	4,093	1,633	2,463
Food101 [3]	101	50,500	20,200	30,300
FGVCAircraft [30]	100	3,334	3,333	3,333
SUN397 [47]	397	15,880	3,970	19,850
DTD [8]	47	2,820	1,128	1,692
EuroSAT [17]	10	13,500	5,400	8,100
UCF101 [37]	101	7,639	1,898	3,783

Table 8. Datasets statistic of datasets in generalized zero-shot learning.

Dataset	CUB [43]	AWA2 [44]	FLO [33]
# of Attributes	312	85	1,024
# of seen classes	150	40	82
# of unseen classes	50	10	20
# of total images	11,788	30,475	8,189

A.2. Generalized zero-shot setting

The current evaluation protocol utilized in base-to-new generalization assumes that base and new classes are completely isolated during testing, which may not reflect a realistic scenario. In contrast, in a more realistic setting, test sets contain a mix of base and new class data, as previously employed in generalized zero-shot learning. We refer to this as the generalized zero-shot setting and re-evaluate base-to-new generalization under this setting. The results of our evaluation are presented in Table 9, which indicates a significant decrease in performance for previous methods such as CoOp [53] and CLIP-Adapter [14] under this more strict setting. Conversely, our proposed method, SHIP, continues to improve performance in new classes.

Table 9. Evaluate base-to-new generalization under the generalized zero-shot setting, where the base and new data are mixed together in the test dataset.

	Average			ImageNet [9]			Caltech101 [12]			OxfordPets [34]		
	Base	**New**	**H**	**Base**	**New**	**H**	**Base**	**New**	**H**	**Base**	**New**	**H**
CLIP [35]	63.37	**67.39**	65.32	70.04	67.52	68.76	93.61	**91.70**	92.65	84.90	**93.51**	89.00
CoOp [53]	79.76	46.91	59.08	72.51	**67.70**	**70.02**	97.68	85.04	90.92	93.67	64.99	76.74
CoOp + SHIP	78.74	58.53	**67.15**	71.96	67.12	69.45	96.45	90.07	**93.15**	**94.58**	89.15	**91.78**
CLIP-Adapter [14]	**82.79**	30.48	44.55	71.94	64.95	68.27	98.06	71.07	82.41	91.65	33.33	48.89
CLIP-Adapter + SHIP	82.53	35.73	49.87	72.02	66.17	68.97	97.61	77.18	86.20	91.81	40.83	56.52
Tip-Adapter-F + SHIP	82.07	49.02	61.38	**75.46**	60.80	67.34	**98.26**	81.55	89.13	93.99	83.17	88.25

	StanfordCars [24]			Flowers102 [33]			Food101 [3]			FGVCAircraft [30]		
	Base	**New**	**H**	**Base**	**New**	**H**	**Base**	**New**	**H**	**Base**	**New**	**H**
CLIP [35]	59.75	**70.96**	64.87	68.00	**73.90**	70.83	85.84	**86.39**	**86.11**	19.33	**29.87**	**23.47**
CoOp [53]	69.24	60.01	64.30	94.40	37.45	53.62	88.73	78.05	83.05	36.07	13.80	19.96
CoOp + SHIP	67.39	67.07	**67.23**	94.97	61.42	**74.59**	87.82	83.83	85.78	33.43	16.80	22.36
CLIP-Adapter [14]	79.26	34.36	47.94	**98.38**	27.66	43.18	88.42	44.51	59.21	**42.50**	8.58	14.28
CLIP-Adapter + SHIP	78.46	39.07	52.16	97.72	34.26	50.73	88.31	51.70	65.22	42.20	10.26	16.50
Tip-Adapter-F + SHIP	**79.81**	51.84	62.86	95.35	35.25	51.47	**89.84**	75.69	82.16	41.54	14.58	21.58

	SUN397 [47]			DTD [8]			EuroSAT [17]			UCF101 [37]		
	Base	**New**	**H**	**Base**	**New**	**H**	**Base**	**New**	**H**	**Base**	**New**	**H**
CLIP [35]	60.34	**64.91**	62.54	42.13	**46.86**	**44.37**	49.81	**45.44**	**47.52**	63.34	**70.20**	66.59
CoOp [53]	78.75	43.44	56.00	72.80	14.86	24.68	90.81	0.64	1.27	82.68	50.08	62.38
CoOp + SHIP	75.49	59.21	**66.37**	75.23	27.78	40.57	90.67	16.74	28.27	78.18	64.63	**70.76**
CLIP-Adapter [14]	**79.06**	20.35	32.37	81.94	2.90	5.60	**93.60**	0.05	0.10	**85.83**	27.47	41.62
CLIP-Adapter + SHIP	78.96	27.63	40.93	**82.29**	7.85	14.33	93.05	0.44	0.87	85.42	37.59	52.20
Tip-Adapter-F + SHIP	76.60	51.80	61.80	79.98	15.82	26.42	89.83	10.41	18.66	82.11	58.30	68.19