This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Prompt Distribution Learning

Yuning Lu[1]*, Jianzhuang Liu[2], Yonggang Zhang[1], Yajing Liu[1], Xinmei Tian[1]†
[1]University of Science and Technology of China
[2]Huawei Noah's Ark Lab
{lyn0,yonggang,lyj123}@mail.ustc.edu.cn, liu.jianzhuang@huawei.com, xinmei@ustc.edu.cn

Abstract

We present prompt distribution learning for effectively adapting a pre-trained vision-language model to address downstream recognition tasks. Our method not only learns low-bias prompts from a few samples but also captures the distribution of diverse prompts to handle the varying visual representations. In this way, we provide high-quality task-related content for facilitating recognition. This prompt distribution learning is realized by an efficient approach that learns the output embeddings of prompts instead of the input embeddings. Thus, we can employ a Gaussian distribution to model them effectively and derive a surrogate loss for efficient training. Extensive experiments on 12 datasets demonstrate that our method consistently and significantly outperforms existing methods. For example, with 1 sample per category, it relatively improves the average result by 9.1% compared to human-crafted prompts.

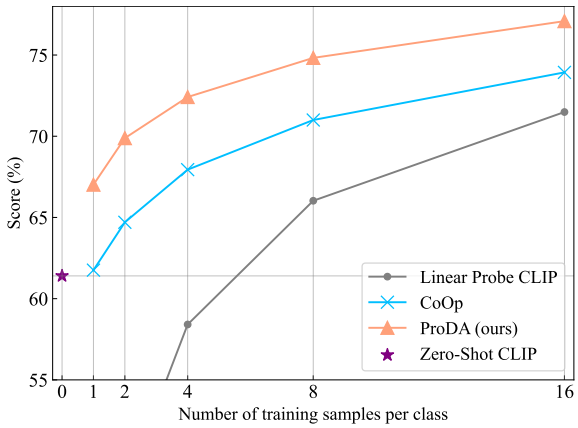


Figure 1. Comparison with existing prompt-based methods of leveraging VLM, i.e., hand-crafted prompts (zero-shot CLIP [30]) and prompt tuning (CoOp [48]), and the linear probing. We report the average results on 12 downstream datasets with various training samples. Our method ProDA consistently and substantially outperforms the previous prompt learning approaches.

1. Introduction

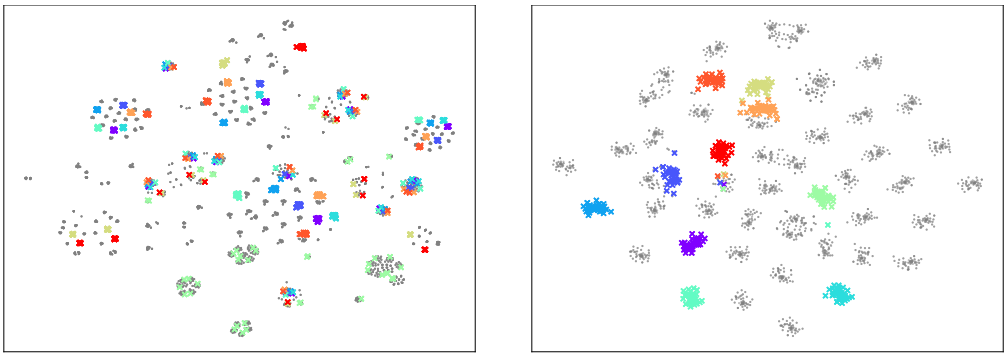
Recent progress in vision-language models (VLMs), e.g., CLIP [30] and ALIGN [16], provides a promising opportunity to explicitly leverage human language for addressing downstream recognition tasks efficiently. VLMs learn aligned embeddings of image and text via contrastive learning [4, 13, 39], encouraging the representations of an image and its language description to be similar. In the downstream task, providing the task-related content, i.e., the category descriptions, can significantly benefit the recognition via the pre-trained VLM, even to perform zero-shot inference without training samples [30].

Leveraging the language, VLMs convert the prior knowledge from humans into exploitable representations to address downstream tasks. The recognition performance of such methods is highly sensitive to the form of the provided content. However, it is still a challenging problem to determine the optimal text descriptions.

VLMs [16, 30] construct category descriptions with the hand-crafted prompt templates. A default prompt is "`a photo of a {class}.`", which works well for generic object recognition (e.g., on ImageNet [7] and STL-10 [6]). However, it is difficult to handle fine-grained object recognition. On the flower dataset (Oxford Flowers 102 [27]), a better choice of prompt is "`a photo of a {class}, a type of flower.`" [30]. In this case, the prompt word "`flower`" indicates the context of the current task, thus providing the more precise description.

From this perspective, the provided text should be adapted to the task-defined context, i.e., *low bias* to the visual representations of the target task. However, manually designing inevitably introduces artificial bias and could be sub-optimal for the target task. Thus, customizing suitable prompts for different recognition tasks relies on repetitive and time-consuming attempts by experts and also requires a large validation set for prompt selection [30].

*This work was done during an internship in Huawei Noah's Ark Lab.
†Corresponding author



(a) Input embeddings (b) Output embeddings

Figure 2. **The t-SNE [40] visualization of the descriptions for 50 random categories on ImageNet.** The descriptions of each category are generated by 80 *hand-crafted* prompts presented by CLIP [30]. For clarity, we randomly select 10 categories and highlight them with different colors. Other categories are in gray. (**a**) The input embeddings of the text encoder, which are obtained by feeding the raw text into the embedding layer. Various descriptions within a category are scattered in the space, resulting in difficulty representing their distribution. (**b**) The output embeddings of the text encoder about category descriptions. Relying on the capability of the text encoder, the output embeddings of the descriptions within a category are close to each other, allowing them to be modeled with a simple distribution. (Best viewed in color.)

Another challenge arises from the diversity of visual content. Due to inherent factors such as pose, deformation, and lighting condition, there exists significant diversity among various examples within a category [43]. This *intra-class variance* prevents a prompt from sufficiently describing visual variation. The prompts are desirable to be *diverse* and *informative*, allowing to handle the variance of visual representations. Existing work [30] ensembles 80 hand-crafted prompts to predict the categories on ImageNet [7], including "`a photo of a small {class}.`", "`a photo of a large {class}.`", *etc*. However, it still has the limitation of manual design, requiring cumbersome efforts to select an appropriate but potentially sub-optimal collection of prompts.

We present PROmpt Distribution leArning (ProDA) as a way for automatically learning diverse prompts from data, which can effectively adapt the pre-trained VLM to downstream recognition tasks. As a data-driven approach, ProDA learns the soft prompts* from a few downstream samples, discovering the task-related content with less bias than manual design. Moreover, rather than learning one soft prompt [48], our ProDA estimates a distribution over diverse and informative prompts to capture the variance of visual representations. In this way, our approach obtains better generalization to various and unknown samples (Fig. 1). Besides, we explicitly differentiate prompts on both construction and semantics to further improve their diversity.

Given the purpose of learning the prompt distribution, the challenge is how to preform the learning efficiently. Considering the soft prompt is a sequence of tokens (each token is represented by a vector), precise modeling relies on a complicated sequence generation model [3, 38], requiring a large number of target samples for training. In addition, the random nature of prompts leads to the weights of a classification model for the target task being random variables, resulting in the exact computation of the classification loss being intractable (discussed in Sec. 3.2).

To address the problem, we adopt an efficient solution which learns the distribution of the output embeddings of the prompts (with class names), i.e., the weights of the target classifier, instead of learning the distribution of the input embeddings of the prompts. The underlying intuition is that, although the various descriptions within a category are significantly different in the raw text (or low-level embeddings) (Fig. 2a), the high-level embeddings of them are usually adjacent (Fig. 2b), which can be modeled using a simple distribution, such as the multivariate Gaussian distribution in our paper. Moreover, based on the Gaussian distribution assumption, we propose a surrogate objective, an upper bound of the original optimization objective, for effective training, avoiding the intractable calculation.

We conduct large-scale experiments on 12 datasets to demonstrate the effectiveness of our method, which has a consistent and significant improvement over existing baselines. For example, ProDA with 1 sample per category relatively improves the average result by 9.1% compared to the human-crafted prompts.

2. Related Work

Vision-Language Pre-Trained Models. A promising way to build a transferable and usable recognition model is vision-language pre-training, which learns the connection between image content and language. A lot of approaches attempt to learn representations by predicting the captions

*Soft prompts, also known as continuous prompts, represent the (word) embeddings of the raw (discrete) prompts.

of images [8, 18, 22, 34, 46]. The main obstacle of them is the size of training data. The models are train on relatively small datasets (e.g., Flickr [18] and COCO Captions [8]), limiting their performances. Recently, VLMs based on contrastive learning have demonstrated impressive results by leveraging web-scale noisy image-text pairs. These methods, CLIP [30] and ALIGN [16], learn the aligned representations of image and text by the contrastive loss, which pulls the representations of matching image-text pairs close and pushes those of mismatching pairs far away. Based on natural language supervision, these VLMs not only learn powerful visual representations but are also easily transferred to various downstream tasks.

Prompt Learning. Prompt learning/engineering stems from recent advances in natural language processing (NLP). A novel prompt-based paradigm [3, 17, 21, 23, 29, 35, 36] for exploiting pre-trained language models has gradually replaced the traditional transfer approach of fine-tuning [10, 31] in NLP. The main idea of prompt learning is to formalize various NLP tasks to masked language modeling problems, which is similar to the pre-training of language models [9, 30, 32], by adopting different prompt templates. Discovering the appropriate prompt is central to this line of works. The preliminary works [3, 29, 32] elaborately design human-crafted prompts, which is known as *prompt engineering*. Since manual design is sensitive and difficult, a series of approaches [17, 36] focus on *automatically* generating desired (discrete) prompts in the natural language space. Recently, some works [12, 21, 23, 47], also known as *prompt tuning*, attempt to learn soft (continuous) prompts directly instead of searching for discrete prompts.

While prompt learning receives considerable attention in NLP, it remains underexplored in computer vision. Pre-trained VLMs [16, 30] introduce hand-crafted prompts to perform zero-shot inference on the downstream tasks. A concurrent work (CoOp [48]) adopts the prompt tuning approach of NLP, which learns a soft prompt via minimizing the classification loss on the target task. CoOp is similar to our approach in the sense that both works learn the prompt(s) in a data-driven manner. However, learning a single prompt [48] neglects the diversity of visual representations, which is challenging to capture various changes of visual content. In contrast, our method learns the distribution of diverse prompts, resulting in better generalization. We relatively improve the average results by 8.5% compared to CoOp in the 1-shot setting on 12 datasets.

3. Method

In this section, we present prompt distribution learning (ProDA), which effectively adapts a pre-trained VLM to various downstream visual recognition tasks. Without loss of generality, we adopt the public implementation of CLIP [30] as our pre-trained model.

3.1. Preliminaries

The VLM consists of an *image encoder* $f(\cdot)$ and a *text encoder* $g(\cdot)$. We denote $\mathbf{z}{=}f(\mathbf{x})/||f(\mathbf{x})||_2$ and $\mathbf{w} = g(\mathbf{t})/||g(\mathbf{t})||_2$, which are the normalized output embeddings of the image $\mathbf{x}$ and the text $\mathbf{t}$, respectively. Notice that $\mathbf{t}$ is the input embedding (of the text encoder), which is obtained by feeding the raw text to an embedding layer. During pre-training, CLIP trains the image and text encoders on massive image-text pairs by the contrastive loss, which considers matching image-text pairs as positive and mismatching pairs as negative.

Prompt Design. Given the pre-trained models $f(\cdot)$ and $g(\cdot)$, CLIP [30] performs the zero-shot inference on a downstream recognition task by manually designing the prompt template. Given the class names of the downstream task, the category descriptions $\{\mathbf{t}_c\}_{c=1}^C$ are generated with the predefined prompt, such as "`a photo of a {class}.`", where C is the number of classes. Then, we can predict the class of the test sample $\mathbf{x}$ with the prediction probability as:

$$p(y|\mathbf{x}) = \frac{e^{\mathbf{z}^T \mathbf{w}_y/\tau}}{\sum_{c=1}^C e^{\mathbf{z}^T \mathbf{w}_c/\tau}}, \tag{1}$$

where $\mathbf{z} \in \mathbb{R}^d$ and $\mathbf{w}_c \in \mathbb{R}^d$ are the normalized embeddings of the image $\mathbf{x}$ and text $\mathbf{t}_c$, respectively, d is the dimensionality of the output embedding, and τ is the temperature[†]. $\mathbf{w}_{1:C}{=}[\mathbf{w}_1^T, ..., \mathbf{w}_C^T]^T \in \mathbb{R}^{dC}$ can be considered as the weights of a linear classifier, which are used to classify the image $\mathbf{x}$.

Prompt Tuning. An alternative way to generate the weight $\mathbf{w}_{1:C}$ of the target classifier is *prompt tuning*, which learns a suitable prompt from a few samples on the target task. Prompt tuning is originally proposed to probe pre-trained language models [21]. Recently, a concurrent work [48] uses it to learn an appropriate prompt for VLMs instead of manually designing it.

Given a learnable continuous prompt $\mathbf{P} \in \mathbb{R}^{p \times e}$ with random initialization, where p is the number of tokens and e is the dimensionality of input (word) embedding, the description of each class $\mathbf{t}_c(\mathbf{P})$ is obtained via concatenating the embeddings of each class name and the prompt $\mathbf{P}$. Then, with the generated weight vector $\mathbf{w}_{1:C}(\mathbf{P}) = [\mathbf{w}_1^T(\mathbf{P}), ..., \mathbf{w}_C^T(\mathbf{P})]^T$, where $\mathbf{w}_c(\mathbf{P}) = g(\mathbf{t}_c(\mathbf{P}))$, we can learn the prompt $\mathbf{P}$ with a few training samples $\mathcal{D}^{tr} = \{(\mathbf{x}_i, y_i)\}_{i=1}^M$ by minimizing the following objective:

$$\mathcal{L}(\mathbf{P}) = \mathop{\mathbb{E}}_{\mathbf{x}_i, y_i} [-\log p(y_i|\mathbf{x}_i, \mathbf{w}_{1:C}(\mathbf{P}))] \tag{2}$$

$$= \mathop{\mathbb{E}}_{\mathbf{x}_i, y_i} \left[-\log \frac{e^{\mathbf{z}_i^T \mathbf{w}_{y_i}(\mathbf{P})/\tau}}{\sum_{c=1}^C e^{\mathbf{z}_i^T \mathbf{w}_c(\mathbf{P})/\tau}} \right], \tag{3}$$

[†]CLIP learns τ in the pre-training. We fix τ in the downstream tasks.

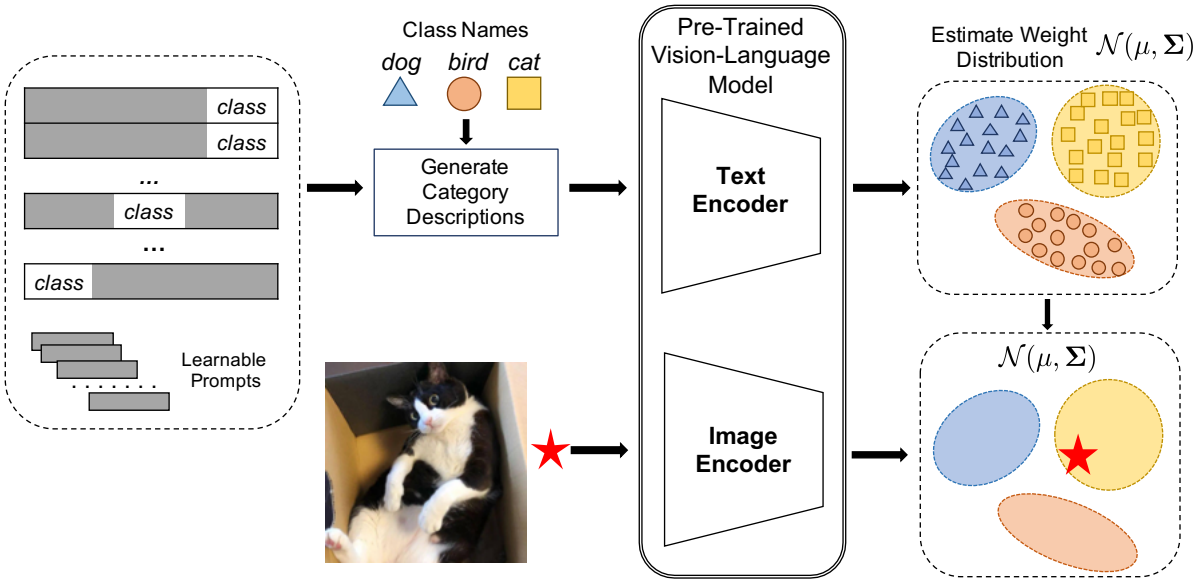


Figure 3. **Overview of the architecture of ProDA.** The class names and various learnable prompts are integrated to generate diverse category descriptions on the downstream recognition task. The output embeddings of these descriptions as the weights of linear classifiers are used to estimate the weight distribution. Given the weight distribution, we can minimize the empirical classification error and predict the classes of test samples.

where $\mathbf{z}_i$ is the normalized embedding of $\mathbf{x}_i$. Notice that all parameters of the pre-trained model are frozen during the learning process. After learning, prompt tuning leverages the learned prompt $\mathbf{P}$ to generate the target classifier and classify test samples.

3.2. Learning the Prompt Distribution

In order to handle diverse visual variations, our approach ProDA aims to learn the distribution of various prompts. Intuitively, we should learn an optimal prompt distribution $p(\mathbf{P})$, which minimizes the empirical classification loss. In this case, the classifier weights $\mathbf{w}_{1:C}(\mathbf{P})$ follow a distribution determined by $p(\mathbf{P})$ and the text encoder $g(\cdot)$, resulting in the prediction probability $p(y|\mathbf{x})$ to be a *marginal likelihood* $\mathbb{E}_{\mathbf{P}}[p(y|\mathbf{x}, \mathbf{w}_{1:C}(\mathbf{P}))]$. Unfortunately, explicitly computing this marginal likelihood is intractable, which requires the integration over $\mathbf{P}$. In a special case where $\mathbf{P}$ is a discrete random variable, the computing is possible. However, it restricts the learning for the overall prompts. Moreover, learning the exact distribution of prompts is difficult, requiring a complicated sequence generation model [3, 38].

In this work, we propose an efficient method to indirectly learn the prompt distribution by learning the distribution of the classifier weights, i.e., the output embeddings of the category descriptions. Although the original distribution of prompt $\mathbf{P}$ is complex, the generated weights $\mathbf{w}_c(\mathbf{P})$ within a category are adjacent, as shown in Fig. 2b, which can be modeled with the multivariate Gaussian distribution. Recent works [24, 25, 41, 45] show that the Gaussian distribution is effective to model the representations learned by neural networks.

Specifically, we assume $\mathcal{N}(\mu_{1:C}, \mathbf{\Sigma}_{1:C})$ is the "true" distribution of the weights $\mathbf{w}_{1:C}$. We maintain a collection of learnable continuous prompts $\mathcal{P}^K \triangleq \{\mathbf{P}_k\}_{k=1}^K$. The mean and covariance of the "true" weight distribution can be estimated from a series of classifier weights $\{\mathbf{w}_{1:C}(\mathbf{P}_k)\}_{k=1}^K$, which are generated by the prompts from $\mathcal{P}^K$. Fig. 3 illustrates the architecture of our model. Next, we propose a surrogate loss for efficient training.

Optimization. Learning the weight distribution relies on learning an optimal prompt collection $\mathcal{P}^K$. Given the weights of the K classifiers $\{\mathbf{w}_{1:C}(\mathbf{P}_k)\}_{k=1}^K$, we can estimate the mean $\mu_{1:C}(\mathcal{P}^K)$ and covariance matrix $\mathbf{\Sigma}_{1:C}(\mathcal{P}^K)$. The prompt collection is trained by minimizing the empirical classification loss:

$$\mathcal{L}(\mathcal{P}^K) = \mathop{\mathbb{E}}_{\mathbf{x}_i, y_i}\left[-\log \mathop{\mathbb{E}}_{\mathbf{w}_{1:C}} p(y_i|\mathbf{x}_i, \mathbf{w}_{1:C})\right] \tag{4}$$

$$= \mathop{\mathbb{E}}_{\mathbf{x}_i, y_i}\left[-\log \mathop{\mathbb{E}}_{\mathbf{w}_{1:C}} \frac{e^{\mathbf{z}_i^T \mathbf{w}_{y_i}/\tau}}{\sum_c e^{\mathbf{z}_i^T \mathbf{w}_c/\tau}}\right] \tag{5}$$

where $\mathbf{w}_{1:C} \sim \mathcal{N}(\mu_{1:C}(\mathcal{P}^K), \mathbf{\Sigma}_{1:C}(\mathcal{P}^K))$.

However, even with the Gaussian distribution assumption, the exact computation of the marginal likelihood is still intractable in the multi-class case [33, 42]. To address this problem, we derive an upper bound of the loss for efficient optimization.

Proposition 1 *Suppose that* $\mathbf{w}_{1:C} = [\mathbf{w}_1^T, .., \mathbf{w}_C^T]^T \in \mathbb{R}^{dC}$ *follows* $\mathcal{N}(\mu_{1:C}(\mathcal{P}^K), \mathbf{\Sigma}_{1:C}(\mathcal{P}^K))$. *Let* $\mathbf{\Sigma}_{ij}(\mathcal{P}^K)$ *be the covariance matrix of* $\mathbf{w}_i$ *and* $\mathbf{w}_j$, $\mu_i(\mathcal{P}^K)$ *be the mean of* $\mathbf{w}_i$, *and* $\mathbf{A}_{i,j} = \mathbf{\Sigma}_{ii} + \mathbf{\Sigma}_{jj} - \mathbf{\Sigma}_{ij} - \mathbf{\Sigma}_{ji}$. *Then it holds that*

$$\mathcal{L}(\mathcal{P}^K) \leq \mathop{\mathbb{E}}_{\mathbf{x}_i, y_i} \left[-\log \frac{e^{\mathbf{z}_i^T \mu_{y_i}(\mathcal{P}^K)/\tau}}{\sum_c e^{\mathbf{z}_i^T \mu_c(\mathcal{P}^K)/\tau + \mathbf{z}_i^T \mathbf{A}_{c,y_i} \mathbf{z}_i / 2\tau^2}} \right] \quad (6)$$

$$\triangleq \mathcal{L}_{upper}(\mathcal{P}^K). \quad (7)$$

The proof is provided in our supplementary materials. By minimizing $\mathcal{L}_{upper}$, our method efficiently trains the prompt collections for estimating the weight distribution, which is used to predict the classes of test image samples.

Inference. Given the learned prompts $\mathcal{P}^K$, the classifier weights $\mathbf{w}_{1:C}$ follow $\mathcal{N}(\mu_{1:C}(\mathcal{P}^K), \mathbf{\Sigma}_{1:C}(\mathcal{P}^K))$. The class of a test sample is predicted by the prediction probability $\mathbb{E}_{\mathbf{w}_{1:C}}[p(y|\mathbf{x}, \mathbf{w}_{1:C})]$. Although the explicit calculation is intractable, some numerical approximations can be used in the inference. A straightforward approach is Monte Carlo [33, 42], which requires sampling multiple classifier weights, but it results in increased inference computation. In our experiments, we find that simply using the mean of the weight distribution for classification works well, i.e., predicting by $p(y|\mathbf{x}, \mathbb{E}(\mathbf{w}_{1:C}))$. It also allows our method to have no additional computational overhead for inference.

3.3. Improving Prompt Diversity

Since the parameters of the weight distribution are estimated from the prompt collection, the quality of the prompts affects the obtained distribution. Diverse prompts can describe the visual content more sufficiently, improving the generalization on the test samples. The work [44] demonstrates that diverse classifiers are able to enhance generalization. To further improve the diversity of prompts, we explicitly differentiate the prompts of $\mathcal{P}^K$.

Position Diversity. A common way to combine the prompt and the category name is to put the category name at the end of the prompt. However, the generated text descriptions are biased. To improve the diversity of the generated text descriptions, we insert the category name in the front, middle, and end positions of different prompts. In our experiments, the proportions of these three types on $\mathcal{P}^K$ are 1/4, 1/4, and 1/2.

Semantic Orthogonality. Different prompts should represent different contents. A natural way is to encourage them to have dissimilar semantics. We feed the prompts $\{\mathbf{P}_k\}_{k=1}^K$ without incorporating the category name into the pre-trained text encoder to obtain their semantic embeddings $\{g(\mathbf{P}_k)\}_{k=1}^K$. The following semantic orthogonality loss is use to encourage the prompts to be dissimilar:

$$\mathcal{L}_{so}(\mathcal{P}^K) = \frac{1}{K(K-1)} \sum_{i=1}^{K} \sum_{j=i+1}^{K} |< g(\mathbf{P}_i), g(\mathbf{P}_j) >|, \quad (8)$$

where $< \cdot, \cdot >$ denotes the cosine similarity. Then the total training loss is:

$$\mathcal{L} = \mathcal{L}_{upper} + \lambda \mathcal{L}_{so}, \quad (9)$$

where λ is a hyper-parameter. We set $\lambda = 0.1$ for all experiments.

Algorithm 1 Pseudocode of ProDA Training.

Require: The pre-trianed VLM encoders of image f and text g
Require: The training set $\mathcal{D}^{tr}$ of the target task
Require: The input word embeddings of class names $\{\mathbf{e}_c\}_{c=1}^C$
Randomly initialize the prompt collection $\mathcal{P}^K$
for $t = 0$ **to** T **do**
  Sample a mini-batch $\{(\mathbf{x}_i, y_i)\}_{i=1}^{B_x}$ from $\mathcal{D}^{tr}$
  Compute $\mathbf{z}_i = f(\mathbf{x}_i)$, i=1, .., B_x
  Sample a mini-batch $\{\mathbf{P}_b\}_{b=1}^{B_P}$ from $\mathcal{P}^K$
  Combine the $\mathbf{P}_b$ and class name $\mathbf{e}_c$ to generate class description $t_c(\mathbf{P}_b)$, c=1, .., C; b=1, .., B_P
  Compute $\mathbf{w}_c(\mathbf{P}_b)$=$g(t_c(\mathbf{P}_b))$, c=1, .., C; b=1, .., B_P
  Let $\mathbf{w}_{1:C}(\mathbf{P}_b) = [\mathbf{w}_1^T(\mathbf{P}_b), ..., \mathbf{w}_C^T(\mathbf{P}_b)]^T$
  Compute the mean μ and covariance martix $\mathbf{\Sigma}$ of B_P vectors $\{\mathbf{w}_{1:C}(\mathbf{P}_b)\}_{b=1}^{B_P}$
  Compute $\mathcal{L}_{upper}$ according to Eq. (7)
  Compute $\mathcal{L}_{so}$ according to Eq. (8)
  Compute the total loss $\mathcal{L}$ according to Eq. (9)
  Update $\{\mathbf{P}_b\}_{b=1}^{B_P}$ by gradient descent
end for

3.4. Implementation

Unless otherwise specified, we adopt the publicly available CLIP model with the ResNet-50 [14] visual backbone as our pre-trained model (d=1024). To reduce memory consumption, we randomly sample a batch of prompts $\{\mathbf{P}_b\}_{b=1}^B$ from the prompt collection in each training iteration, instead of using all the prompts. These B prompts and C category names are coupled to generate $B \times C$ category descriptions, which are used to form B classifiers for estimating the distribution of classifier weights. Then we can minimize Eq. 9 on these B prompts. In inference, all prompts of the collection are used to estimate the distribution of the classifier weights. Besides, we approximate the covariance matrix $\mathbf{\Sigma}_{i,j}$ with the diagonal matrix to further reduce memory consumption. In this way, we train our model with 1 GPU on most datasets. On ImageNet, we adopt 4 GPU to accelerate the training. Algorithm 1 provides the pseudo-code of the training procedure.

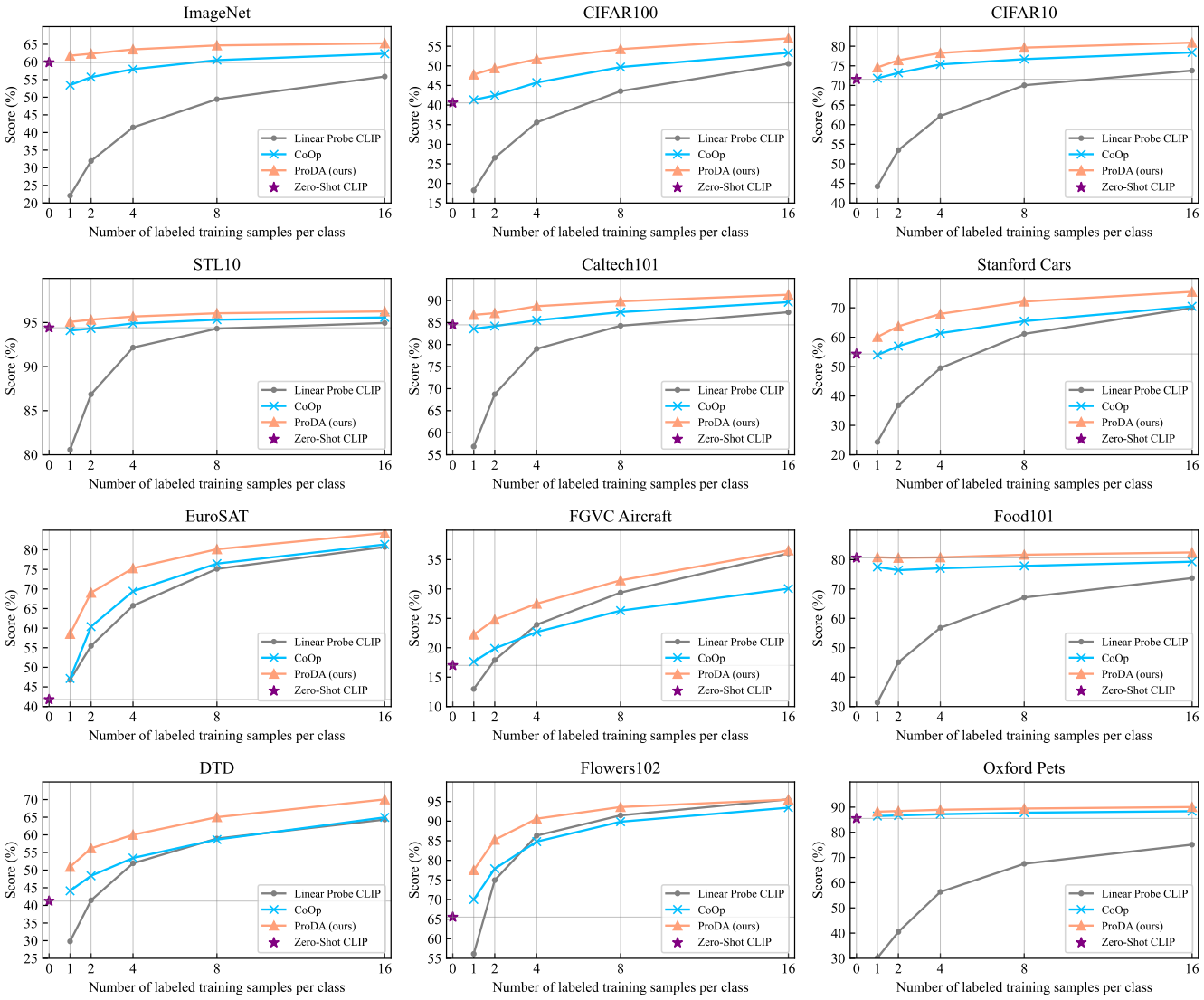


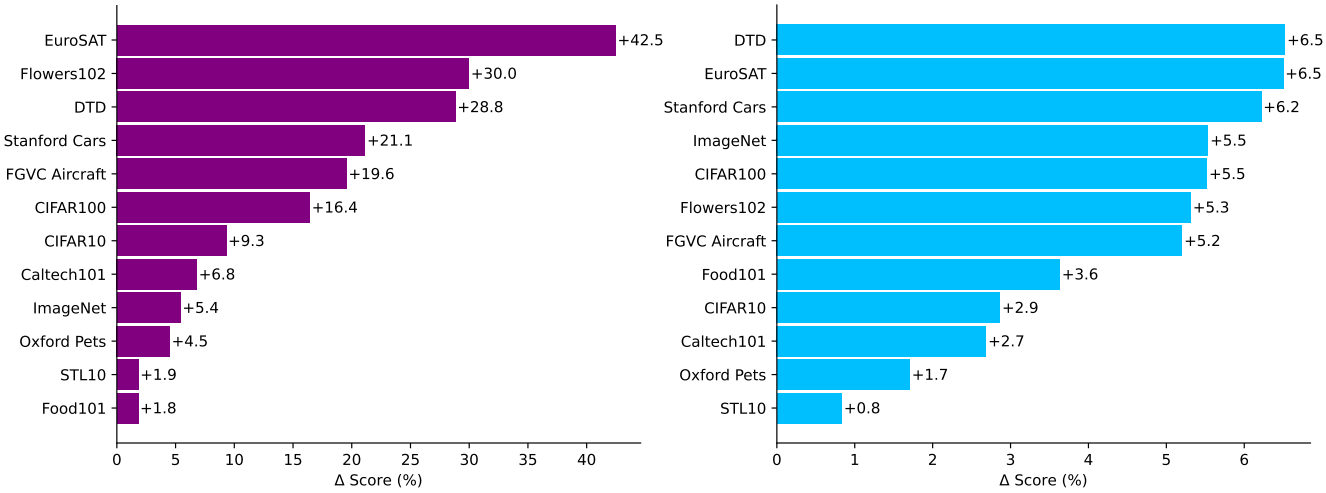
Figure 4. Comparison with two prompt-based methods and the linear probe on various downstream tasks. Our method consistently and significantly outperforms these methods.

4. Experiments

Datasets. We evaluate our ProDA on 12 downstream classification datasets, including general object recognition (ImageNet-1k [7], CIFAR-10 [20], CIFAR-100 [20], STL-10 [6], and Caltech-101 [11]), fine-grained object recognition (Oxford-IIIT Pets [28], Food-101 [2], Stanford Cars [19], Oxford Flowers 102 [27], and FGVC Aircraft [26]), remote sensing recognition (EuroSAT [15]), and texture recognition (DTD [5]). The details and evaluation metrics of each dataset are provided in the supplementary materials.

Training Details The number of tokens in each prompt and the number of prompts in the collection are set to 16 and 32, respectively. The batch size of prompts is 4. We train the prompts for 100 epochs with SGD optimizer. The momentum of SGD is 0.9. We set the learning rate using the linear scaling rule $lr \times \text{ImageBatchSize}/5$, with a base $lr = 0.001$. The batch size of images is 20 on most datasets. We use the larger batch size 100 on ImageNet. The learning rate has a cosine decay schedule. We use the model of the last training epoch for evaluation.

Baselines. We compare our approach with two existing prompt-based methods (zero-shot CLIP and prompt tuning) and the linear probe CLIP. The **zero-shot CLIP** [30] uses hand-crafted prompts to generate the target classifier on the downstream task, as discussed in Sec. 3.1. The prompt templates applied in each dataset are the same as CLIP [30]. We note that CLIP uses the full validation set



(a) ProDA vs Hand-Crafted Prompt (b) ProDA vs CoOp

Figure 5. **Comparison with prompt-based methods.** We show the *absolute improvement* of our approach compared to hand-crafted prompts [30] and prompt tuning (CoOp [48]) on each downstream task. (**a**) We compare the hand-crafted prompts to our ProDA with 16 samples per category. (**b**) Our method is compared with the prompt tuning by their average results of various shots (1, 2, 4, 8, and 16).

of each dataset, which often has thousands of samples, to manually design these prompts [30]. In addition, following the guideline of CLIP, we ensemble multiple classifiers for improving the performance of zero-shot CLIP. The prompt tuning (**CoOp** [48]) learns a soft prompt by minimizing the classification loss, which is discussed in Sec. 3.1. Our implementation obtains slightly better results than those reported in CoOp [48]. For **linear probe CLIP**, we train a logistic regression classifier on the features of training images. Existing work demonstrates that training a linear classifier on the embeddings of the pre-trained model is a strong baseline for few-shot learning [37]. Details of the baseline methods are provided in the supplementary materials.

Evaluation protocols. We follow the few-shot transfer setting on CLIP [30], which learns with 1, 2, 4, 8, and 16 labeled samples per class on each downstream task. Training examples are sampled from the training set of each dataset. After training, each method is evaluated on the full test set of the downstream task with the corresponding metric. We report the average results over 3 runs.

4.1. Main Results

Fig. 4 shows the comparison with the baseline methods on 12 downstream tasks. More detailed results are provided in the supplementary materials. The average results over all datasets are given in Fig. 1. We also provide a summary of the absolute improvement of our approach compared to the two prompt-based methods in Fig. 5. All methods adopt the same pre-trained CLIP model.

In comparison to the hand-crafted prompts (zero-shot CLIP), our approach substantially improves the performance. Our ProDA relatively improves the average results by 9.1% with 1 training sample per class and 25% with 16 training samples per class. In the uncommon datasets such as EuroSAT and DTD, the relative improvements are more significant (40% and 25% in the 1-shot setting). We consider that, for these special images (remote sensing or texture images), selecting prompts based on human experience would be more difficult and introduce more artificial bias. These results support our motivation of learning the low-bias prompt distribution.

In addition, our approach consistently and significantly outperforms CoOp. We have 8.5% relative average performance improvement in 1-shot and 4.3% in 16-shot compared with it. These results suggest the necessary of learning the distribution of diverse prompts for handling the variance of visual contents.

The comparison with linear probe CLIP demonstrates the benefit of using category names for recognition in the few-shot setting. In 1-shot, our approach has relatively 77% higher average score than the linear probe CLIP (67.0% vs 37.8%). Natural language provides dense task-related information rather than images. Our results indicate that prompt learning is an efficient way to address vision tasks.

Overall, our method ProDA substantially outperforms its prompt learning/engineer counterparts. These results demonstrate the effectiveness of our approach, which learns a low-bias and diverse prompt distribution. They also indicate that leveraging natural language to provide the task-related content can be a promising paradigm to address downstream recognition tasks efficiently.

	# of training samples per class				
	1	2	4	8	16
CoOp [48]	61.8	64.7	67.9	71.0	73.9
Ours w/o $\mathcal{L}_{upper}$	65.8	68.8	71.6	74.2	76.6
Ours w/o pos. div.	66.6	69.4	71.8	74.3	76.6
Ours w/o sem. orth.	66.8	69.6	72.2	74.5	76.8
Ours	**67.0**	**69.9**	**72.4**	**74.8**	**77.1**

Table 1. **Ablation study** of our ProDA approach. We show the average scores on 12 downstream tasks of various training samples. w/o $\mathcal{L}_{upper}$: compute the standard classification loss by treating the mean of the classifier weights as an ensemble weight, without learning the weight distribution; w/o pos. div.: all prompts are combined with the category name at the end; w/o sem. orth.: the semantic orthogonal loss $\mathcal{L}_{so}$ is not used.

		# of training samples per class				
	text bsz	1	2	4	8	16
mini-batch prompts	1×	**74.6**	**76.4**	**78.3**	79.6	80.9
all prompts	8×	74.4	76.3	**78.3**	**79.7**	**81.0**

Table 2. **Sampling mini-batch prompts.** We compare the sampling strategy with using all prompts on CIFAR-10 [20]. The "text bsz" denotes the batch size of input texts to the text encoder in each training forward.

4.2. Ablation Study

In this section, we ablate the different components in our proposed ProDA.

Weight distribution. Another way to learn diversity prompts is to aggregate the classifiers generated by multiple prompts and optimize the prompts using standard classification losses. Table 1 shows the comparison of our approach and this strategy. Our method consistently outperforms it, demonstrating the effectiveness of learning the weight distribution. We find that this strategy is also significantly better than prompt tuning [48], which supports our motivation of learning diverse prompts to capture the visual content.

Diversity constraint. Table 1 shows the effect of the prompt diversity constraints on recognition performance. Encouraging the prompts with diverse positions improve the average scores. In addition, constraining prompt semantics orthogonally also slightly improves the performance.

Number of learnable prompts. Fig. 6 shows the effect of the numbers of prompts on CIFAR-100 [20] recognition results. More prompts can improve the performance on the downstream task. Increasing the number of prompts brings more diverse descriptions, enabling sufficiently representing visual variations.

Sampling mini-batch prompts. We sample mini-batch prompts in each training iteration to reduce the memory overhead instead of using all prompts. As shown in Table 2, the sampling strategy has similar results compared to using all prompts. However, using all prompts requires eight times the size of input texts, limited by the GPU memory size.

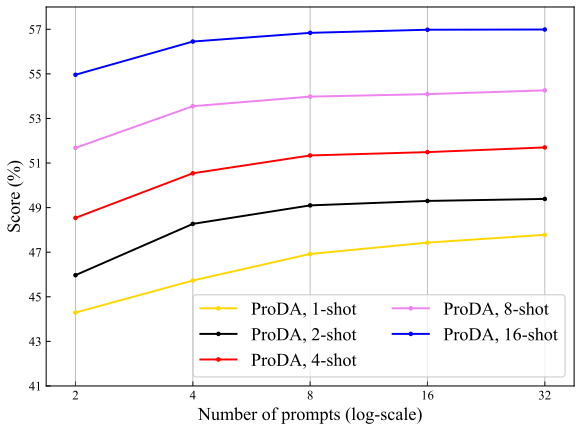


Figure 6. **Number of prompts.** We show the results of our ProDA on CIFAR-100 [20] with different shots. Larger prompt collection results in improvements. More prompts enable more precise estimation of the prompt distribution.

5. Discussion and Conclusion

This paper proposes a novel prompt learning method that learns the distribution of diverse prompts to address downstream visual recognition tasks with a pre-trained VLM. Prompt learning is naturally suited to the problem of natural language, attracting significant attention recently. We believe it is also crucial to computer vision and could be a promising way to address vision tasks efficiently. The information of an images is not as abstract as language, which exacerbates the difficulty of learning a concept with limited visual supervision. In contrast, language generated by humans has dense information and semantics. In this way, a few text descriptions are capable of providing considerable task-related content. Our method demonstrates substantial improvement over the linear probe, which is a strong baseline of few-shot learning. We hope our approach will inspire future work.

Limitation. Prompt distribution learning proposed in this paper focuses on object/image recognition. Computer vision has many other tasks, such as object detection, semantic segmentation, image style transfer, *etc*. Our current methods cannot be applied to these tasks. We believe that with dedicated modifications, our method can help some of them, which will be studied in future work.

Acknowledgement

The research was partially supported by the National Natural Science Foundation of China No. 61872329, and by MindSpore [1] which is a new deep learning computing framework.

References

[1] MindSpore. https://www.mindspore.cn/. 8

[2] Lukas Bossard, Matthieu Guillaumin, and Luc J. Van Gool. Food-101 – mining discriminative components with random forests. In *ECCV*, 2014. 6

[3] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *NeurIPS*, 2020. 2, 3, 4

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 1

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

[6] Adam Coates, Andrew Y. Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *AISTATS*, 2011. 1, 6

[7] 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. 1, 2, 6

[8] Karan Desai and Justin Johnson. Virtex: Learning visual representations from textual annotations. In *CVPR*, 2021. 3

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina N. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2018. 3

[10] Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems*, volume 32, pages 13042–13054, 2019. 3

[11] 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. *CVIU*, 2007. 6

[12] Xu Han, Weilin Zhao, Ning Ding, Zhiyuan Liu, and Maosong Sun. Ptr: Prompt tuning with rules for text classification. *arXiv:2105.11259*, 2021. 3

[13] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020. 1

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 5

[15] 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. 6

[16] 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, 3

[17] Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. How can we know what language models know. In *EMNLP*, 2020. 3

[18] Armand Joulin, Laurens van der Maaten, Allan Jabri, and Nicolas Vasilache. Learning visual features from large weakly supervised data. In *ECCV*, 2016. 3

[19] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *ICCVW*, 2013. 6

[20] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009. 6, 8

[21] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv:2104.08691*, 2021. 3

[22] Ang Li, Allan Jabri, Armand Joulin, and Laurens van der Maaten. Learning visual n-grams from web data. In *ICCV*, 2017. 3

[23] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *ACL*, 2021. 3

[24] Yajing Liu, Zhiwei Xiong, Ya Li, Yuning Lu, Xinmei Tian, and Zheng-Jun Zha. Category-stitch learning for union domain generalization. *TOMM*, 2022. 4

[25] Yajing Liu, Zhiwei Xiong, Ya Li, Xinmei Tian, and Zheng-Jun Zha. Domain generalization via encoding and resampling in a unified latent space. *TMM*, 2021. 4

[26] S. Maji, J. Kannala, E. Rahtu, M. Blaschko, and A. Vedaldi. Fine-grained visual classification of aircraft. *arXiv:1306.5151*, 2013. 6

[27] M.-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *ICVGIP*, 2008. 1, 6

[28] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *CVPR*, 2012. 6

[29] Fabio Petroni, Tim Rocktäschel, Patrick S. H. Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. Language models as knowledge bases. In *EMNLP-IJCNLP*, 2019. 3

[30] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 1, 2, 3, 6, 7

[31] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018. 3

[32] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. 2019. 3

[33] Carl Edward Rasmussen and Christopher K I Williams. *Gaussian Processes for Machine Learning*. 2005. 4, 5

[34] Mert Bulent Sariyildiz, Julien Perez, and Diane Larlus. Learning visual representations with caption annotations. In *ECCV*, 2020. 3

[35] Timo Schick and Hinrich Schütze. It's not just size that matters: Small language models are also few-shot learners. In *NAACL*, 2021. 3

[36] Taylor Shin, Yasaman Razeghi, Robert L. Logan, Eric Wallace, and Sameer Singh. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. In *EMNLP*, 2020. 3

[37] Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B. Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: A good embedding is all you need? In *ECCV*, 2020. 7

[38] Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alexander Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *SSW*, 2016. 2, 4

[39] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv:1807.03748*, 2018. 1

[40] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *JMLR*, 2008. 2

[41] Yulin Wang, Gao Huang, Shiji Song, Xuran Pan, Yitong Xia, and Cheng Wu. Regularizing deep networks with semantic data augmentation. *TPAMI*, 2021. 4

[42] C.K.I. Williams and D. Barber. Bayesian classification with gaussian processes. *TPAMI*, 1998. 4, 5

[43] Tianjun Xiao, Yichong Xu, Kuiyuan Yang, Jiaxing Zhang, Yuxin Peng, and Zheng Zhang. The application of two-level attention models in deep convolutional neural network for fine-grained image classification. In *CVPR*, 2015. 2

[44] Shaofeng Zhang, Meng Liu, and Junchi Yan. The diversified ensemble neural network. In *NeurIPS*, 2020. 5

[45] Yonggang Zhang, Mingming Gong, Tongliang Liu, Gang Niu, Xinmei Tian, Bo Han, Bernhard Schölkopf, and Kun Zhang. Adversarial robustness through the lens of causality. In *ICLR*, 2022. 4

[46] Yuhao Zhang, Hang Jiang, Yasuhide Miura, Christopher D Manning, and Curtis Langlotz. Contrastive learning of medical visual representations from paired images and text. In *arXiv:2010.00747*, 2021. 3

[47] Zexuan Zhong, Dan Friedman, and Danqi Chen. Factual probing is [mask]: Learning vs. learning to recall. In *NAACL*, 2021. 3

[48] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *arXiv:2109.01134*, 2021. 1, 2, 3, 7, 8