Neurocomputing 583 (2024) 127530

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Cross-Modal Concept Learning and Inference for Vision-Language Models

Yi Zhang [a,b], Ce Zhang [b], Yushun Tang [b], Zhihai He [b,c,*]

[a] Harbin Institute of Technology, Harbin 150001, China
[b] Southern University of Science and Technology, Shenzhen 518055, China
[c] Pengcheng Laboratory, Shenzhen 518000, China

ARTICLE INFO

Communicated by X. Gu

Keywords:
Vision-Language Models
Concept learning
Few-shot learning
Domain generalization

ABSTRACT

Large-scale pre-trained Vision-Language Models (VLMs), such as CLIP, establish the correlation between texts and images, achieving remarkable success on various downstream tasks with fine-tuning. In existing fine-tuning methods, the class-specific text description is matched against the whole image. We recognize that this image-scale matching is not effective since images from the same class often contain a set of different semantic objects, and an object further consists of a set of semantic parts or concepts. Individual semantic parts or concepts may appear in image samples from different classes. To address this issue, in this paper, we develop a new method called cross-model concept learning and inference (CCLI). Using the powerful text-image correlation capability of CLIP, our method automatically learns a large set of distinctive visual concepts from images using a set of semantic text concepts. Based on these visual concepts, we construct a discriminative representation of images and learn a concept inference network to perform downstream image classification tasks, such as few-shot learning and domain generalization. Extensive experimental results demonstrate that our CCLI method is able to improve the performance of the current state-of-the-art methods by large margins, for example, by up to 8.0% improvement on few-shot learning and by up to 1.3% for domain generalization.

1. Introduction

Recently, large-scale pre-trained Vision-Language Models (VLMs) have emerged as an important research topic that has achieved remarkable success on various downstream tasks, such as zero-shot and few-shot image recognition [1–3], image retrieval [4,5], visual grounding [6,7], and visual question answering [5,8]. Compared to traditional methods [9,10], those pre-trained VLMs create a robust connection between texts and images by encoding and mapping texts and images into a unified space, resulting in better transfer capabilities [2,11]. In particular, VLMs such as CLIP [1] and ALIGN [12] are trained using contrastive learning on large-scale image-text pairs collected from the web. They learn aligned embeddings for both images and text, promoting similarity between the representations of an image and its corresponding language descriptions. It should be noted that since these pre-trained VLMs are of massive size and computationally impractical to re-train. Thus, it remains a challenging task to adapt the well-learned knowledge of VLMs to downstream tasks.

To address this issue, a number of approaches have been developed to efficiently adapt such models with very limited supervision. Those approaches can be classified into two categories, namely, prompt tuning methods [3,13,14] and adapter-based methods [2,15,16]. The prompt tuning methods, such as CoOp [3] and CoCoOp [13], focus on designing delicate prompts and introducing learnable context to distill the task-relevant information from the rich knowledge encoded in CLIP. In contrast, adapter-based methods, such as CLIP-Adapter [2] and Tip-Adapter [15], fine-tune the representations generated by CLIP's encoders to better represent images and texts.

We observe that in the current fine-turning methods for CLIP, the class-specific text is matched against the whole image. We recognize that this matching method is not effective because: (1) Images from the same class often contain a set of different semantic objects that correspond to different text descriptions. (2) An object further consists of a set of different semantic parts that also have different text descriptions. (3) On the other hand, individual semantic objects and parts, known as concepts, may appear in image samples from different classes. For example, the `Cat` and `Car` images may both contain the tree object. The `Car` and `Truck` images may both contain the semantic part of wheels or have the same color concepts. This mixture of visual concepts in a natural image will cause major problems when we attempt to match the class-specific text description against the whole image. This problem, if not efficiently addressed, will hinder our capabilities in general image-text understanding and downstream tasks.

To address this important issue, in this paper, we establish and learn a semantic concept-level representation and inference of the image-text

* Corresponding author at: Southern University of Science and Technology, Shenzhen 518055, China.
E-mail address: hezh@sustech.edu.cn (Z. He).

https://doi.org/10.1016/j.neucom.2024.127530
Received 27 July 2023; Received in revised form 27 December 2023; Accepted 4 March 2024
Available online 6 March 2024

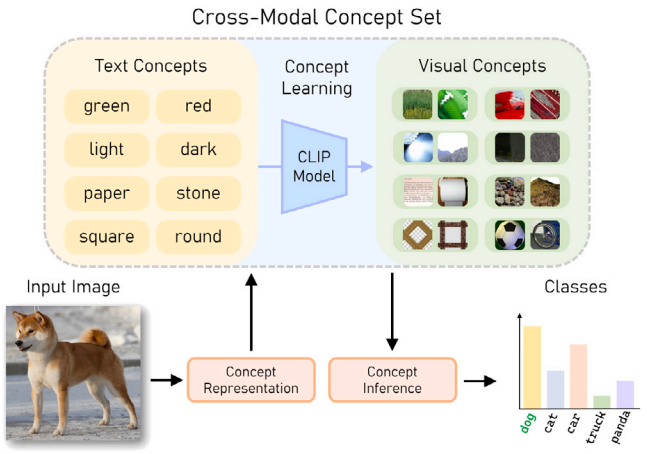


Fig. 1. An illustration of our major idea. Empowered by CLIP, we build a cross-modal concept set to enable concept-level representations for input images and train a concept inference network. This approach leads to improved accuracy in classification results.

pairs. This method, called cross-modal concept learning and inference (CCLI), provides a new approach to exploring the potential of CLIP for correlating text and images. Specifically, as illustrated in Fig. 1, based on the powerful text-image correlation capability of CLIP, our method automatically learns a large set of distinctive visual concepts from images using a set of pre-defined semantic text concepts. Based on these visual concepts, we construct a discriminative representation of images and learn a concept inference network to perform downstream image classification tasks. We observe that this concept-level representation and inference is able to provide better consistency between vision and language modalities, resulting in a much-improved generalization capability. The proposed CCLI method is successfully applied to few-shot image classification and domain generalization tasks. It achieves significantly improved performance, outperforming the current state-of-the-art methods by large margins, for example, by up to 8.0% improvement on few-shot learning and by up to 1.3% for domain generalization.

The main contributions of this work are summarized as follows:

(1) We propose a state-of-the-art method called cross-modal concept learning and inference (CCLI), which provides a new approach to further explore the potential of CLIP for correlating text and images.

(2) We use CLIP to automatically learn a large set of distinctive visual concepts from images based on a set of semantic text concepts. Utilizing these visual concepts, we build a discriminative representation of images and train a concept inference network to perform downstream tasks.

(3) In order to evaluate the effectiveness of our proposed method, we conduct experiments on few-shot classification and domain generalization. Our comprehensive empirical results demonstrate the significantly superior performance of CCLI compared to existing state-of-the-art methods.

The remainder of this paper is structured as follows: The related works are reviewed in Section 2. The review of CLIP and the proposed CCLI method are presented in Section 3. Section 4 provides details on the experimental setup and analyzes the experimental results. Finally, Section 5 presents the conclusion and outlines future work for this paper.

2. Related work

In this section, we review related works on large-scale pre-trained VLMs, fine-tuning for VLMs, few-shot image classification, generalization under distribution shift, and visual concept learning.

(1) Large-scale pre-trained VLMs. Large-scale pre-trained VLMs have been developed to learn general visual representation under the supervision of natural languages [1,17]. Recent research has explored the semantic correspondence between the linguistic and visual modalities by leveraging a large number of image-text pairs available on the internet [1,11,12]. For instance, CLIP [1] is obtained by contrastive learning on 400 million curated image-text pairs, while ALIGN [12] exploits 1.8 billion noisy image-text pairs from the raw alt-text data. Despite the success of VLMs in many downstream applications [5,8], recent studies have also highlighted concerns regarding VLMs' ability to comprehend relation, attribution, and order [18]. **(2) Fine-tuning for VLMs.** Fine-tuning is crucial for VLMs to adapt to various downstream tasks. In this work, we mainly focus on the image classification task. Recent works can be categorized into prompt tuning methods and adapter-based methods.

Prompt tuning methods are motivated by the success of prefix-tuning within the domain of natural language processing [19–21]. As the seminal work in this field, CoOp [3] enhances the prompt context by optimizing it through the utilization of a collection of trainable vectors. Zhou et al. [13] extends the CoOp method to address the generalization problem for unseen classes by learning to generate vectors conditioned on each image. To avoid prompt tuning from forgetting the general knowledge, ProGrad [22] proposes to update the prompts whose gradients are well aligned. [23] introduces a method that correlates the vision and language modalities using optimal transport.

Adapter-based methods, which are inspired by parameter-efficient finetuning methods [24,25], directly tune the representations generated by the CLIP's visual and text encoders. For example, CLIP-Adapter [2] proposes an additional feature adapter to boost conventional fine-tuning results. Tip-Adapter [15] achieves enhanced outcomes by constructing a key–value cache model using low-shot samples. SAL-Adapter [16] combines the inherent strengths of vision-language pre-training and self-supervised representation learning to achieve enhanced performance.

(3) Few-shot image classification. Few-shot learning has been proposed to enable generalization to new tasks with only a few supervised samples [26]. Traditional few-shot learning methods leverage meta learning [27], metric learning [28], and transfer learning [29] to achieve remarkable adaptation capabilities. However, these methods typically require training from base classes in the source domain, which limits their generalization capabilities. Recent advances in pre-trained VLMs have demonstrated a promising alternative approach that does not rely on source-domain training datasets. By keeping the pre-trained weights fixed and training supplementary adaptable modules for downstream tasks, these models can achieve remarkable performance with very limited training data [3,30,31]. For example, Zhang et al. [15] establish a key–value cache model based on the few-shot training set to serve as a weight initialization, Lin et al. [30] propose a cross-modal adaptation approach to learn from few-shot instances spanning different modalities, Najdenkoska et al. [31] defines a meta-mapper network to efficiently bridge frozen large-scale VLMs and leverage their already learned capacity.

(4) Generalization under distribution shift. Distribution shift refers to the discrepancy between the distributions of training data in the source domain and test data in the target domain [32]. The ability to generalize to out-of-distribution (OOD) data is a natural aptitude for humans but remains a challenging task for artificial intelligence models. To address this problem, a number of methods have been developed within the context of domain adaptation [33] and test-time adaptation [34,35]. In this work, we focus on domain generalization [36,37], which aims to address the performance degradation under data distribution shifts, by training models only on the source domains that are generalizable to new unseen target domains [36]. Large-scale pre-trained vision and language models, like CLIP, have showcased remarkable generalization abilities when applied to zero-shot scenarios

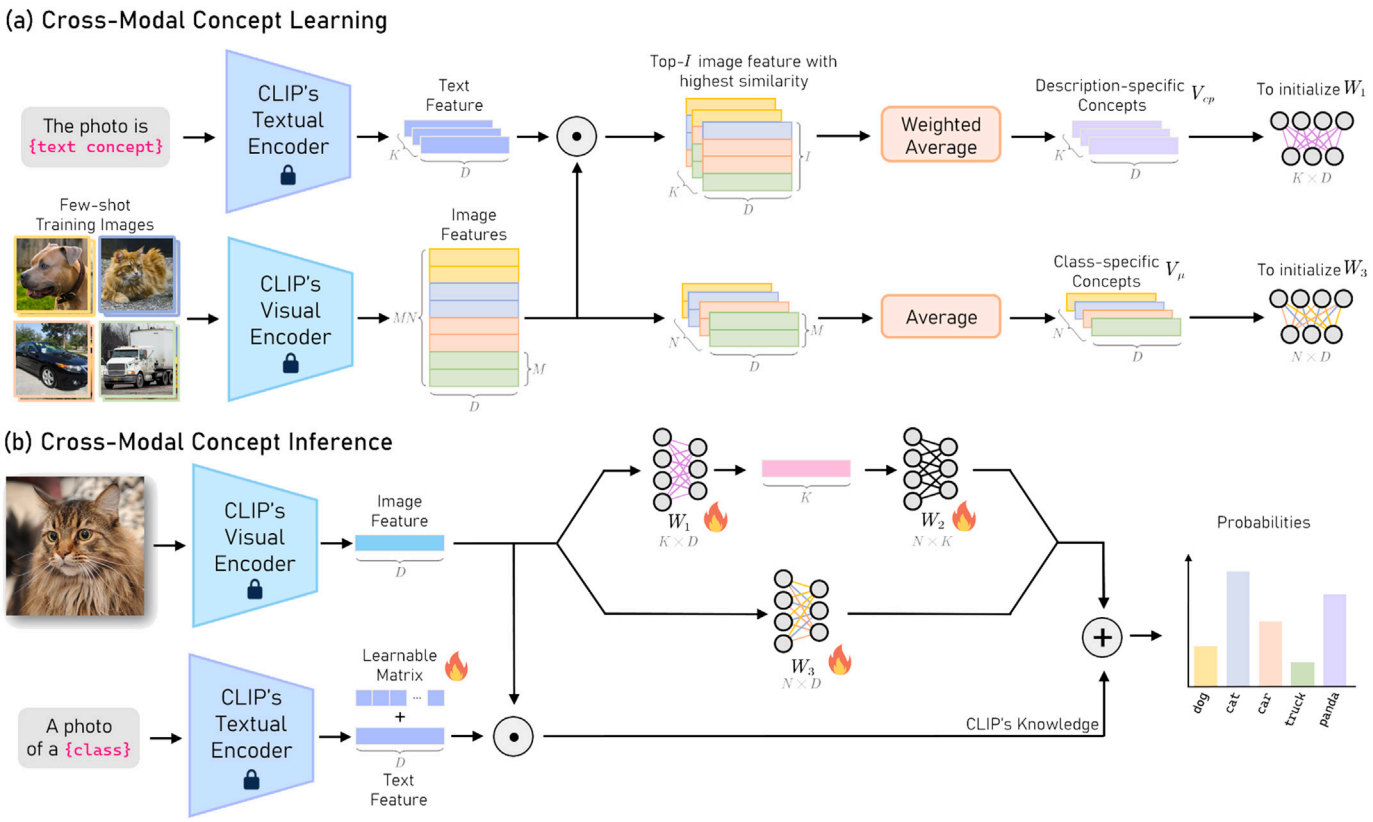


Fig. 2. Overview of our Cross-modal Concept Learning and Inference (CCLI) method. (a) shows the concept learning process of our method, the learned visual concepts include: the description-specific visual concepts which are used to initialize the W_1 of the description-specific inference network, and the class-specific visual concepts which are utilized to initialize the W_3 of class-specific concept inference network. (b) shows the concept inference process, the final logits of our method is obtained by fusing the output logits of the concept inference and the enhanced CLIP. The fire icon means the parameters will be updated during training.

with distribution shifts in various downstream tasks. [1]. This ability to generalize without any fine-tuning on task-specific data is a highly desirable characteristic of VLMs, and presents a promising direction for advancing machine learning methods.

(5) Visual concept learning. Visual concepts/attributes have demonstrated great potential as cues for a variety of visual tasks, *e.g.*, object recognition [38], semantic segmentation [39], and zero-shot transfer [40]. There are two major approaches to visual concept learning that have been explored in the existing literature. The first approach typically requires manual semantic concept labeling (such as colors, textures, and fabric) for the training images [41]. To alleviate the labeling cost, several studies [42,43] propose to learn concepts under triplet supervision, where human annotators should only provide labels for similar and dissimilar objects. The second approach focuses on designing data-driven concepts through unsupervised learning [44]. However, these learned concepts may not have true meaning in most cases. In this work, empowered by CLIP [1], we design an unsupervised concept learning method that is able to learn a large set of visual concepts with true semantic meaning from images using a set of pre-defined text concepts.

3. Method

In this section, we present our method of cross-modal concept learning and inference (CCLI) in detail.

3.1. Revisiting the CLIP model

CLIP [1] consists of two parallel encoders, one for image and the other for text. It normally utilizes a ResNet [45] or a ViT [46] as an image encoder, which maps an image into a visual representation vector. The text encoder is a Transformer, which takes the text as input and generates a textual feature. During the training process, CLIP exploits a contrastive loss to enforce similarity between image-text pairs. We denote CLIP's encoders as $\{E_t, E_v\}$, where E_t is the text encoder and E_v is the image encoder. After training, CLIP can be utilized for image classification in zero-shot scenarios with a hand-crafted prompt [2]. Given a test image $X_{te} \in \mathbb{R}^{C\times H\times W}$ of class y for a N-class classification problem, in the zero-shot setting, we first append the class name text of every y_i in $\{y_i\}_{i=1}^N$ to a hand-crafted prompt denoted by π, such as $\pi =$ "a photo of a", to build a class-specific text inputs $\{\pi; y_i\}$. Then, we generate the text features $\{t_i\}_{i=1}^N$ using the text encoder E_t, where $t_i = E_t(\{\pi; y_i\})$. The cosine similarity score between the text feature t_i and the image feature $v = E_v(X_{te})$ is given by

$$\mathrm{sim}\left(t_i, v\right) = \frac{t_i \cdot v}{\|t_i\|\|v\|}. \tag{1}$$

The prediction probability on X_{te} is computed as

$$p(y=i|X_{te}) = \frac{\exp\left(\mathrm{sim}\left(t_i, v\right)/\tau\right)}{\sum_{j=1}^{K}\exp\left(\mathrm{sim}\left(t_j, v\right)/\tau\right)}, \tag{2}$$

where τ is the temperature hyper-parameter of the softmax function learned by CLIP.

3.2. Overview of our proposed CCLI method

Fig. 2 illustrates the overview of our method. **(a)** outlines how our method learns concepts. We begin by creating a dictionary, Ω_t, containing text descriptions for common visual concepts. Then, we craft specific prompts for each concept. Using the Text Encoder E_t, we generate features for these text concepts. With a few training samples available, we first generate features for the images. For each

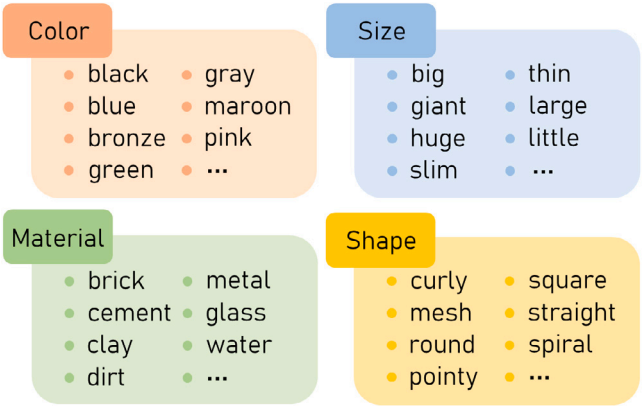


Fig. 3. Examples of text concepts in the concepts dictionary Ω_t. Here, we present some example words describing color, size, material and shape in our dictionary. Each word represents a specific visual concept.

text concept feature, we measure its similarity to the features of the training samples. We then calculate a weighted average of the top I image features that show the highest similarity scores. These averages create description-specific visual concepts for each text description in Ω_t. Simultaneously, for each class of training images, we calculate the average feature to represent the class-specific concept. Importantly, these description-specific visual concepts initialize W_1 of the description-specific inference network, while the class-specific visual concepts initialize the class-specific concept inference network.

(b) illustrates the concept inference process. We improve the original CLIP by adding a text adapter to the text features. When given an input image, we start by using the Visual Encoder E_v to extract the image feature. Initially, we compute enhanced CLIP's logits L_c by calculating the cosine similarity scores between the visual feature and textual feature. Subsequently, we input the image feature into the description-specific concept inference network to obtain the logits, L_a. Then, using the class-specific concept inference network, we derive the logits L_q for class-specific concepts. The final logits of our method result from merging the output logits of the concept inference and the enhanced CLIP.

3.3. Cross-modal concept learning

As shown in Fig. 2(a), we first construct a comprehensive dictionary Ω_t of text concepts of size K which describe major visual concepts in all images. This dictionary contains $K = 1000$ common text descriptions for visual attributes collected from existing visual attribute datasets [41,47], including words describing colors, textures, shapes, actions, materials, expressions, *etc*. The set of text concepts is empirically designed, trying to include different types of descriptions of objects. Some example words in this dictionary are presented in Fig. 3. We denote this dictionary by $\Omega_t \triangleq \{d_i\}_{i=1}^K$. Following the zero-shot setting of CLIP, we first append d_i to a hand-crafted prompt $\pi =$ "The photo is" to build a concept-specific text input $\{\pi; d_i\}$. Then, we can generate text concept features $T \triangleq \{t_i\}_{i=1}^K$ using the text encoder E_t,

$$t_i = E_t(\{\pi; d_i\}). \tag{3}$$

In our proposed method for few-shot learning and domain generalization, the set of visual concepts is learned from the training images using the text concept features T and the CLIP model. For example, for M-shot N-class few-shot learning, we have M annotated images in each of the N classes. The training set is denoted as $X \triangleq \{x_j\}_{j=1}^{MN}$. Using the CLIP visual encoder E_v, we can generate their image features $V \triangleq \{v_j\}_{j=1}^{MN}$, where $v_j = E_v(x_j)$.

For every text concept feature t in T, we calculate the similarity score between t and every visual feature in V by Eq. (1), $S_t = \text{sim}(t, v_j) = tv_j$, in which both t and v_j are normalized. Thus, for each text concept feature t, we have MN similarity scores. Then, we select the top I image features with the highest similarity scores. We compute the weighted average of these top I image features as

$$\bar{v} = \frac{\sum_{i=1}^{I} w_i v_i}{\sum_{i=1}^{I} w_i}, \tag{4}$$

where w_i is the CLIP similarity score between image feature v_i and the text feature t. In this way, for all text concepts, we have obtained their corresponding visual concepts, which can be denoted as $V_{cp} \triangleq \{\bar{v}_i\}_{i=1}^K$. In this work, this set of visual concepts is referred to as the description-specific visual concepts. Fig. 4 shows the top five images for four distinct text concepts selected by the concept learning process to show the effectiveness of our method.

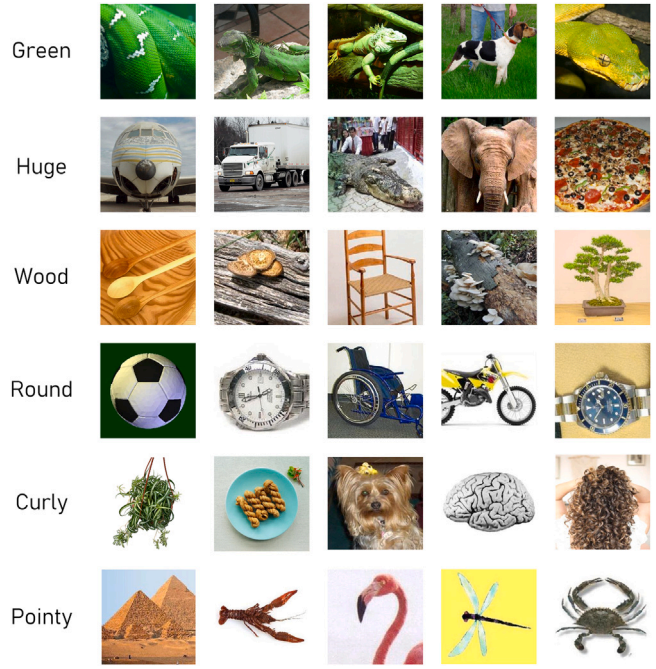


Fig. 4. Examples of top 5 images in the concept learning process. Here, we present images that are most related to the four concepts: green, huge, wood, round, curly, and pointy.

Besides the description-specific visual concepts, we also construct class-specific visual concepts. Specifically, for each class of training images, we calculate the mean feature of M-shot images generated by the visual encoder. We then obtain N class-specific features, $V_\mu \triangleq \{\mu_n\}_{n=1}^N$.

3.4. Cross-modal concept inference

Once the collection of visual concepts is learned, during concept inference, we represent the input image using this set of visual concepts. As shown in Fig. 2(b), based on this visual concept representation, we learn an inference network to classify the image. Our concept inference network consists of two parallel networks appended to the image encoder of CLIP.

The first one is a two-layer network after the image encoder of CLIP. We initialize the first layer with weights $W_1 \in \mathbb{R}^{K \times D}$ with V_{cp}, so that a higher concept score can be obtained when the input feature is consistent with a more compatible description-specific concept feature. Then, the second layer of the network (with weights of $W_2 \in \mathbb{R}^{N \times K}$) integrates all concept scores of the input image and performs effective

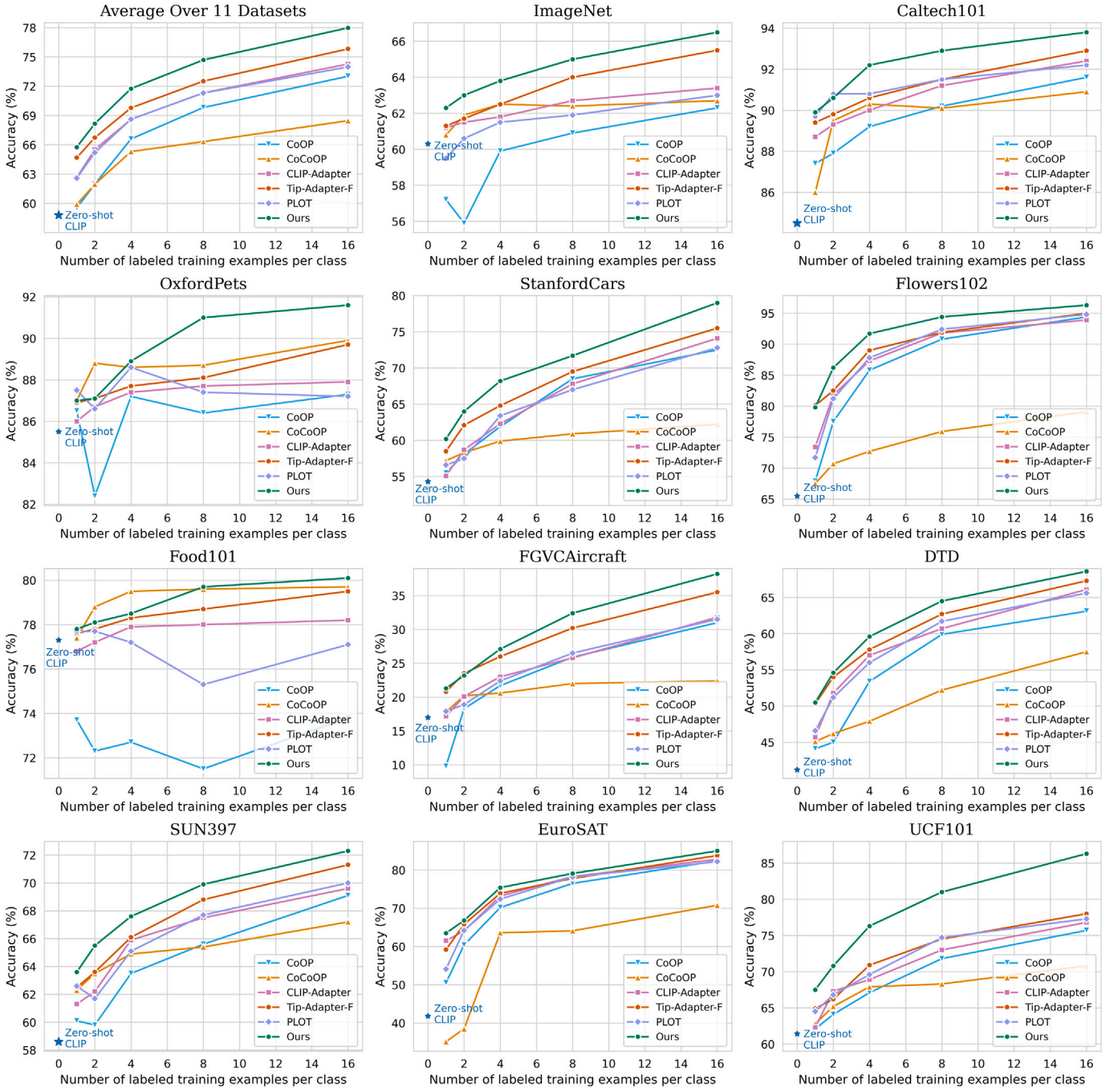


Fig. 5. Classification performance comparison on few-shot learning, *i.e.*, 1-/2-/4-/8-/16-shot, on 11 benchmark datasets. The top-left is the averaged accuracy over the 11 datasets. Overall, our method performs the best in few-shot learning and obtains the highest test accuracy on average over other state-of-the-art methods.

concept inference. This two-layer network, which is a part of the concept inference model, can be denoted as

$$A(x) = W_2\left(\mathrm{ReLU}\left(W_1 x\right)\right). \tag{5}$$

During training, the weights W_1 and W_2 are updated by gradient descent. After supervised learning, the concept features can be optimized for a specific dataset to learn more discriminative concept-level representations. On top of the concept inference model, the affinities [15,48] can be further computed as

$$\mathrm{Aff}(x) = \exp\left(-\delta\left(1 - x\right)\right), \tag{6}$$

where δ is a hyper-parameter for adjusting the sharpness, which controls the influence of the most compatible attribute-specific visual features on the final prediction. The exponential function is utilized to convert the outputs into non-negative values. Given the $L2$ normalized feature $v \in \mathbb{R}^{1\times D}$ of the training image, which is generated by the visual encoder E_v, the logits $L_a \in \mathbb{R}^{1\times N}$ of the concept inference model can be denoted as

$$\begin{aligned} L_a &= \mathrm{Aff}\left(A\left(v\right)\right) \\ &= \exp\left(-\delta(1 - \mathrm{ReLU}(vW_1^{\top})W_2^{\top})\right), \end{aligned} \tag{7}$$

and used for final category classification. Similarly, the second network is simply a one-layer network used to provide class-specific concept inference, denoted as

$$Q(x) = W_3\left(x\right), \tag{8}$$

Table 1
The detailed statistics of datasets used in experiments. The first 11 datasets are used for few-shot learning evaluation, and the last 4 datasets are used for domain generalization.

Dataset	Classes	Training size	Testing size	Task
Caltech101 [50]	100	4128	2465	Object recognition
DTD [51]	47	2820	1692	Texture recognition
EuroSAT [52]	10	13,500	8100	Satellite image recognition
FGVCAircraft [53]	100	3334	3333	Fine-grained aircraft recognition
Flowers102 [54]	102	4093	2463	Fine-grained flowers recognition
Food101 [55]	101	50,500	30,300	Fine-grained food recognition
ImageNet [56]	1000	1.28M	50,000	Object recognition
OxfordPets [57]	37	2944	3669	Fine-grained pets recognition
StanfordCars [58]	196	6509	8041	Fine-grained car recognition
SUN397 [59]	397	15,880	19,850	Scene recognition
UCF101 [60]	101	7639	3783	Action recognition
ImageNet-V2 [56]	1000	–	10,000	Robustness of collocation
ImageNet-Sketch [61]	1000	–	50,889	Robustness of sketch domain
ImageNet-A [62]	200	–	7500	Robustness of adversarial attack
ImageNet-R [63]	200	–	30,000	Robustness of multi-domains

we initialize W_3 as class-specific visual concept V_μ. According to Eq. (6), the logits of class-specific concept inference can be denoted as

$$L_q = \exp\left(-\eta(1 - vW_3^\top)\right), \tag{9}$$

where η is a hyper-parameter similar to δ for adjusting the sharpness.

3.5. CCLI for few-shot learning

Inspired by CoOp and TaskRes [3,49], as illustrated in Fig. 2, we enhance the original CLIP by appending a learnable matrix to the text features $f_t \in \mathbb{R}^{N\times D}$ generated by text encoder E_t. Unlike existing prompt learning methods, our method directly operates on the text features generated by the text encoder, so there is no need to encode the text every time during training. This preserves the original knowledge of CLIP while also allowing for the acquisition of few-shot learning knowledge in an efficient manner. We define the text adapter as $\hat{f}_t = f_t + \beta Z$, where Z is a learnable matrix with the same shape of f_t, β is a hyper-parameter that controls how much of Z we use to combine with f_t. The logits of enhanced CLIP is:

$$L_e = v\hat{f}_t^\top = v(f_t + \beta Z)^\top, \tag{10}$$

where v is the image features generated by E_v. For each task, we learn a task-specific text adapter Z. In this way, we can preserve the prior knowledge of CLIP and obtain the knowledge from new tasks, so that CLIP can be better adapted to downstream tasks.

During few-shot learning, we combine the output logits of the concept inference and the text adapter, and the total logits of the input image v used for the final classification are calculated as

$$\text{Logits} = \alpha L_a + \lambda L_q + L_e, \tag{11}$$

where α is a hyper-parameter that controls the ratio of different logits introduced in Eq. (7), from concept inference with enhanced CLIP. The hyperparameter λ controls the ratio of logits introduced in Eq. (9) from class-specific concept inference. The sensitivity levels of the hyper-parameters are evaluated in Section 4.3. The pseudo-code of the proposed CCLI method is shown as Algorithm 1 in Appendix.

4. Experimental results

In this section, we present performance comparisons with state-of-the-art methods on the few-shot learning and domain generalization tasks, and ablation studies to demonstrate the effectiveness of our proposed method. We summarize the detailed statistics of datasets used in experiments in Table 1.

Table 2
Few-shot classification accuracy (%) on ImageNet [64] of different methods with quantitative values. The results marked as **bold** represent the highest performance, while the second-best results are indicated by being underlined.

Few-shot setup	1	2	4	8	16
Zero-shot CLIP [1]	60.33	60.33	60.33	60.33	60.33
Linear-probe CLIP [1]	22.17	31.90	41.20	49.52	56.13
CoOp [3]	57.21	55.93	59.88	60.91	62.26
CoCoOp [13]	60.78	61.91	62.49	62.38	62.70
CLIP-Adapter [2]	61.20	61.52	61.84	62.68	63.59
Tip-Adapter-F [15]	61.32	61.69	62.52	64.00	65.51
PLOT [23]	59.54	60.64	61.49	61.92	63.01
DeFo [65]	59.44	59.72	60.28	61.73	64.00
CCLI (Ours)	**62.27**	**62.96**	**63.76**	**64.95**	**66.53**

4.1. Few-shot learning

The objective of few-shot learning is to transfer a trained model to novel tasks with limited available supervision. Pre-trained VLMs, such as CLIP, provide a new paradigm for this task.

4.1.1. Experimental Settings

Following prior methods [3,15], we adopt the few-shot evaluation protocol to assess our method on 11 widely-used image classification datasets in Table 1, spanning the breadth of generic object classification (ImageNet [56], Caltech101 [50]), fine-grained object classification (OxfordPets [57], StandfordCars [58], Flowers102 [54], Food-101 [55], FGCV Aircraft [53]), texture classification (DTD [51]), remote sensing recognition (EuroSAT [52]), scene recognition (SUN397 [59]), and action recognition (UCF101 [60]). These datasets provide a comprehensive benchmark to evaluate the few-shot learning performance for each method.

4.1.2. Comparison methods

We compare our method with eight baseline methods reviewed in Section 2: zero-shot CLIP [1], linear probe CLIP [1], CoOp [3], CoCoOp [13], CLIP-Adapter [2], Tip-Adapter-F [15], PLOT [23] and DeFo [65]. Therein, zero-shot CLIP relies on manually designed prompts. For a fair comparison, we choose CoOp's best-performance setting - with the class token placed at the end of 16-token prompts. We also choose the best variant of CLIP-Adapter and the fine-tuned version of Tip-Adapter (Tip-Adapter-F) in our experiments.

4.1.3. Implementation details

Our model is built upon the publicly available CLIP model. We use the ResNet-50 image encoder and transformer text encoder as the CLIP backbone. Throughout the training process, we keep both the visual

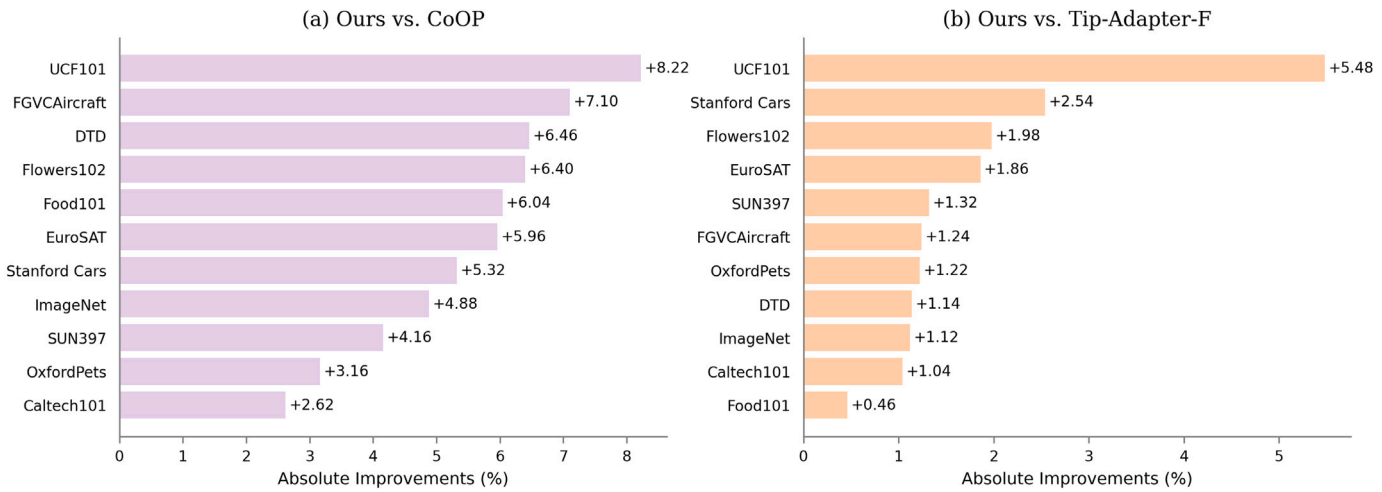


Fig. 6. Comparison with CoOp [3] and Tip-Adapter-F [15]. We show the absolute improvement of our method compared with prompt tuning method (CoOp) and adapter-based method (Tip-Adapter-F). These comparisons are conducted by their average results of few shots (1, 2, 4, 8, and 16) per category.

and text encoders frozen. We follow the data pre-processing protocol in CLIP, including operations of resizing, random cropping, *etc*. In our experiments, the hyper-parameter I to control the number of top visual features is set to 5. We train our model for 100 epochs on ImageNet and 50 epochs on other datasets. The text feature adapter, which is a learnable matrix with the same shape as the text features generated by the text encoder, is initialized with zeros. We set β in Eq. (10) to 0.8 for ImageNet and 0.6 for the rest datasets. We use a batch size of 256 and an initial learning rate of 10^{-3}. Our models are optimized by AdamW [66] optimizer with a cosine annealing scheduler. One single NVIDIA RTX 3090 GPU is used for training. We adhere to the conventional evaluation protocol for few-shot learning, where training involves a random selection of 1, 2, 4, 8, and 16 shots per class, followed by testing on the complete test set.

4.1.4. Performance results

In Table 2, we compare the few-shot classification accuracy of our method on ImageNet [64] with other state-of-the-art methods. The detailed results are shown in Table 9 of Appdendix. Our proposed method obtains promising results in this dataset and an average of +1% improvement can be observed in all few-shot settings.

Fig. 5 shows the comparison with five baseline methods on all 11 datasets, and the average accuracy is shown in the top-left sub-figure of Fig. 5. We observe that our method performs the best in few-shot learning and obtains the highest test accuracy on average over other state-of-the-art methods. Notably, with the increase in the number of shots, the performance gain over other methods becomes larger. This proves that training with more shots enables our model to build a more robust and discriminative concept-level representation. In comparison to zero-shot CLIP, our method consistently surpasses it by huge margins on all datasets. In Fig. 5, our method performs worse on 1 and 2 shots for OxfordPets and Food101 datasets. This is because, when computing the class-specific concepts, we only have 1 or 2 images to represent the whole class. This is not fully effective and degrades the performance. However, the overall performance has shown the effectiveness of our proposed CCLI method.

Comparison with prompt tuning methods. Compared to CoOp [3], which is one of the state-of-the-art prompt learning methods, our approach consistently yields better recognition performance. The absolute performance improvement compared to CoOp on each dataset is shown on the left side of Fig. 6. We can see that the largest gain over CoOp is +8.2% on UCF101, and even the smallest gain is +2.6%. Furthermore, our method outperforms CoCoOp by a huge margin on the average performance as indicated in Fig. 5. This demonstrates that our methods yield superior performance against the prompt learning methods. These prompt tuning methods primarily rely on lightweight, trainable parameters. However, their performance in few-shot scenarios remains limited due to a lack of explicit utilization of pre-trained prior knowledge within these added components. In contrast, our proposed Cross-Modal Concept Learning and Inference (CCLI) method takes a significant step forward. It explicitly leverages pre-trained few-shot knowledge while also integrating visual concepts to augment the final inference process. By strategically combining pre-existing knowledge with the incorporation of visual concepts, CCLI aims to significantly enhance the model's ability to reason and generalize effectively in few-shot scenarios.

Comparison with adapter-style methods. As shown in the top-left sub-figure of Fig. 5, our method exhibits significantly superior performance compared to the CLIP-Adapter [2] and Tip-Adapter [15] on these 11 datasets. Compared to the CLIP-Adapter, our method obtains superior performance on all the datasets. Tip-Adapter-F is the top-performing method with adapter style. Our method outperforms Tip-Adapter-F by an average of +1.8% on all datasets. The largest gain over Tip-Adapter-F is +8.3% on the UCF101 with a 16-shot setting. The absolute performance improvements compared to Tip-Adapter-F on each dataset are shown on the right side in Fig. 6. We can see that our model achieves the largest performance gain of +5.5% over Tip-Adapter-F on UCF101. Overall, our method substantially outperforms its baselines in few-shot learning tasks. These findings serve as strong evidence showcasing the effectiveness of our approach. Upon further analysis, it was observed that the Tip-Adapter [15] solely utilizes the entire image feature to correspond with class-specific text, disregarding the combination of visual concepts. This approach might lead to misclassifications. For instance, in Caltech101, both airplane and flamingo images feature the tree object and share the same red color. The Tip-Adapter incorrectly categorizes some flamingo images as airplanes, whereas our method achieves accurate classification. Our approach employs reasoning based on a wide array of learned distinctive visual concepts, effectively resolving the aforementioned issues.

4.2. Domain generalization

Robustness to distribution shift is critical for the generalization ability of machine learning models. Pre-trained VLMs such as CLIP have exhibited strong robustness to distribution shifts.

Table 3
Comparison with other methods on robustness (%) to natural distribution shifts (from ImageNet to ImageNet-V2/-Sketch/-A/-R). The Experiments are conducted on ResNet-50 and ViT-B/16 respectively. The results marked as **bold** represent the highest performance, while the second-best results are indicated by being underlined.

Method	Visual backbone	Source	Target				
		ImageNet	-V2	-Sketch	-A	-R	OOD average
Zero-shot CLIP [1]	ResNet-50	60.33	53.27	35.44	21.65	56.00	41.59
Linear probe CLIP [1]		56.13	45.61	19.13	12.74	34.86	28.09
CoOp [3]		63.33	55.40	34.67	23.06	56.60	42.43
CoCoOp [13]		62.81	55.72	34.48	23.32	57.74	42.82
ProGrad [22]		62.17	54.70	34.40	23.05	56.77	42.23
PLOT [23]		63.01	55.11	33.00	21.86	55.61	41.40
DeFo [65]		64.00	**58.41**	33.18	21.68	55.84	42.28
TPT [14]		60.74	54.70	35.09	26.67	59.11	43.89
CCLI (Ours)		**66.53**	58.18	**37.17**	**30.93**	**59.79**	**46.52**
Zero-shot CLIP [1]	ViT-B/16	67.83	60.83	46.15	47.77	73.96	57.18
Linear probe CLIP [1]		65.85	56.26	34.77	35.68	58.43	46.29
CoOp [3]		71.51	64.20	47.99	49.71	75.21	59.28
CoCoOp [13]		71.02	64.07	48.75	50.63	76.18	59.91
ProGrad [22]		70.45	63.35	48.17	49.45	75.21	59.05
TPT [14]		68.98	63.45	47.94	54.77	77.06	60.81
CCLI (Ours)		**74.57**	**67.15**	**49.78**	**58.03**	**77.83**	**63.20**

4.2.1. Experimental settings

We evaluate the domain generalization performance of our method by 16-shot training on ImageNet [64] and testing on four ImageNet variant datasets: ImageNet-V2 [56], ImageNet-Sketch [61], ImageNet-A [62], and ImageNet-R [63] in Table 1. ImageNet-V2 [56] serves as a replicated test set comprising 10,000 natural images obtained from an alternative source. This collection encompasses 1000 ImageNet classes. ImageNet-Sketch [61] encompasses a dataset of 50,000 monochrome sketch images, all of which belong to the same set of 1000 ImageNet classes. ImageNet-A [62] is a collection of naturally adversarially filtered images, featuring 7500 pictures from 200 classes selected out of the original 1000 classes found in ImageNet. ImageNet-R [63] is a dataset containing images with artistic renditions of ImageNet categories, including 30,000 images belonging to 200 of ImageNet's 1000 categories.

4.2.2. Comparison methods

We include nine previous methods reviewed in Section 2 for comparisons: zero-shot CLIP [1], linear probe CLIP [1], CoOp [3], CoCoOp [13], ProGrad [22], PLOT [23], DeFo [65], TPT [14], TPT + CoOp [14]. Therein, TPT + CoOp is a method that applies TPT to prompts learned by CoOp and performs better than standalone TPT.

4.2.3. Performance results

Table 3 summarizes the results with two different visual backbones: ResNet-50 and ViT-B/16. We report the classification accuracy of the source domain (ImageNet), target domain (ImageNet-V2, ImageNet-Sketch, ImageNet-A, ImageNet-R), and the target average accuracy (OOD Average). We can see that our method outperforms all other methods in most scenarios, which shows our model's remarkable robustness to distribution shifts. In detail, using the ResNet-50 visual backbone, our proposed CCLI demonstrates superior performance across all settings, except for Imagenet-v2. Specifically, our method outperforms the second-best by up to +4.2% on ImageNet-A and +2.6% on the out-of-distribution (OOD) average. TPT [14], dynamically learning adaptive prompts in real-time with a single test sample, achieves the second-best performance on ImageNet-A, ImageNet-R, and the OOD average. Interestingly, Zero-shot CLIP [1] achieves the second-best performance on ImageNet-Sketch. For the ViT-B/16 backbone, we observe that our method CCLI surpasses all baselines across all datasets, achieving a performance gain of +2.4% on the OOD average. These findings indicate our method's ability to effectively transfer knowledge from the source domain to the target domain by leveraging visual concepts.

Table 4
Effectiveness of different components in our method. We compare the performance of zero-shot CLIP combined with different components. CI is concept inference, and TA represents the text adapter.

Few-shot setup	1	2	4	8	16
Zero-shot CLIP	60.33	60.33	60.33	60.33	60.33
+ CI	62.06	62.59	63.60	64.73	66.38
+ CI + TA	**62.27**	**62.96**	**63.76**	**64.95**	**66.53**

4.3. Ablation studies

To systematically evaluate our proposed method, we provide an empirical analysis of our design choices and illustrate the effects of different components of our method in this section. Ablations on the visual backbones and the number of shots are also reported in this section. All the experiments are conducted on ImageNet.

Contributions of major algorithm components. Our method has two major new components, namely concept inference (CI) and text adapter (TA) in Section 3.5, as shown in Table 4, we find that both components contribute significantly to the overall performance. First, the concept inference network effectively leverages well-learned description-specific and class-specific concepts to carry out downstream image classification tasks. This leads to an improvement of up to 6% compared to zero-shot CLIP in a 16-shot setting. Our results demonstrate the efficacy of this dual-branch structure design, enhancing CLIP's adaptation capabilities. Second, the text adapter yields an additional improvement of up to 0.4% by acquiring task-specific knowledge and enhancing the original textual representation of CLIP. Notably, this text adapter functions by fine-tuning the text features without damaging the CLIP's prior knowledge.

Description-specific and class-specific visual concepts. Table 5 shows the few-shot accuracy on ImageNet [64] obtained by removing description-specific and class-specific visual concepts from our proposed model. The results demonstrate the substantial contributions of both types of concepts to the overall performance. Eliminating either of these components results in a significant decrease in accuracy, underscoring the importance of both description-specific and class-specific visual concepts in our method.

Visual backbones. Table 6 summarizes the results on 16-shot ImageNet using various visual backbones containing ResNets and ViTs. The anticipated results aligns with the expectation: the performance improves with more advanced backbone models. In addition, no matter which visual backbone is used, our method consistently outperforms CLIP-Adapter [2] and Tip-Adapter [15].

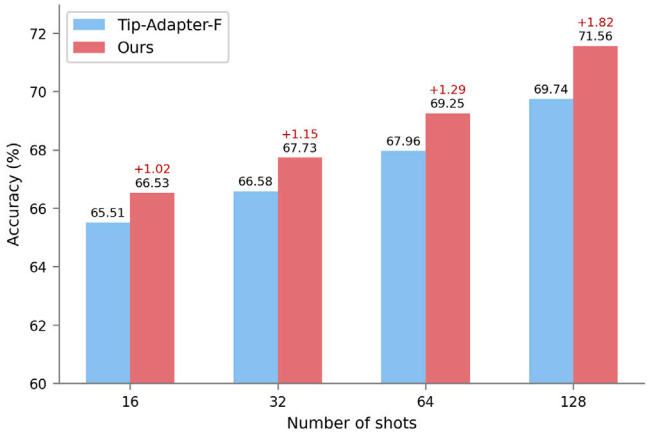


Fig. 7. More shots for training. We compare the performance of training with more shots (32, 64, 128). The experiments are conducted with ResNet-50 visual backbone on ImageNet.

Table 5
Effectiveness of description-specific and class-specific concepts in our method. We report the accuracy without each kind of concept on ImageNet [64] dataset. V_μ and V_{cp} represent class-specific and description-specific concepts, respectively.

Few-shot setup	1	2	4	8	16
Ours	62.27	62.96	63.76	64.95	66.53
w/o V_μ	61.73	62.29	62.37	62.93	64.02
w/o V_{cp}	61.22	61.48	62.11	62.28	63.35

Table 6
Evaluation of various visual backbones on ImageNet. We report the results using a 16-shot setting for training.

Backbone	ResNet-50	ResNet-101	ViT-B/32	ViT-B/16
Zero-shot CLIP	60.33	62.53	63.80	67.83
CLIP-Adapter	63.59	65.39	66.19	71.13
Tip-Adapter-F	65.51	68.56	68.65	73.69
CCLI (Ours)	**66.53**	**69.36**	**69.60**	**74.57**

More shots for training. The result is shown in Fig. 7. Our method achieves remarkable performance with more than 16 shots for training. As the number of shots grows, our method obtains more improvement in recognition accuracy. Compared to Tip-Adapter-F [15], our method achieves significant performance gains ranging from +1.02% (16-shot) to +1.82% (128-shot). This implies that with an increase in the number of shots, our cross-modal concept learning and inference method achieves greater robustness and accuracy.

Sensitivity of hyper-parameters. In our experiments on ImageNet [64], we set the hyper-parameters α, δ, and β to 1.5, 4.5, and 0.8, respectively. To analyze the sensitivity of our model to these hyper-parameters, we conducted experiments by varying each of them and evaluated their impact on the model's performance. Table 7 shows that the value of α, which controls the ratio of different components in the final logit, has a significant impact on the model's performance. When α is set to 0, the method degrades to zero-shot CLIP with only a text adapter. A moderate value of 1.5 leads to the best performance for our model. The hyper-parameter δ, which controls the sharpness, has a relatively limited impact on performance. Our experiments show that the best performance is achieved when we set $\delta = 4.5$. The sensitivity analysis of hyper-parameter β indicates that its influence on the model's performance is minor. Varying β had only a negligible effect on the model's performance. Finally, we also conduct an ablation study of I on the 16-shot ImageNet with $I = 1, 3, 5, 7, 9$. The accuracy varies from 63.78% to 66.53% and $I = 5$ yields the optimal performance.

We also conduct experiments on the value of K, which stands for the number of text descriptions in the concept dictionary Ω_I. The results shown in Table 7 show that our method achieves the best performance when $K = 1000$. If the number is too small, there will not be sufficient text descriptions available to cover the range of visual concepts, resulting in inferior performance. Conversely, if the number is too large, performance might decline due to multiple text descriptions matching the same visual concept.

Table 7
Sensitivity of hyper-parameters. All the results are reported on a 16-shot setting on ImageNet [64]. α controls the ratio of different components in the final logits. δ is used to adjust the sharpness. β is used to control the scaling of the residual, I is the count of image features used to generate visual concepts. K is the number of text descriptions of concepts.

Sensitivity of hyper-parameters						
α	0.0 61.60	0.5 64.21	1.0 65.67	**1.5** **66.53**	2.0 64.92	2.5 64.71
δ	0.5 65.97	2.5 66.26	**4.5** **66.53**	6.5 66.23	8.5 66.11	10.5 66.05
β	0.1 66.38	0.2 66.42	0.4 66.48	0.6 66.50	**0.8** **66.53**	1.0 66.40
I	1 63.78	3 65.42	**5** **66.53**	7 66.37	9 66.18	11 65.94
K	100 63.21	500 65.08	**1000** **66.53**	1500 66.21	2000 65.95	2500 65.86

Table 8
Efficiency and accuracy for different methods on ImageNet-16-shot. The experimental evaluations are conducted using a batch size of 32 on a single NVIDIA GeForce RTX 3090 GPU. The final column displays the performance improvement of each method over zero-shot CLIP.

Method	Epochs	Time	Accuracy	Gain
Zero-shot CLIP [1]	0	0	60.33	0
Linear probe CLIP [1]	–	13 min	56.13	−4.20
CoOp [3]	200	14 h 40 min	62.26	+1.93
ProGrad [22]	200	17 h	63.45	+3.12
CLIP-Adapter [2]	200	50 min	63.59	+3.26
Tip-Adapter-F [15]	20	5 min	65.51	+5.18
Ours	20	**4 min**	**66.53**	**+6.20**

4.4. Complexity analysis

Table 8 compares the performance and training time of our proposed method with state-of-the-art methods for 16-shot image classification on ImageNet [64]. Based on the information provided in the table, it is evident that our approach significantly improves the accuracy while requiring relatively short training time.

5. Conclusion

The major contributions of this work can be succinctly outlined as follows. (1) We explore the powerful capabilities of CLIP in correlating texts and images and develop a new method to automatically learn visual concepts from training images based on a collection of semantic text concepts. (2) Based on these visual concepts, we are able to construct a discriminative representation of images and learn a concept inference network to perform downstream tasks. (3) Extensive experimental results on few-shot image classification and domain generalization have demonstrated our proposed CCLI method outperforms the current state-of-the-art methods by large margins.

The proposed idea can be naturally incorporated into other CLIP-based visual learning tasks, such as visual question answering, image captioning, and visual grounding. In the future, we hope to apply our approach to these tasks.

Table 9
Per-dataset results on the ResNet-50 backbone. We also include results from existing works for easier comparison. We **bold** the best result for each shot and each dataset, and underline the second best result.

Method	Shots	Dataset											
		Caltech [50]	ImageNet [64]	DTD [51]	EuroSAT [52]	Aircraft [53]	Food [55]	Flowers [54]	Pets [57]	Cars [58]	SUN397 [59]	UCF101 [60]	Average
Zero-shot CLIP [1]	0	84.5	60.3	41.2	41.8	17.0	77.3	65.5	85.5	54.3	58.6	61.4	58.8
CoOp [3]	1	87.4	57.2	44.1	50.5	9.8	73.7	67.9	86.5	55.5	60.1	62.1	59.5
	2	87.9	55.9	45.0	60.4	18.3	72.3	77.5	82.4	58.1	59.8	64.1	62.0
	4	89.2	59.9	53.4	70.2	21.7	72.7	85.8	87.2	61.9	63.5	67.1	66.6
	8	90.2	60.9	59.9	76.5	25.9	71.5	90.8	86.4	68.5	65.6	71.8	69.8
	16	91.6	62.3	63.1	82.4	31.0	73.8	94.4	87.3	72.5	69.1	75.7	73.0
CoCoOP [13]	1	86.0	60.8	45.1	35.1	17.8	77.4	67.5	87.0	57.2	62.3	62.8	59.9
	2	89.5	61.9	46.2	38.5	20.2	**78.8**	70.7	**88.8**	58.3	63.5	65.2	62.0
	4	90.3	62.5	47.9	63.6	20.6	**79.5**	72.7	88.6	59.9	64.9	67.9	65.3
	8	90.1	62.4	52.2	64.1	22.0	79.6	75.9	88.7	60.9	65.4	68.3	66.3
	16	90.9	62.7	57.5	70.8	22.4	79.7	79.1	89.9	62.2	67.2	70.8	68.5
CLIP-Adapter [2]	1	88.7	61.2	45.7	61.5	17.2	76.8	73.4	86.0	55.1	61.3	62.3	62.7
	2	89.3	61.5	51.8	64.1	20.1	77.2	81.8	86.7	58.7	62.2	67.3	65.5
	4	90.0	61.8	57.0	73.2	23.0	77.9	87.3	87.4	62.3	65.9	68.9	68.6
	8	91.2	62.7	60.7	78.3	25.8	78.0	91.8	87.7	67.8	67.5	73.0	71.3
	16	92.4	63.6	66.1	82.8	31.8	78.2	93.9	87.9	74.1	69.6	76.8	74.3
Tip-Adapter-F [15]	1	89.4	61.3	50.3	59.2	20.8	77.6	**80.1**	86.9	58.5	62.5	64.9	64.7
	2	89.8	61.7	54.0	65.8	**23.5**	77.8	82.5	87.1	62.1	63.6	66.2	66.7
	4	90.6	62.5	57.8	73.9	26.0	78.3	89.0	87.7	64.8	66.1	70.9	69.8
	8	91.5	64.0	62.7	77.8	30.2	78.7	91.9	88.1	69.5	68.8	74.5	72.5
	16	92.9	65.5	67.3	83.8	35.5	79.5	95.0	89.7	75.5	71.3	78.0	75.8
PLOT [23]	1	89.7	59.5	46.6	54.1	17.9	77.7	71.7	**87.5**	56.6	62.6	64.5	62.6
	2	**90.8**	60.6	51.2	64.2	18.9	77.7	81.2	86.6	57.5	61.7	66.8	65.2
	4	90.8	61.5	56.0	72.4	22.4	77.2	87.8	88.6	63.4	65.1	69.6	68.6
	8	91.5	61.9	61.7	78.2	26.5	75.3	92.4	87.4	67.0	67.7	74.7	71.3
	16	92.2	63.0	65.6	82.2	31.5	77.1	94.8	87.2	72.8	70.0	77.3	74.0
Ours	1	**89.9**	**62.3**	**50.5**	**63.5**	**21.3**	**77.8**	79.8	87.0	**60.2**	**63.6**	**67.5**	**65.8**
	2	90.6	**63.0**	**54.6**	**66.8**	23.2	78.1	**86.2**	87.1	**64.0**	**65.5**	**70.8**	**68.2**
	4	**92.2**	**63.8**	**59.6**	**75.4**	**27.1**	78.5	**91.7**	**88.9**	**68.2**	**67.6**	**76.3**	**71.8**
	8	**92.9**	**65.0**	**64.5**	**79.1**	**32.4**	**79.7**	**94.4**	**91.0**	**71.7**	**69.9**	**81.0**	**74.7**
	16	**93.8**	**66.5**	**68.6**	**85.0**	**38.2**	**80.1**	**96.3**	**91.6**	**79.0**	**72.3**	**86.3**	**78.0**

CRediT authorship contribution statement

Yi Zhang: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Project administration, Resources, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Ce Zhang:** Resources, Software, Visualization, Writing – original draft, Writing – review & editing. **Yushun Tang:** Data curation, Resources, Validation, Visualization, Writing – review & editing. **Zhihai He:** Project administration, Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Appendix

In this section, we include the pseudo-code and the detailed results of few-shot classification in Fig. 5. The pseudo-code of the proposed CCLI method is shown as Algorithm 1. The detailed results of Fig. 5 are illustrated in Table 9.

Algorithm 1: Pseudocode of our CCLI method.

Input: Pre-trained CLIP image and text encoder E_v, E_t;
Input: The dictionary of text concepts Ω_t;
Input: Training set $\mathcal{D}^{tr}$ of target task;
Input: Class names $\{c_i\}_{i=1}^{N}$ and hand-crafted prompt π;
Generate text concept features T for all text concepts in dictionary Ω_t by Equation (3);
Learn description-specific concepts V_{cp};
Learn class-specific concepts V_{μ};
Initialize W_1 with description-specific concepts V_{cp};
Initialize W_3 with class-specific concepts V_{μ};
Initialize the task-specific text adapter Z with zeros;
for _ ***in*** *iterations* **do**
  Sample a batch $\{(x_j, y_j)\}_{j=0}^{J}$ from $\mathcal{D}^{tr}$;
  Compute $f_t = \{E_t(\{\pi, c_i\})\}, i = 1, \ldots, N$;
  Compute $\hat{f}_t$ using the learnable matrix Z;
  Compute $v = \{E_v(x_j)\}, j = 1, \ldots, J$;
  Let $Labels = \{y_j\}_{j=1}^{J}$;
  Compute L_a, L_q and L_e according to Equation (7), (9) and (10) ;
  Compute $Logits$ according to Equation (11);
  Compute $loss = CrossEntropyLoss(Logits, Labels)$;
  Update W_1, W_2, W_3, Z by gradient descent;
end

References

[1] A. Radford, J.W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al., Learning transferable visual models from natural language supervision, in: International Conference on Machine Learning, 2021.

[2] P. Gao, S. Geng, R. Zhang, T. Ma, R. Fang, Y. Zhang, H. Li, Y. Qiao, Clip-adapter: Better vision-language models with feature adapters, 2021, arXiv preprint arXiv:2110.04544.

[3] K. Zhou, J. Yang, C.C. Loy, Z. Liu, Learning to prompt for vision-language models, Int. J. Comput. Vis. 130 (9) (2022) 2337–2348.

[4] J. Lu, D. Batra, D. Parikh, S. Lee, Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks, in: Advances in Neural Information Processing Systems, Vol. 32, 2019.

[5] J. Duan, L. Chen, S. Tran, J. Yang, Y. Xu, B. Zeng, T. Chilimbi, Multi-modal alignment using representation codebook, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 15651–15660.

[6] L.H. Li, P. Zhang, H. Zhang, J. Yang, C. Li, Y. Zhong, L. Wang, L. Yuan, L. Zhang, J.-N. Hwang, et al., Grounded language-image pre-training, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 10965–10975.

[7] Y. Yao, A. Zhang, Z. Zhang, Z. Liu, T.-S. Chua, M. Sun, Cpt: Colorful prompt tuning for pre-trained vision-language models, 2021, arXiv preprint arXiv:2109.11797.

[8] M. Zhou, L. Yu, A. Singh, M. Wang, Z. Yu, N. Zhang, Unsupervised vision-and-language pre-training via retrieval-based multi-granular alignment, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 16485–16494.

[9] C. Yan, B. Gong, Y. Wei, Y. Gao, Deep multi-view enhancement hashing for image retrieval, IEEE Trans. Pattern Anal. Mach. Intell. 43 (4) (2020) 1445–1451.

[10] C. Yan, Y. Hao, L. Li, J. Yin, A. Liu, Z. Mao, Z. Chen, X. Gao, Task-adaptive attention for image captioning, IEEE Trans. Circuits Syst. Video Technol. 32 (1) (2021) 43–51.

[11] J. Yu, Z. Wang, V. Vasudevan, L. Yeung, M. Seyedhosseini, Y. Wu, CoCa: Contrastive captioners are image-text foundation models, Trans. Mach. Learn. Res. (2022).

[12] C. Jia, Y. Yang, Y. Xia, Y.-T. Chen, Z. Parekh, H. Pham, Q. Le, Y.-H. Sung, Z. Li, T. Duerig, Scaling up visual and vision-language representation learning with noisy text supervision, in: International Conference on Machine Learning, 2021, pp. 4904–4916.

[13] K. Zhou, J. Yang, C.C. Loy, Z. Liu, Conditional prompt learning for vision-language models, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 16816–16825.

[14] S. Manli, N. Weili, H. De-An, Y. Zhiding, G. Tom, A. Anima, X. Chaowei, Test-time prompt tuning for zero-shot generalization in vision-language models, in: Advances in Neural Information Processing Systems, 2022.

[15] R. Zhang, W. Zhang, R. Fang, P. Gao, K. Li, J. Dai, Y. Qiao, H. Li, Tip-adapter: Training-free adaption of clip for few-shot classification, in: European Conference on Computer Vision, 2022.

[16] O. Pantazis, G. Brostow, K. Jones, O. Mac Aodha, SVL-adapter: Self-supervised adapter for vision-language pretrained models, in: British Machine Vision Conference, 2022.

[17] K. Desai, J. Johnson, Virtex: Learning visual representations from textual annotations, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 11162–11173.

[18] M. Yuksekgonul, F. Bianchi, P. Kalluri, D. Jurafsky, J. Zou, When and why vision-language models behave like bags-of-words, and what to do about it? in: International Conference on Learning Representations, 2023.

[19] M. Deng, J. Wang, C.-P. Hsieh, Y. Wang, H. Guo, T. Shu, M. Song, E.P. Xing, Z. Hu, Rlprompt: Optimizing discrete text prompts with reinforcement learning, 2022, arXiv preprint arXiv:2205.12548.

[20] T. Gao, A. Fisch, D. Chen, Making pre-trained language models better few-shot learners, in: Annual Meeting of the Association for Computational Linguistics, ACL, 2021, pp. 3816–3830.

[21] Z. Jiang, F.F. Xu, J. Araki, G. Neubig, How can we know what language models know? Trans. Assoc. Comput. Linguist. 8 (2020) 423–438.

[22] B. Zhu, Y. Niu, Y. Han, Y. Wu, H. Zhang, Prompt-aligned gradient for prompt tuning, 2022, arXiv preprint arXiv:2205.14865.

[23] G. Chen, W. Yao, X. Song, X. Li, Y. Rao, K. Zhang, Prompt learning with optimal transport for vision-language models, in: International Conference on Learning Representations, 2023.

[24] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, S. Gelly, Parameter-efficient transfer learning for NLP, in: International Conference on Machine Learning, PMLR, 2019, pp. 2790–2799.

[25] J.O. Zhang, A. Sax, A. Zamir, L. Guibas, J. Malik, Side-tuning: a baseline for network adaptation via additive side networks, in: European Conference on Computer Vision, Springer, 2020, pp. 698–714.

[26] Y. Wang, Q. Yao, J.T. Kwok, L.M. Ni, Generalizing from a few examples: A survey on few-shot learning, ACM Comput. Surv. 53 (3) (2020) 1–34.

[27] C. Finn, P. Abbeel, S. Levine, Model-agnostic meta-learning for fast adaptation of deep networks, in: International Conference on Machine Learning, PMLR, 2017, pp. 1126–1135.

[28] P. Bateni, R. Goyal, V. Masrani, F. Wood, L. Sigal, Improved few-shot visual classification, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 14493–14502.

[29] H. Qi, M. Brown, D.G. Lowe, Low-shot learning with imprinted weights, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 5822–5830.

[30] Z. Lin, S. Yu, Z. Kuang, D. Pathak, D. Ramana, Multimodality helps unimodality: Cross-modal few-shot learning with multimodal models, 2023, arXiv preprint arXiv:2301.06267.

[31] I. Najdenkoska, X. Zhen, M. Worring, Meta learning to bridge vision and language models for multimodal few-shot learning, in: International Conference on Learning Representations, 2023.

[32] P.W. Koh, S. Sagawa, H. Marklund, S.M. Xie, M. Zhang, A. Balsubramani, W. Hu, M. Yasunaga, R.L. Phillips, I. Gao, et al., Wilds: A benchmark of in-the-wild distribution shifts, in: International Conference on Machine Learning, 2021, pp. 5637–5664.

[33] F. Wang, Z. Han, Y. Gong, Y. Yin, Exploring domain-invariant parameters for source free domain adaptation, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 7151–7160.

[34] J. Liang, R. He, T. Tan, A comprehensive survey on test-time adaptation under distribution shifts, 2023, arXiv preprint arXiv:2303.15361.

[35] Z. Kan, S. Chen, C. Zhang, Y. Tang, Z. He, Self-correctable and adaptable inference for generalizable human pose estimation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 5537–5546.

[36] K. Zhou, Z. Liu, Y. Qiao, T. Xiang, C.C. Loy, Domain generalization: A survey, IEEE Trans. Pattern Anal. Mach. Intell. 45 (4) (2023) 4396–4415.

[37] J. Wang, C. Lan, C. Liu, Y. Ouyang, T. Qin, W. Lu, Y. Chen, W. Zeng, P. Yu, Generalizing to unseen domains: A survey on domain generalization, IEEE Trans. Knowl. Data Eng. (2022).

[38] M. Liu, D. Zhang, S. Chen, Attribute relation learning for zero-shot classification, Neurocomputing 139 (2014) 34–46.

[39] S. Yang, Y. Wang, K. Chen, W. Zeng, Z. Fei, Attribute-aware feature encoding for object recognition and segmentation, IEEE Trans. Multimed. 24 (2021) 3611–3623.

[40] Z. Al-Halah, R. Stiefelhagen, How to transfer? zero-shot object recognition via hierarchical transfer of semantic attributes, in: IEEE Winter Conference on Applications of Computer Vision, 2015, pp. 837–843.

[41] K. Pham, K. Kafle, Z. Lin, Z. Ding, S. Cohen, Q. Tran, A. Shrivastava, Learning to predict visual attributes in the wild, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 13018–13028.

[42] E. Amid, A. Ukkonen, Multiview triplet embedding: Learning attributes in multiple maps, in: International Conference on Machine Learning, 2015, pp. 1472–1480.

[43] I. Nigam, P. Tokmakov, D. Ramanan, Towards latent attribute discovery from triplet similarities, in: IEEE/CVF International Conference on Computer Vision, 2019, pp. 402–410.

[44] C. Huang, C.C. Loy, X. Tang, Unsupervised learning of discriminative attributes and visual representations, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2016, pp. 5175–5184.

[45] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[46] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16x16 words: Transformers for image recognition at scale, in: International Conference on Learning Representations, 2020.

[47] B. Zhao, Y. Fu, R. Liang, J. Wu, Y. Wang, Y. Wang, A large-scale attribute dataset for zero-shot learning, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2019.

[48] L. Ru, Y. Zhan, B. Yu, B. Du, Learning affinity from attention: End-to-end weakly-supervised semantic segmentation with transformers, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 16846–16855.

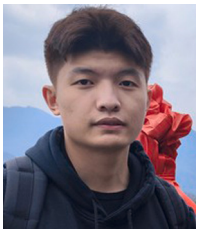
[49] T. Yu, Z. Lu, X. Jin, Z. Chen, X. Wang, Task residual for tuning vision-language models, 2022, arXiv preprint arXiv:2211.10277.

[50] L. Fei-Fei, R. Fergus, P. Perona, Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2004, pp. 178–178.

[51] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, A. Vedaldi, Describing textures in the wild, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2014, pp. 3606–3613.

[52] P. Helber, B. Bischke, A. Dengel, D. Borth, Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification, IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 12 (7) (2019) 2217–2226.
[53] S. Maji, E. Rahtu, J. Kannala, M. Blaschko, A. Vedaldi, Fine-grained visual classification of aircraft, 2013, arXiv preprint arXiv:1306.5151.
[54] M.-E. Nilsback, A. Zisserman, Automated flower classification over a large number of classes, in: Indian Conference on Computer Vision, Graphics & Image Processing, 2008.
[55] L. Bossard, M. Guillaumin, L. Van Gool, Food-101–mining discriminative components with random forests, in: European Conference on Computer Vision, 2014, pp. 446–461.
[56] B. Recht, R. Roelofs, L. Schmidt, V. Shankar, Do imagenet classifiers generalize to imagenet? in: International Conference on Machine Learning, 2019.
[57] O.M. Parkhi, A. Vedaldi, A. Zisserman, C. Jawahar, Cats and dogs, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2012, pp. 3498–3505.
[58] J. Krause, M. Stark, J. Deng, L. Fei-Fei, 3D object representations for fine-grained categorization, in: IEEE/CVF International Conference on Computer Vision Workshops, 2013, pp. 554–561.
[59] J. Xiao, J. Hays, K.A. Ehinger, A. Oliva, A. Torralba, Sun database: Large-scale scene recognition from abbey to zoo, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2010, pp. 3485–3492.
[60] K. Soomro, A.R. Zamir, M. Shah, UCF101: A dataset of 101 human actions classes from videos in the wild, 2012, arXiv preprint arXiv:1212.0402.
[61] H. Wang, S. Ge, Z. Lipton, E.P. Xing, Learning robust global representations by penalizing local predictive power, in: Advances in Neural Information Processing Systems, Vol. 32, 2019.
[62] D. Hendrycks, K. Zhao, S. Basart, J. Steinhardt, D. Song, Natural adversarial examples, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 15262–15271.
[63] D. Hendrycks, S. Basart, N. Mu, S. Kadavath, F. Wang, E. Dorundo, R. Desai, T. Zhu, S. Parajuli, M. Guo, The many faces of robustness: A critical analysis of out-of-distribution generalization, in: IEEE/CVF International Conference on Computer Vision, 2021, pp. 8340–8349.
[64] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2009, pp. 248–255.
[65] F. Wang, M. Li, X. Lin, H. Lv, A. Schwing, H. Ji, Learning to decompose visual features with latent textual prompts, in: International Conference on Learning Representations, 2023.
[66] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, arXiv preprint arXiv:1412.6980.

Yi Zhang is currently pursuing a Ph.D. in Artificial Intelligence at the Southern University of Science and Technology (SUSTech) and Harbin Institute of Technology, with an expected graduation date in July 2025. Prior to this, he earned a Bachelor's degree in Software Engineering from Northeastern University (China) and a Master's degree in Information Systems from The University of Texas. His current research focuses on vision-language models, Few-shot Learning, and Visual Reasoning.



Ce Zhang is currently pursuing his Master's degree in Machine Learning at Carnegie Mellon University and looking forward to graduating in December 2024. Before this, he earned his B.Eng. in Communication Engineering from Southern University of Science and Technology (SUSTech). His current research interests lie in vision-language models and scene understanding.

Yushun Tang is currently pursuing a Ph.D. in Intelligent Manufacturing and Robotics at the Southern University of Science and Technology (SUSTech), anticipated to complete in July 2025. He holds a Bachelor's degree in Optoelectronic Information Science and Engineering from Harbin Engineering University and a Master's degree in Electronic Science and Technology from SUSTech. His current research focuses on Computer Vision, Transfer Learning, and Domain Adaptation.

Zhihai He, chair professor of Department of Electronics of Southern University of Science and Technology, chair scholar of Changjiang Scholar (2023), the Pearl River Talents of Guangdong Province (2022), Shenzhen Overseas High level Talents (2021), IEEE Fellow (2015). Before returning to China full-time in 2021, he worked for 18 years in the Department of Electronic Engineering at the University of Missouri in the United States (2003-2021). Before leaving, he was a tenured full professor in the department and a chair professor at Robert Lee Tatum. In 2001, he obtained a doctoral degree in Electronic Engineering from the University of California, Santa Barbara. Since 2003, long-term and in-depth cutting-edge research has been conducted on artificial intelligence, the Internet of Things, and Smart Cyber Physical Systems.