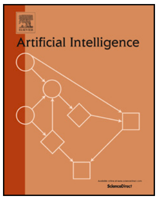
Contents lists available at ScienceDirect

Artificial Intelligence

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



Rethinking visual prompt learning as masked visual token modeling

Ning Liao [a,*], Bowen Shi [a], Xiaopeng Zhang [b], Min Cao [c], Junchi Yan [a], Qi Tian [b,*]

[a] Shanghai Jiao Tong University, Shanghai, 200240, China
[b] Huawei Inc., Shenzhen, 518129, China
[c] Soochow University, Suzhou, 215006, China

ARTICLE INFO

Keywords:
Prompt learning
Generative pre-trained model
Task reformulation
Prototypical verbalizer

ABSTRACT

Prompt learning has achieved great success in efficiently exploiting large-scale pre-trained models in natural language processing (NLP). It reformulates the downstream tasks as the generative pre-training ones to achieve consistency, thus improving the performance stably. However, when transferring it to the vision area, current visual prompt learning methods are almost designed on discriminative pre-trained models, and there is also a lack of careful design to unify the forms of pre-training and downstream tasks. To explore prompt learning on the generative pre-trained visual model, as well as keeping the task consistency, we propose Visual Prompt learning as masked visual Token Modeling (VPTM) to transform the downstream visual classification task into the pre-trained masked visual token prediction task. In addition, we develop the prototypical verbalizer for mapping the predicted visual token with implicit semantics to explicit downstream labels. To our best knowledge, VPTM is the first visual prompt method on the generative pre-trained visual model, which achieves consistency between pre-training and downstream visual classification by task reformulation. Experiments show that VPTM outperforms other visual prompt methods and achieves excellent efficiency. Moreover, the task consistency of VPTM contributes to the robustness against prompt location, prompt length and prototype dimension, and could be deployed uniformly.

1. Introduction

Large-scale pre-trained models (PMs) have greatly promoted the development in the computer vision (CV) field [1–4]. The common paradigm is firstly pre-training, then fine-tuning the entire model with different task-specific objectives in downstream applications, which is prohibitive. Such a significant problem also arises in the natural language processing (NLP) field and is even trickier due to the larger scales of pre-trained language models.

To mitigate the issue in the paradigm, namely “pre-train, then fine-tune” [5–8] in NLP, a new paradigm, namely “pre-train, prompt, then predict” [9] has been proposed [10–12]. Based on the generative pre-trained language models, *the core technology is to reformulate downstream tasks to be the same form as the pre-training language modeling tasks*, as shown in Fig. 1 (a). In this way, when PMs are applied to downstream tasks, the knowledge of PMs can be naturally exploited with same objectives as in pre-training tasks, and contributes to better performance stably.

[*] Corresponding authors.
E-mail addresses: liaoning@sjtu.edu.cn (N. Liao), tian.qi1@huawei.com (Q. Tian).

https://doi.org/10.1016/j.artint.2025.104417
Received 12 December 2024; Received in revised form 7 June 2025; Accepted 1 September 2025

Available online 10 September 2025

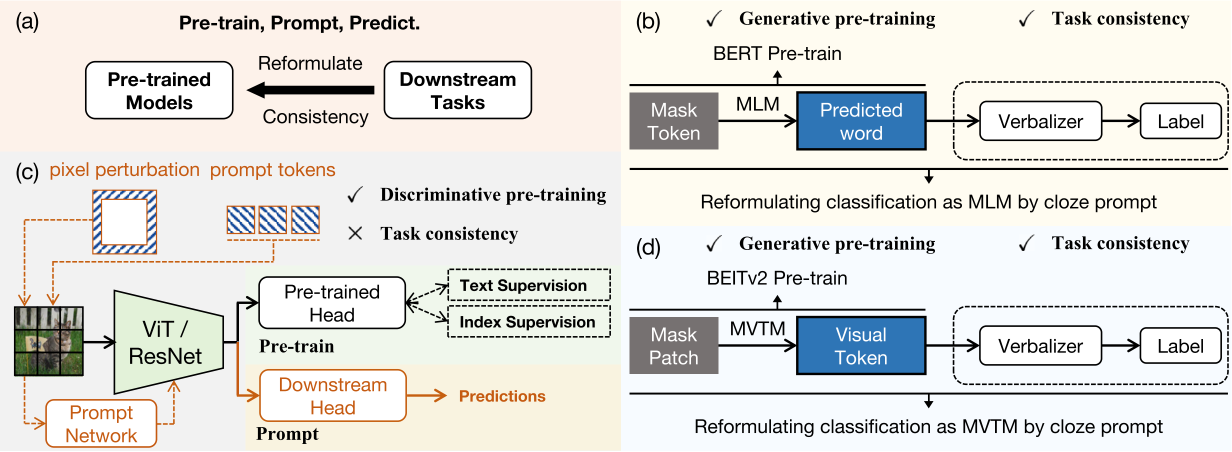


Fig. 1. **(a)** Prompt learning in NLP reformulates downstream tasks as *generative* pre-training tasks to keep consistency. **(b)** Taking BERT [13] as an example, the downstream classification task is reformulated as the pre-trained masked language modeling (MLM) task by cloze prompt. The predicted word in the masked place is then mapped to downstream labels by the verbalizer. **(c)** Current visual prompt learning methods are almost designed on discriminative pre-trained models, including supervised pre-trained ViT [20] and ResNets [21] with index supervision, contrastively pre-trained ViT in CLIP [22] with text supervision. There are three typical ways in current visual prompt learning methods: i) concatenating prompt tokens to the image patch tokens (VPT [23]); ii) adding pixel-wise perturbation as prompts (VP [24], ILM-VP [25], EVP [26]); iii) learning prompt network (PGN [27]). When prompt tuning in downstream applications, the objectives of these methods are different from the pre-training ones, causing the lack of task consistency. **(d)** To design a visual prompt learning method on generative pre-trained model as well as keeping the task consistency, our method reformulates the downstream visual classification task as the generative masked visual token modeling (MVTM) pre-training task in BEITv2 [28]. The predicted visual token in masked place is mapped to downstream labels by the verbalizer.

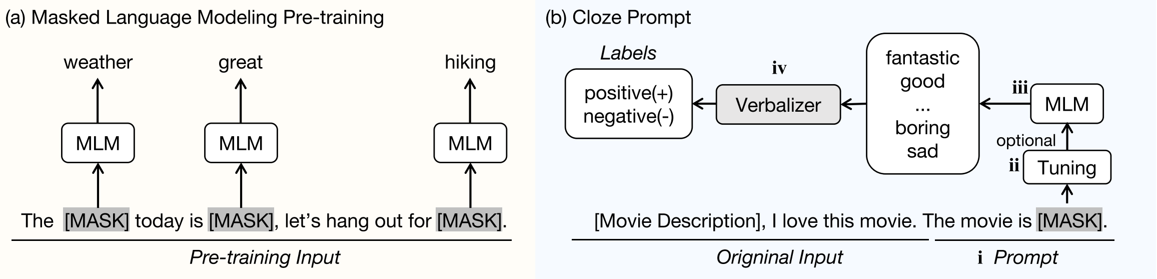


Fig. 2. (a) Language models such as BERT [13] take the masked language modeling (MLM) task for pre-training. (b) To be consistent with the MLM pre-training task, the text classification task is reformulated by the cloze prompt. The verbalizer maps the words predicted from the mask to downstream labels.

Taking the masked language modeling (MLM) pre-training task as an example [13,14], as shown in Fig. 1 (b) and Fig. 2 (a), classification tasks in NLP are usually reformulated by cloze prompt, which follows four steps [15,16] in Fig. 2 (b): (i) adding prompts with masks to the original input; (ii) performing prompt tuning optionally; (iii) predicting the word in the masked place from the vocabulary by MLM; (iv) mapping the predicted word to downstream labels using verbalizer. Specifically, the predicted words are usually not the downstream labels, thus verbalizer [17–19,12] is devised to establish connections between them, e.g., the verbalizer in Fig. 2 (b) maps "fantastic, good" to "positive". As such, prompt learning can be well applied in solving tasks such as text classification, named entity recognition, etc.

Witnessing the success of prompt learning in NLP, such a technique has been introduced into vision applications [29,27,30]. VPT [23] prepends a few parameterized embeddings as prompts to the input sequence of ViT [20], which has been supervised pre-trained on ImageNet-21k [31] in a discriminative way. Visual prompting (VP) [24] modifies the pixel space with learnable parameters to perform visual prompt learning on CLIP [22], which has been pre-trained by contrastive learning. Till now, the current visual prompt learning methods [27,30,26,29] are all designed on discriminative pre-trained models shown in Fig. 1 (c). *There lacks prompt learning method carefully designed for the generative pre-trained visual model.* Particularly, regardless of the efforts paid on adding prompts in the input space [23,24,26,25], learning prompt networks [27] or designing prompt blocks [29], *unifying the forms of pre-training and downstream applications by task reformulation to achieve consistency remains unexplored.* In view of the improved performance, efficiency and stability brought by the task consistency of prompt learning in NLP, we aim at the task consistency of generative visual prompt learning by inheriting the generative pre-training task to the downstream visual classification task.

BERT [13] tokenizes texts into textual tokens within the vocabulary, BEIT-series models [32,28] tokenize image patches into visual tokens within the codebook. Since text classification task could be reformulated as masked token prediction task based on contextual tokens, on the basis of the same task and data formula as in tokens, perhaps image classification could also be reformulated as masked token prediction based on contextual tokens. In this way, the pre-training and visual classification could also be unified in the visual space. Regarding the output space, textual tokens naturally equip with rich semantics, while visual tokens more carry low-level information rather than high-level semantics. Therefore, the last step that image classification could be reformulated as masked token prediction is learning a semantic-rich visual codebook. For this, the selected visual model is the generative model BEITv2 [28], in which the visual tokens of masked patches are predicted from the semantic-rich codebook in pre-training inspired by BERT [13]. In this paper, we propose Visual Prompt learning as masked visual Token Modeling (VPTM) for the visual classification task, as shown in Fig. 1 (d). Specifically, we concatenate continuous prompts and pre-trained mask token to the input sequence in prompt tuning. *The downstream classification is achieved by mapping the prediction in the masked place, as in the pre-training phase, to the downstream labels by the verbalizer*. Considering that the semantics of visual tokens are implicit and constructing a verbalizer manually is intractable, we introduce the prototypical verbalizer into VPTM inspired by NLP [33,34].

Experimentally, VPTM outperforms other visual prompt learning methods [23–25,27,26] with better efficiency. Extensive experiments show the consistency between pre-training and downstream visual classification contributes to the robustness against learning strategies for different datasets, prompt locations, prompt length, and prototype dimensions. As a result, the VPTM equips with the capability of unified development. Our contributions include:

1) We propose a visual prompt learning method, which reformulates the visual classification as the generative masked visual token modeling task. To the best of our knowledge, it is the first visual prompt learning method on the generative pre-trained visual model and achieves a close match between the forms of downstream and pre-training tasks.

2) For mapping predicted visual tokens with implicit semantics to downstream labels, we introduce the prototypical verbalizer into the vision area to construct the mapping rule, instead of manual construction.

3) Extensive experiments validate the task consistency achieved by reformulation in visual prompt tuning contributes to better performance and training stability, as well as robustness against prompt locations, prompt lengths and prototype dimensions.

2. Related work

2.1. Prompt learning in NLP

Reformulating downstream tasks as pre-training tasks. Considering a limited application of fine-tuning the entire large-scale pre-trained model [5,35,13] for downstream tasks, and the impaired performance due to the gap between pre-training and fine-tuning tasks, researchers in NLP pioneered prompt learning [10,12], in which the downstream tasks are reformulated in the same form as the pre-training ones, and only a few additional prompt-relevant parameters are optimized [11,36]. Specifically, based on the generative *masked language modeling* pre-training task [13], the text classification [37,38,18], named entity recognition [39] and commonsense reasoning [40] are transformed into *cloze prompt*, in which prompts with the mask token are concatenated with the original input. The tokens predicted in masked places by the pre-training task are then mapped to the answers by the verbalizer. Based on the generative *casual language modeling* pre-training task [35], the question answering [41,42], text generation [35,43,44] and automatic evaluation of text generation [45] are reformulated via *prefix prompt*, in which a prefix string is prepended to the original input for autoregressive answer text generation as in pre-training. *Our method is inspired by the core idea in prompt learning, i.e., reformulating the downstream tasks as the pre-training task to keep their consistency.*

Verbalizer. In the cloze prompt, the predicted words in masked places are usually not the actual labels. To map the predicted words to labels, handcrafted verbalizer [18,38] was initially proposed by manually designed rules. To avoid the expert dependence and prediction bias in handcrafted verbalizer, the method [46] uses gradient descent to search the mapping. KPT [17] incorporates external knowledge bases into verbalizer for text classification. Soft verbalizer [47,48] regards each label as a trainable token, which is optimized together with prompt tuning. Prototypical verbalizer [33,34] learns prototype vectors, which represent classes, as the verbalizer. The similarity between masked embedding and prototypes is adopted as the classification rule, i.e., the mapping from generated words to downstream labels. However, the semantic meaning of visual token in codebooks is implicit. Manually designing the mapping rule requires the explicit semantic meaning as the language words equipped with. Thus, it is inapplicable in visual prompt learning. Inspired by the prototypical verbalizer, we introduce it into our method to solve the problem of constructing mapping between visual tokens with implicit meaning and downstream labels.

2.2. Visual prompt learning

As an effective and efficient alternative technology of fine-tuning, prompt learning has been introduced into unimodal vision area [49–51,27,30,52–56]. There are three typical ways in visual prompt learning. The first is to concatenate learnable prompt tokens to the image patch sequences in Transformer-based visual models [20]. For example, VPT [23] is a representative visual prompt method based on supervised pre-trained ViT, which optimizes the prepended prompt-relevant parameters together with the newly added classification head for downstream visual tasks including classification. LPT [57] concatenates shared prompts for all classes and group-specific prompts to image patch sequence for long-tailed image classification. The second typical way is to learn pixel perturbation as prompts. Visual prompting (VP) [24] introduces learnable pixel perturbation on the image encoder of CLIP [22] as prompts to be optimized. Based on VP [24], EVP [26] learns more diversified pixel perturbation as prompts by applying data augmentations. The third typical way is to learn prompt networks. Pro-tuning [29] adapts pre-trained ResNets [21] to downstream tasks by introducing lightweight prompt blocks. PGN [27] learns a network to generate the prompts conditioned on input image. However, these methods are all designed on the discriminative pre-trained visual models. Prompt learning on generative pre-trained visual models by keeping consistency between pre-training and downstream applications remains unexplored. In this paper, we concentrate on *prompt learning on generative pre-trained unimodal vision model, and reformulating the downstream visual classification task as the pre-training one to achieve task consistency.*

2.3. Masked modeling pre-training

Masked language modeling (MLM) is a representative generative pre-training task in language models such as BERT [13] and ERNIE [14]. With masking pieces of the input sentences, it aims at predicting the masked text pieces based on the context, as a result of which the comprehensive understanding ability is equipped for the pre-trained model. Motivated by MLM, masked image modeling (MIM) was proposed to boost the visual pre-trained models [58–60]. MAE [58] masks patches randomly and reconstructs the missing pixels with a lightweight decoder. SimMIM [61] adopts a large masked patch size, and performs raw pixel value regression with a lightweight prediction head during the pre-training phase. MFM [62] transfers the reconstruction from the spatial domain to the frequency domain to reveal underlying image patterns. Different from the above methods that focus on learning low-level features, EVA [63] concentrates on high-level semantic learning, and is pre-trained by reconstructing the vision features of the masked patches extracted by CLIP [22] conditioned on visible ones. Similar to EVA, MCM [64] masks the input feature maps extracted by the CLIP [22] model and reconstructs the missing features for learning semantics. Except for the above MIM methods without specific visual tokenization, BEIT [32] was proposed to predict the visual tokens, which are tokenized by the codebook of DALL-E [65], of the masked patches. However, BEIT [32] is limited to learn the low-level features in the codebook of DALL-E. To solve the drawback of semantic-less representations of BEIT [32], BEITv2 [28] was then proposed with a semantic-rich tokenizer guided by CLIP [22] or DINO [2]. The pre-training strategies of BEIT and BEITv2 are the same. As the visual tokens are supposed to be equipped with high-level semantics as the language words in the cloze prompt, we propose the visual prompt learning in consistency with masked visual token modeling on BEITv2.

3. Preliminary

Before elaborating our method, we first introduce the masked language modeling (MLM) pre-training task in Sec. 3.1. The cloze prompt in NLP is then presented in Sec. 3.2, which inspires us significantly, including motivation and technical design. After that, we introduce the masked visual token modeling (MVTM) pre-training task of BEIT-series models [32,28] in Sec. 3.3, which is the foundation of our method.

3.1. Masked language modeling pre-training

Masked language modeling (MLM) is a representative pre-training task in language models, e.g., BERT [13]. Given a tokenized sentence $\boldsymbol{s} = [t_1, t_2, ..., t_n]$ consisting of n tokens t_i from a corpora C, MLM randomly masks 15% tokens from $\boldsymbol{s}$, and gets the masked sentence $\boldsymbol{s}^{\mathcal{M}}$, in which $\mathcal{M}$ is the index set of masked tokens. Based on $\boldsymbol{s}^{\mathcal{M}}$, MLM predicts the original tokens v of the `[MASK]` token, as shown in Fig. 2 (a). The objective of the MLM pre-training task is formulated as:

$$\mathcal{L} = -\sum_{s \in C} \sum_{i \in \mathcal{M}} \log p(v_i | \boldsymbol{s}^{\mathcal{M}}). \tag{1}$$

After the MLM pre-training, the model is equipped with the comprehensive language understanding ability based on the context, which lays the foundation for the cloze prompt, a classical prompt format in NLP.

3.2. Cloze prompt in NLP

In the “pre-train, then fine-tune” paradigm, the pre-trained language models (LMs) are applied to downstream tasks by fine-tuning all parameters according to different task-specific objectives, which causes expensive computation costs and task discrepancy. Taking the MLM pre-trained model BERT [13] as an example, if the downstream tasks can be resolved in the cloze format, in which the desired output is predicted from the masked places given the input as the context, the LMs can be easily applied to different downstream tasks in the uniform way by which they have been pre-trained.

Based on this idea, the cloze prompt is designed on MLM pre-trained models [13,14] to fit downstream tasks that require *word-level* outputs, as shown in Fig. 2 (b). The key design is to transform the task-specific predictions as the generative predictions in the `[MASK]` token. Given the original input sentence tokenized as $\boldsymbol{s} = [t_1, t_2, ..., t_n]$, the pre-trained `[CLS]` token, `[MASK]` token, and prompt text $\boldsymbol{P_T}$ are concatenated with $\boldsymbol{s}$ to get the transformed input $\boldsymbol{s}^t$:

$$\boldsymbol{s}^t = [\texttt{[CLS]}, \mathtt{t}_1, \mathtt{t}_2, ..., \mathtt{t}_\mathtt{n}, \boldsymbol{P_T}, \texttt{[MASK]}]. \tag{2}$$

The prompt text $\boldsymbol{P_T}$ can be categorized into two types: unoptimizable discrete prompts that consist of real words in the vocabulary, and optimizable continuous prompts composed of virtual parameterized embeddings. The unoptimizable discrete prompts are usually adopted at zero-shot predictions, while the optimizable continuous prompts are usually tuned with the pre-trained model kept frozen specifically on downstream datasets, known as “prompt tuning”.

After getting the predicted word Y_p from the `[MASK]` token in $\boldsymbol{s}^t$ by MLM, the downstream task is finally achieved through the mapping from Y_p to the final labels Y using the verbalizer $\boldsymbol{V}$, which is formulated as:

$$Y = \boldsymbol{V}(Y_p). \tag{3}$$

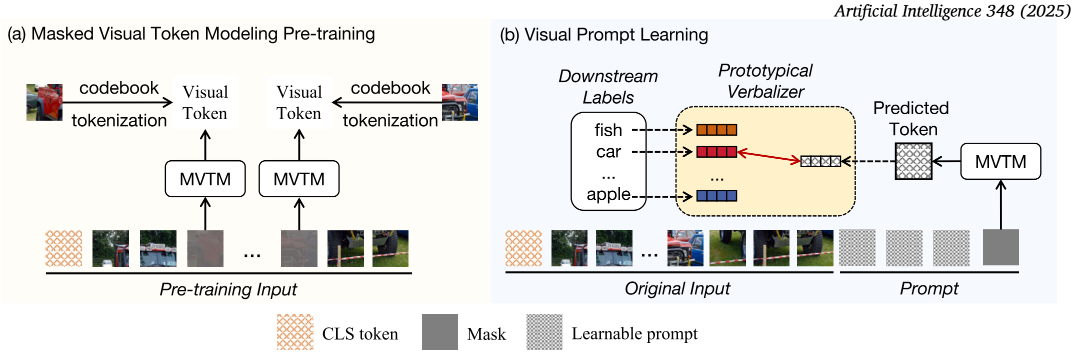


Fig. 3. (a) BEIT-series models [32,28] are pre-trained by predicting the visual tokens of the masked patches. (b) The proposed VPTM reformulates the visual classification task as the generative masked visual token modeling (MVTM) task with pre-trained BEITv2. The proposed prototypical verbalizer constructs the connection between predicted visual token with implicit semantics and the downstream labels. The positions that the prompts and masks relative to the original input are ablating studied in experiments.

3.3. *Masked visual token modeling pre-training*

Following BERT [13], BEIT-series (*BERT Pre-Training of Image Transformers*) models [32,28] are pre-trained by the generative masked visual token modeling (MVTM). Specifically, each image $\boldsymbol{x}$ in dataset $\mathcal{D}$ is processed into patches. Then, all patches are tokenized into visual tokens within the codebook. In the pre-training phase, part of the patches indexed within a set $\mathcal{M}$ is replaced with `[MASK]`. The model is trained to predict the visual token z of the `[MASK]` patches in the masked image $\boldsymbol{x}^{\mathcal{M}}$, as shown in Fig. 3 (a). The pre-training loss is:

$$\mathcal{L} = -\sum_{x\in\mathcal{D}}\sum_{i\in\mathcal{M}} \log p(z_i|\boldsymbol{x}^{\mathcal{M}}). \tag{4}$$

As can be seen from Eq. (1) and Eq. (4), based on the high similarity between MLM and MVTM pre-training tasks, and inspired by the cloze prompt in NLP, we propose the visual prompt learning method, namely VPTM, which is the first prompt method designed on the generative pre-trained visual model as well as keeping the task consistency.

4. Method

In this section, we elaborate the proposed method VPTM, which reformulates the downstream visual classification task as the generative masked visual token modeling (MVTM) pre-training task. The overview of our method is shown in Fig. 3. We first introduce the task reformulation in Sec. 4.1. The prototypical verbalizer for mapping the predicted visual token to downstream labels is devised in Sec. 4.2.

4.1. *Visual prompt learning as MVTM*

Based on the masked visual token modeling (MVTM) pre-training in Fig. 3 (a), we propose the visual prompt learning method VPTM, as shown in Fig. 3 (b). Each image $\boldsymbol{x} \in \mathbb{R}^{H\times W\times C}$ is firstly processed into N patches $\{\boldsymbol{x}_i^p\}_{i=1}^N$ with $N = HW/P^2$ and the patch size as $P \times P$. H, W is the image resolution, C is the number of channels. The patches are transformed into d-dimension patch embeddings $\boldsymbol{e}_i \in \mathbb{R}^d$ with positional encoding by the pre-trained BEITv2 [28] through an embedding function $\texttt{Embed}_{pre}$:

$$\boldsymbol{e}_i = \texttt{Embed}_{pre}(\boldsymbol{x}_i^p), i = 1, 2, ..., N. \tag{5}$$

The classification token $\boldsymbol{e}_{\texttt{[CLS]}}$ of the pre-trained model is then prepended in the front of the sequence of patch embeddings to obtain the original input, and keep the same token interactions as in pre-training.

Instead of adding a new prediction head on the classification token to perform downstream tasks as current visual prompt methods [23–25] do, *our method reformulates the visual classification task as the MVTM task*. To preserve the image information, we concatenate the `[MASK]` token $\boldsymbol{e}_{\texttt{[MASK]}}$, which is also from the pre-trained model, to the original input sequence. Besides, for continuously tuning towards downstream datasets, we insert additional N_p parameterized learnable prompts $\boldsymbol{p}_i \in \mathbb{R}^d, i \in [1, N_p]$ into the sequence, the dimension of $\boldsymbol{p}_i$ is the same as the hidden dimension of the pre-trained BEITv2 [28]. Then, we obtain the final input sequence $\boldsymbol{H}_{vp}$:

$$\boldsymbol{H}_{vp} = [\boldsymbol{e}_{\texttt{[CLS]}}, \boldsymbol{e}_1, ..., \boldsymbol{e}_N, \boldsymbol{p}_1, ..., \boldsymbol{p}_{N_p}, \boldsymbol{e}_{\texttt{[MASK]}}]. \tag{6}$$

The positions of the prompts $\boldsymbol{p}_i$ and `[MASK]` token $\boldsymbol{e}_{\texttt{[MASK]}}$ related to the original image patches are ablating studied in Sec. 5.7 in experiments.

As such, the input sequence $\boldsymbol{H}_{vp}$ in Fig. 3 (b) bottom is similar to the input sequence in the pre-training stage in Fig. 3 (a), in which each sequence includes the `[MASK]` token. The input sequence $\boldsymbol{H}_{vp}$ can be regarded as the context for predicting the visual token that supposed to be in the masked place by MVTM. Our method fulfills the cloze format prompt in vision area, which is similar to the cloze prompt in NLP, as shown in Fig. 2 (b). After feeding $\boldsymbol{H}_{vp}$ into the pre-trained model, we get the embedding of the [MASK] token denoted as $\boldsymbol{h}_{\texttt{[MASK]}} \in \mathbb{R}^d$. To achieve visual classification by MVTM, the last step is to map the visual token to downstream labels, which is introduced in Sec. 4.2.

4.2. Prototypical verbalizer

In the vision area, the visual tokens are equipped with implicit semantic meaning. Designing the mapping rule from the visual token to downstream labels manually is intractable. To solve the mapping problem, we propose the prototypical verbalizer inspired by [33,34].

For each class, we devise the corresponding learnable prototype vector $\boldsymbol{c}_k \in \mathbb{R}^t, k \in [1, N_C]$, in which N_C is the number of downstream classes and t is the dimension of the prototype vector. After getting the embedding of the [MASK] token $\boldsymbol{h}_{\texttt{[MASK]}} \in \mathbb{R}^d$, we project it into the prototype space using a linear function $\mathcal{F} : \mathbb{R}^d \to \mathbb{R}^t$, then get vector $\boldsymbol{u}_{\texttt{[MASK]}} \in \mathbb{R}^t$:

$$\boldsymbol{u}_{\texttt{[MASK]}} = \mathcal{F}(\boldsymbol{h}_{\texttt{[MASK]}}). \tag{7}$$

The mapping from the visual token to downstream labels is transformed as the similarity between vector $\boldsymbol{u}_{\texttt{[MASK]}}$ and each prototype vector. The similarity between an image $\boldsymbol{x}_i$ and its class C_i with prototype $\boldsymbol{c}_i$ is thus calculated as:

$$\texttt{sim}(\boldsymbol{x}_i, C_i) = \boldsymbol{u}^{i}_{\texttt{[MASK]}} \cdot \boldsymbol{c}_i^T, \tag{8}$$

where T is the transpose manipulation.

For a batch of N images, the loss is:

$$\begin{aligned}\mathcal{L}_{vp} &= -\frac{1}{N}\sum_{i=1}^{N} \log \frac{\exp(\texttt{sim}(\boldsymbol{x}_i, C_i))}{\sum_{k=1}^{N_C} \exp(\texttt{sim}(\boldsymbol{x}_i, C_k))} \\ &= -\frac{1}{N}\sum_{i=1}^{N} \log \frac{\exp(\boldsymbol{u}^{i}_{\texttt{[MASK]}} \cdot \boldsymbol{c}_i^T)}{\sum_{k=1}^{N_C} \exp(\boldsymbol{u}^{i}_{\texttt{[MASK]}} \cdot \boldsymbol{c}_k^T)}.\end{aligned} \tag{9}$$

Overall, by inheriting the MVTM task, the learnable visual prompts and class prototypes are optimized to obtain the dataset-specific visual prompts and construct the mapping relationship from the prediction in the [MASK] place to downstream labels. In the prompt tuning phase, all parameters in the pre-trained vision model are kept frozen.

5. Experiments

5.1. Datasets

To evaluate the performance of the proposed VPTM, we select 10 datasets in our experiments. The visual prompts are learned on the training sets and evaluated on the test sets. In the datasets, CIFAR100 [66] includes 100 classes, CIFAR10 [66] includes 10 classes, Oxford Flowers102 [67] includes 102 classes, Food101 [68] includes 101 classes, EuroSAT [69] includes 10 classes, SVHN [70] includes 10 classes, Oxford Pets [71] includes 37 classes, DTD [72] includes 47 classes, Resisc45 [73] includes 45 classes, Patch Camelyon (PatchCame) [74] includes 2 classes.

5.2. Baseline methods

We compare our method with other downstream fine-tuning and visual prompt methods as baselines:

(1) *Fine tune (FT).* Optimizing the entire model.

(2) *Linear probe (LP).* Only optimizing the classification head on the [CLS] token.

(3) *Prompting on the image encoder of CLIP.* As the codebook of BEITv2 is distilled from CLIP, here we set some prompting methods designed on the image encoder of CLIP as baselines for direct comparisons. They are: a) fine-tuning CLIP; b) linear probing on CLIP; c) using textual prompt (TP); d) TP + visual prompt (VP) [24], which adds perturbation on the pixel; e) TP + PGN [27], which generates prompts for input; f) EVP [26], which adds prompts on the pixel with improved generalization; g) ILM-VP [25], which adds prompts on the pixel and learns a label mapping.

(4) *Visual prompt tuning (VPT)* [23]. Optimizing parameters in prompts and the newly added classification head for the downstream task. The prompts are prepended only at the first layer of the model. The classification head is devised on the [CLS] token of the last layer of the model.

Additionally, our method focuses on *unimodal visual prompt learning*. Regarding the *initialization* of the prototypes, as will be introduced in the implementation details in the Sec. 5.3, *the prototypes are all initialized as zeros*. Regarding the *optimization* of the prototypes, according to the Eq. (9), the *prototypes are all optimized purely by the cross-entropy loss* based on the similarity metric calculated between the projected token and prototypes. Therefore, although we set prototypes for each class, we do not introduce any text-level information on it. Thus, we do not compare with multi-modal prompt methods [75–77] performed on texts. We also could not compare with multi-modal prompt methods that the visual and textual prompts are learned jointly and could not be separated, such as UPT [78] and MaPLe [79].

Table 1
The accuracy comparisons between our method and baseline methods. LP: Linear probe. FT: Fine tune. TP: Textual prompt.

Methods	CIFAR100	CIFAR10	Flowers	Food101	EuroSAT	SVHN	Pets	DTD	Resisc45	PatchCame	Average
FT+BEITv2	92.35	98.94	99.01	92.70	99.28	98.11	93.92	81.65	97.51	87.42	94.09
LP+BEITv2	78.13	93.50	85.57	81.91	96.76	63.11	90.16	69.47	89.75	81.03	82.94
FT+CLIP	82.10	95.80	97.40	80.50	97.90	95.70	88.50	72.30	94.40	N.R.	89.40
LP+CLIP	80.00	95.00	96.90	84.60	95.30	65.40	89.20	74.60	66.00	N.R.	83.00
TP+CLIP	63.10	89.00	61.90	79.80	40.00	5.10	85.90	43.00	42.40	N.R.	56.69
TP+VP+CLIP	75.30	84.20	70.30	78.90	96.40	88.40	85.00	57.10	81.40	N.R.	79.67
TP+PGN+CLIP	79.30	96.10	**94.00**	82.50	98.00	94.20	**91.50**	71.50	92.10	N.R.	88.80
EVP	**81.20**	96.60	82.30	84.10	**98.70**	90.50	90.00	68.40	**92.30**	N.R.	87.12
VPTM (Ours, 100e)	80.15	**98.20**	91.35	**84.28**	98.57	**91.76**	91.25	**76.81**	90.86	N.R.	**89.25**
ILM-VP	N.R.	94.40	83.70	79.10	96.90	91.20	N.R.	63.90	N.R.	N.R.	84.87
VPTM (Ours, 100e)	N.R.	**98.20**	**91.35**	**84.28**	**98.57**	**91.76**	N.R.	**76.81**	N.R.	N.R.	**90.16**
VPT+BEITv2 (100e)	**83.12**	96.71	89.46	**88.29**	94.06	90.46	88.88	74.31	90.81	83.04	87.91
VPTM (Ours, 50e)	79.43	97.12	90.67	82.65	98.37	91.35	**91.41**	75.96	**90.86**	81.29	87.91
VPTM (Ours, 100e)	80.15	**98.20**	**91.35**	84.28	**98.57**	**91.76**	91.25	**76.81**	**90.86**	**83.91**	**88.71**

5.3. *Implementation details*

We experiment with the BEITv2 [28], which has been pre-trained on ImageNet-1k [31] by masked visual token modeling and the codebook is guided by CLIP [22]. Main experiments of our method are performed on NVIDIA Tesla V100 GPU with batch size 64. We use the AdamW optimizer with the weight decay set as 0.01 and the momentum set as 0.9. The base learning rate is 0.001. We use the cosine scheduler with 5 warm up epochs. The prompts and prototypes are all initialized as zeros. Fine-tuning and linear probing on BEITv2 [28] are performed for 50 epochs following the official code.[1] We implement VPT on the weights of BEITv2 for a fair comparison.[2]

5.4. *Comparison to baseline methods*

The accuracy comparisons between our method and baselines are shown in Table 1. In the 6-th row from the bottom, Patch Camelyon is not considered for a fair comparison with TP+CLIP, TP+VP+CLIP, TP+PGN+CLIP and EVP. In the 4-th row from the bottom, CIFAR100, Pets, Resisc45 and Patch Camelyon are not considered for a fair comparison with ILM-VP. In the bottom row, all benchmarks are counted for a fair comparison with VPT.

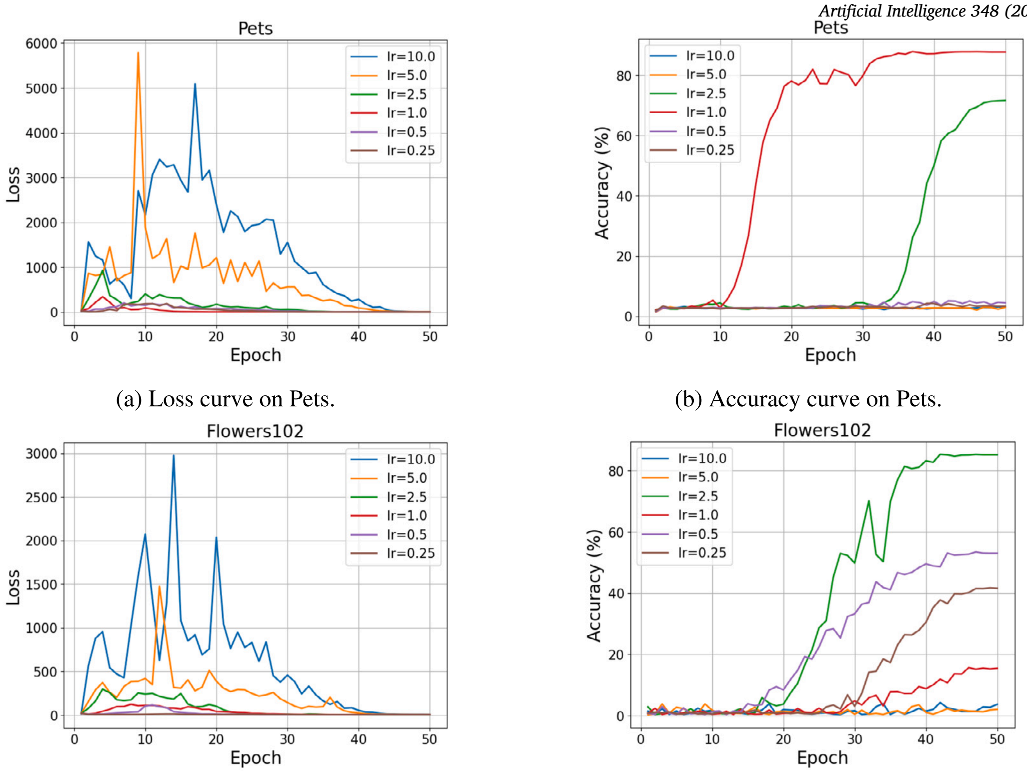
Comparison with FT & LP on BEITv2. By only optimizing 0.17% parameters of the entire model on average, which will be discussed in Sec. 5.5, our method is inferior to fine-tuning the entire model. In NLP, for example, BERT is pre-trained with the vocabulary consisting of 30000 tokens. Based on that, prompt learning which inherits MLM pre-training exhibits competitive performance to fine-tuning. We partially attribute it to the rich granularity, high semantics, and large vocabulary. However, in the field of vision, the codebook consists only of 8192 tokens. The current visual codebook is not as high quality and granularity as the vocabulary in NLP, which causes the prediction space to be not fine-grained enough, limiting the performance. On the other hand, our method consistently surpasses linear probe on all datasets more than about 5% on average. In particular, our method outperforms the linear probe by nearly 30% on SVHN dataset. These indicate that reformulating the classification task as the MVTM pre-training task in our method is superior to conventionally performing classification on the `[CLS]` token by linear probing.

Comparison with prompting methods on the image encoder of CLIP. Our method achieves competitive performance compared with fine-tuning the entire CLIP [22]. Concerning the results of VP [24] and PGN [27] combined with textual prompts, our method achieves the best average performance even without the assistance of texts. In addition, compared with EVP [26] and ILM-VP [25], which directly perform prompt learning on the image encoder of CLIP, our method still exhibits a great performance advantage. Hence, compared with the above baseline methods which add image perturbation in the pixel or learn a network to generate prompts, our method is more effective.

Comparison with VPT [23]. VPT and our method both concatenate learnable tokens with the original input patch sequence in prompt learning. Based on BEITv2, the only difference between VPT and our method is that our method inherits the pre-trained MVTM task by task reformulation, while VPT does not. For a fair comparison, we implement VPT-shallow on BEITv2 for 100 epochs following the official setting of VPT [23]. The prompt length of each dataset is the same as the optimal setting in VPT. By tuning VPTM for only 50 epochs, our method could achieve average performance comparable to that of VPT under 100 epochs. The accuracy of VPTM is better on 7 out of 10 datasets. By tuning VPTM for 100 epochs, it outperforms VPT on the average accuracy by nearly 1 point, and achieves better performance on 8 out of 10 datasets.

Furthermore, it is worth mentioning that VPT severely relies on the hyperparameters of learning strategy. As reported in the paper of VPT [23], different datasets adopt different parameters of the learning rate and the weight decay. Given that, when implementing

[1] https://github.com/microsoft/unilm/tree/master/beit2.
[2] VPT is officially performed on supervised pre-trained ViT [20].



(a) Loss curve on Pets.

(b) Accuracy curve on Pets.

(c) Loss curve on Flowers.

(d) Accuracy curve on Flowers.

Fig. 4. Loss and accuracy curves with different hyperparameters in VPT [23]. The deployment of VPT is complex and time-consuming. The optimal learning rates on different datasets are different. One optimal learning rate for a dataset can cause failed learning on others. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

Table 2
Result comparisons between the proposed VPTM and VPT. The prompt length is set as optimal for each dataset, or 10 uniformly. The VPT is implemented using the same learning strategy as VPTM. The accuracy gap is the difference between the results achieved under the two sets of prompt length.

	Methods	CIFAR100	CIFAR10	Food101	EuroSAT	Pets
Prompt-Len (Optimal)	VPT+ BEITv2 (100e)	28.76	43.71	59.92	72.76	4.69
	VPTM (Ours, 50e)	79.43	97.12	82.65	98.37	**91.41**
	VPTM (Ours, 100e)	**80.15**	**98.20**	**84.28**	**98.57**	91.25
Prompt-Len ($N_p = 10$)	VPT+ BEITv2 (100e)	67.06	78.05	81.53	78.91	59.55
	VPTM (Ours, 50e)	**72.09**	**96.54**	**81.76**	**97.06**	**88.42**
Accuracy Gap	VPT+ BEITv2 (100e)	38.30	34.34	21.61	6.15	54.86
	VPTM (Ours, 50e)	7.34	0.58	0.89	1.31	2.99

VPT on BEITv2, we search the optimal hyperparameters within $[50.0, 25.0, 10.0, 5.0, 2.5, 1.0, 0.5, 0.25, 0.1]$ for the learning rate and $[0.0, 0.01, 0.001, 0.0001]$ for the weight decay. Within the 36 sets of hyperparameters, it is observed that one set of hyperparameters suitable for one dataset usually causes failed learning on other datasets. Taking Oxford Pets and Oxford Flowers102 datasets as examples, as shown in Fig. 4, different learning rates result in various performance on the training loss and test accuracy. Moreover, the accuracy on Flowers with the same learning rate as Pets (i.e., 1.0) is about 70% lower than that with the optimal learning rate of 2.5. In comparison, the proposed VPTM are uniformly tuned with one set of hyperparameters across all datasets, and is easy to be deployed. We infer that the insensitivity of VPTM to hyperparameters specific on different datasets is due to its inheritance of the pre-training task. Based on the consistency between the pre-training and reformulated downstream tasks, the knowledge of the pre-trained model could be stably exploited in downstream applications. *In short, we can effectively and stably gain the performance advantage without complex process in searching the optimal training hyperparameters.*

In addition, we make comparisons between our method and VPT under the same learning strategies, i.e., setting the learning rate, weight decay, and momentum as 0.001, 0.01, 0.9 for all datasets. By setting the prompt length as optimal for each dataset, i.e., keeping the same setting corresponding to that in Table 1, or 10 uniformly, the results are shown in Table 2. Our method achieves the best performance and exhibits decisive superiority under the two settings of prompt length. Moreover, compared with the results

Table 3
The comparisons between the proposed VPTM and VPT [23] on prototype dimension t, prompt length N_p, GFLOPs, the ratio of the amount of tuned parameters to the entire parameters, and the cost time of prompt tuning for 50 epochs in the optimal setting.

Index		CIFAR100	CIFAR10	Flowers	Food101	EuroSAT	SVHN	Pets	DTD	Resisc45	PatchCame	Average
VPT	Prompt-Len (N_p)	100	100	200	100	50	200	50	1	50	5	85.6
	GFLOPS	26.99	26.99	36.78	26.99	22.24	36.78	22.24	17.67	22.24	18.04	25.70
	Tuned / Total(%)	0.18	0.10	0.27	0.18	0.05	0.19	0.08	0.04	0.09	0.01	**0.12**
	Time (s)	6056.21	5946.02	268.89	10370.18	1990.44	14178.10	508.11	321.58	1283.25	17239.00	5816.18
VPTM	Proto-Dim (t)	128	128	256	128	64	64	64	256	128	256	–
	Prompt-Len (N_p)	20	100	20	20	100	100	50	10	50	20	**49.0**
	GFLOPS	19.53	27.09	19.53	19.53	27.09	27.09	22.34	18.60	22.34	19.53	**22.27**
	Tuned / Total(%)	0.14	0.19	0.23	0.14	0.14	0.14	0.10	0.23	0.15	0.23	0.17
	Time (s)	6025.61	6149.32	230.67	10458.09	1925.40	10133.13	472.66	320.04	1254.46	15876.48	**5284.59**

Table 4
The comparisons between the MLP-1, MLP-2 and the prototypical verbalizer (PV).

Index		CIFAR100	CIFAR10	Flowers	Food101	EuroSAT	SVHN	Pets	DTD	Resisc45	PatchCame	Average
MLP-1	Tuned / Total(%)	0.10	0.09	0.10	0.10	0.09	0.09	0.07	0.05	0.08	0.02	0.08
	Accuracy (%)	67.07	97.03	71.85	67.32	96.52	90.14	79.18	73.24	77.11	77.87	79.73
MLP-2	Tuned / Total(%))	0.14	0.19	0.17	0.14	0.19	0.19	0.15	0.18	0.15	0.15	0.17
	Accuracy (%)	74.42	97.01	81.92	75.04	97.09	91.16	90.19	76.81	90.52	79.39	85.36
PV	Tuned / Total(%)	0.14	0.19	0.23	0.14	0.14	0.14	0.10	0.23	0.15	0.23	0.17
	Accuracy (%)	79.43	97.12	90.67	82.65	98.37	91.35	91.41	75.96	90.86	81.29	**87.91**

of VPT in Table 1, under the optimal setting of the prompt length, the accuracy on CIFAR100 drops from 83.12 to 28.76 significantly. VPT is shown to be quite sensitive to the learning strategies per dataset, and unsatisfying results could be caused when adopting a unified learning strategy for all datasets.

Moreover, VPTM achieves a far lower accuracy gap between setting the prompt length as the optimal one and 10, compared with that of VPT. The proposed VPTM is shown to be much more robust against the prompt length than VPT, which will be further ablating studied in Sec. 5.8.

In conclusion, based on the BEITv2 model, reformulating visual classification as the masked visual token modeling pre-training to achieve task consistency in visual prompt learning is rational. Compared with VPT, the proposed VPTM is shown to be more effective in performance, more stable in tuning, and more robust against the prompt length.

5.5. Parameter efficiency validation

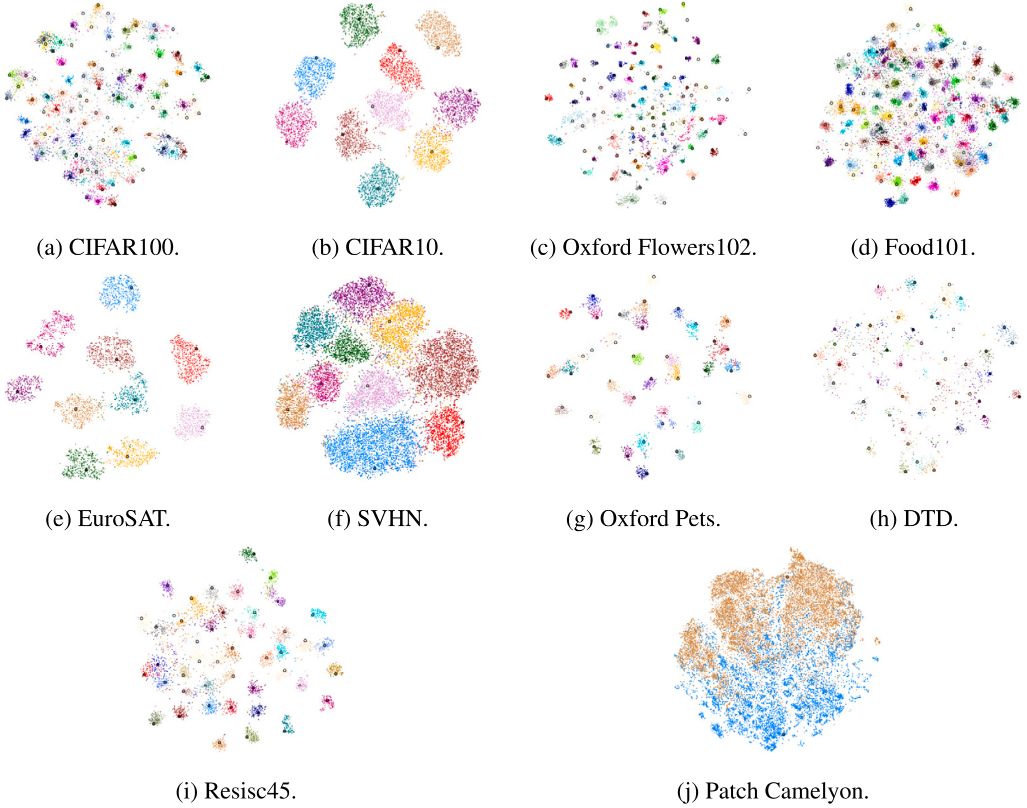
To validate the parameter efficiency of VPTM, we compare our method with VPT under the optimal setting corresponding to Table 1 on prompt length N_p, GFLOPs, the ratio of the amount of tuned parameters to the entire parameters $Tuned/Total$, and the cost time of prompt tuning for 50 epochs. Besides, the prototype vectors are also counted as optimized parameters in VPTM, we show the prototype dimension t together in Table 3.

Regarding N_p, the average prompt length of VPTM is almost half of that of VPT. The average GFLOPs of VPTM is lower than that of VPT by 3.43. Due to the existence of the parameters in verbalizer and prototypes, regarding the ratio $Tuned/Total$, the value of VPTM is 0.05% higher than that of VPT. *Though VPT tunes relatively fewer parameters than VPTM, VPTM is still more efficient from the GFLOPs comparison.* We analyze the reason as that the calculation cost is mostly caused by the token interaction. VPT requires two times of prompt tokens compared with our method, which results in more calculation burden. To compare the efficiency of the proposed VPTM and the VPT more intuitively, we show the cost time of prompt tuning for 50 epochs. From the comparison, the average time cost for prompt tuning of the proposed VPTM is more than 500 seconds shorter than that of VPT, it further validates that the VPTM is much more efficient than VPT. Therefore, by inheriting the pre-training task to keep consistency, VPTM is proved to be more efficient and requires much less resource consumption.

5.6. Effectiveness validation of the prototypical verbalizer

To validate the effectiveness of the prototypical verbalizer, we replace it with the 1-layer and 2-layer MLP added on the prediction in the masked place to perform classification. The 1-layer MLP (MLP-1) directly maps the prediction in the masked place to the number of classes. The 2-layer MLP (MLP-2) firstly maps the prediction in the masked place to a 128 dimensional vector, which is then mapped to the number of classes. Accuracy and the ratio of the amount of tuned parameters to the entire parameters $Tuned/Total$ are delivered in Table 4.

Compared with using the prototypical verbalizer, when using MLP-1, the amount of optimized parameters is fewer, and the average accuracy is lower by 8.18%. When increasing the amount of optimized parameters to be the same as that when using the prototypical verbalizer, the average accuracy achieved by using MLP-2 is still lower by 2.55%. The results demonstrate that mapping



(a) CIFAR100. (b) CIFAR10. (c) Oxford Flowers102. (d) Food101.

(e) EuroSAT. (f) SVHN. (g) Oxford Pets. (h) DTD.

(i) Resisc45. (j) Patch Camelyon.

Fig. 5. Visualizations of the prototypes and the transformed tokens $\boldsymbol{u}_{\texttt{[MASK]}}$ in testing phase using TSNE. Different colors represent different classes. The triangles with dark circles are the prototypes.

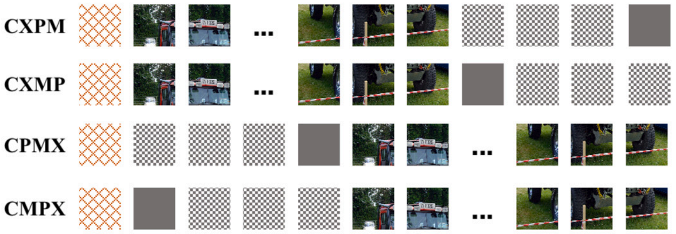


Fig. 6. Four sets of positions that prompts and `[MASK]` token relative to the original input. “C”: CLS token; “X”: image patches; “P”: prompt tokens; “M”: mask.

the prediction in the masked place to the classes is inferior to the prototypical verbalizer. In the pre-training phase, the `[MASK]` token is supervised by the visual token. By inheriting the pre-training task, in our method, the prediction in the masked place is supposed to be like a word in the language vocabulary, but not equipped with explicit semantic meaning. It is not a comprehensive representation as the `[CLS]` token. To achieve classification in this method, the rational way is to search for a mapping between the predicted token and downstream labels, but not to regard the `[MASK]` as a comprehensive representation and simply conduct classification on it by adding MLPs.

In addition, to see the details of the prototypical verbalizer in constructing the mapping from predictions in the masked place to downstream labels, we visualize the distributions of the prototypes $\boldsymbol{c}_k \in \mathbb{R}^t, k \in [1, N_C]$ and the transformed tokens $\boldsymbol{u}_{\texttt{[MASK]}}$ by TSNE, as shown in Fig. 5. For datasets that exhibit high accuracy, such as CIFAR10, EuroSAT and Resisc45, the transformed tokens $\boldsymbol{u}_{\texttt{[MASK]}}$ predicted from the testing samples distribute tightly with their corresponding prototypes. For datasets on which the accuracy is not so high, including CIFAR100 and DTD, the prototypes can be separated clearly. There exists some overlap on the transformed tokens from different classes. We infer the reason as the low granularity of the visual tokens in the codebook.

5.7. Ablation on position of prompts and `[MASK]` *token*

To show the impact of the position of prompts and `[MASK]` token relative to the original input, we ablate 4 sets of positions that the prompts and `[MASK]` token relative to the original input, as shown in Fig. 6 The position relationships are represented by the

Table 5
The ablation study on the positions that the prompts and `[MASK]` token relative to the original input. The order in the strings indicates the position relationships. "C": `[CLS]` token; "X": image patch embeddings; "P": prompts; "M": `[MASK]` token.

Positions	CIFAR100	CIFAR10	Flowers	Food101	EuroSAT	SVHN	Pets	DTD	Resisc45	PatchCame	Average
CXPM	79.43	96.74	90.52	82.65	98.37	81.80	90.35	75.96	90.86	80.39	86.71
CXMP	73.77	96.90	90.58	81.84	97.30	90.43	90.62	75.27	89.75	81.29	86.78
CPMX	72.79	97.12	90.57	82.02	97.09	90.05	91.22	75.80	88.46	78.33	86.35
CMPX	72.55	96.95	90.67	82.64	97.09	91.35	91.41	75.59	88.63	79.65	86.65

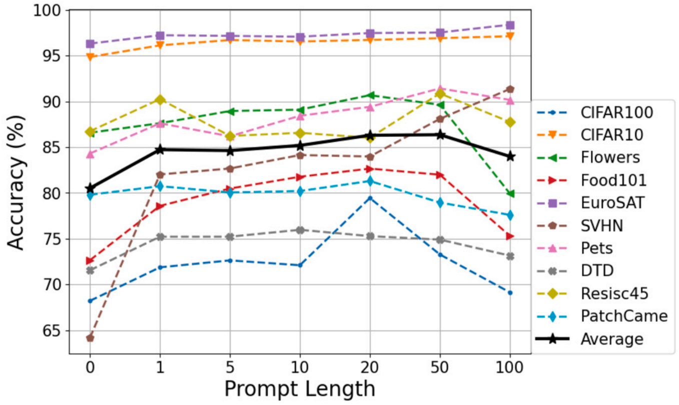


Fig. 7. Accuracy under different settings of prompt length.

order of their abbreviations. Based on the optimal setting in Table 1, the ablation results are shown in Table 5. The greatest margin on the average accuracy is only 0.43. We analyze the reasons of the stable performance as follows: 1) 40% patches within each image are randomly block-wisely masked for pre-training, so that the pre-trained model can achieve relatively stable predictions regardless of the position of `[MASK]` token; 2) more importantly, the VPTM inherits the pre-training task, bringing the robustness of VPTM against the positional changes.

5.8. Ablation on prompt length N_p

Setting the prompt length $N_p \in [0, 1, 5, 10, 20, 50, 100]$, results are compared in Fig. 7.

When $N_p = 0$, which refers to only the verbalizer works, the lowest average accuracy 80.51 is achieved. The results validate the necessity of introducing learnable prompts.

When $N_p > 0$, most datasets such as EuroSAT and CIFAR10 are less likely to be impacted by the prompt length. Regarding the average accuracy as shown in the dark line in Fig. 7, the largest margin between the highest ($N_p = 50, Acc = 86.35$) and lowest ($N_p = 100, Acc = 83.98$) average accuracy is 2.37. Our method exhibits stable performance against the change of prompt length. Moreover, the average accuracy when $N_p = 100$ (except $N_p = 0$) is the lowest, while the results with shorter prompt length are even better. It indicates that our method does not rely on longer prompts, i.e., more parameters that can be optimized.

5.9. Ablation on prototype dimension t

Under the setting of Table 1, the comparisons on the dimension of prototypes $t \in [64, 128, 256]$ are given in Fig. 8. Almost equal performance is achieved on datasets such as CIFAR10 and SVHN. The highest average accuracy 86.41 is achieved with $t = 256$. Quite close to the highest one, the average accuracy when $t = 128$ is 86.40. When $t = 64$, the average accuracy 84.26 is the lowest. Our method also performs stably under different dimensions of prototypes. Considering the parameter-efficiency and the performance comprehensively, setting the dimension as 128 is optimal.

5.10. Exploration of the role of `[MASK]` token

To further explore the role of `[MASK]` token in prediction, we perform experiments with the same setting in the Table 1 to compare the performance when predicting from the `[CLS]` token and from a learnable token. Specifically, we select the learnable token as the first prompt token from left to right. All experiments are performed by prompt tuning for 50 epochs. Results are given in Table 6. Predictions from the `[MASK]` token exhibit a significant performance advantage of 3 points over those from the `[CLS]` token and the learnable token. The results validate the effectiveness of prediction from `[MASK]` token in the proposed VPTM.

In the pre-training phase of BEITv2, the mask token inherently acquires the capability to reconstruct masked regions based on the contexts. Building upon this, visual token sequences can be treated analogously to textual sequences in BERT, enabling the exploitation

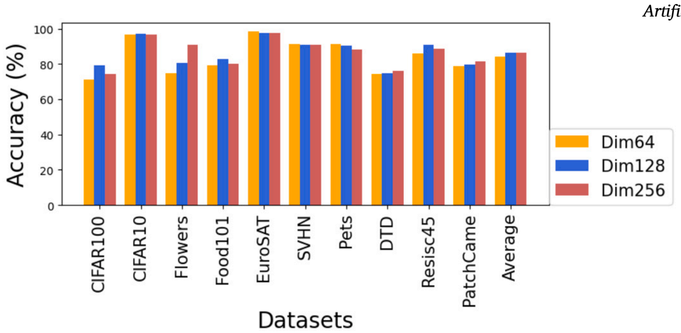


Fig. 8. Accuracy under different prototype dimensions.

Table 6
The ablation study on prediction from the `[MASK]` token, `[CLS]` token and the learnable token, which is selected as the first prompt token from left to right.

Prediction Token	CIFAR100	CIFAR10	Flowers	Food101	EuroSAT	SVHN	Pets	DTD	Resisc45	PatchCame	Average
`[MASK]` Token	**79.43**	**97.12**	**90.67**	**82.65**	**98.37**	**91.35**	**91.41**	**75.96**	**90.86**	81.29	**87.91**
`[CLS]` Token	77.52	92.96	82.19	78.05	97.85	84.36	89.65	72.13	90.57	84.31	84.96
Learnable Token	74.52	92.40	81.08	75.69	98.30	83.62	89.00	73.62	90.82	**85.13**	84.42

Table 7
The comparison of results of our method by using the weights of BEIT [32] and BEITv2 [28]. “IN”: “ImageNet”.

Methods	CIFAR100	CIFAR10	Flowers	Food101	EuroSAT	SVHN	Pets	DTD	Resisc45	PatchCame	Average
VPTM + BEIT (IN-21k)	15.22	53.95	10.98	13.31	86.43	37.96	7.93	22.23	57.27	74.92	38.02
VPTM + BEITv2 (IN-1k)	79.43	97.12	90.67	82.65	98.37	91.35	91.41	75.96	90.86	81.29	87.91

of sequence modeling relationships to achieve classification predictions via mask prediction. In NLP, it is very intuitive to use prompts like “I love this movie. The movie is `[MASK]`.” to realize classification through mask prediction. The same is true for the visual field. The visible image patch sequence itself is part of the contextual information, and is analogous to “I love this movie.”. Besides, the prompts are optimized to construct the information background for classification, which is similar to “The movie is `[MASK]`.”. Finally, the image classification task could be realized through mask prediction.

Therefore, The information that the mask token learns in the prompt tuning phase is the classification-driven semantic token prediction. The role that the mask token plays is to predict the desired information within the codebook based on the contextual information. It is noteworthy that the contextual information includes not only visible image patches to be classified, but also the prompts that are optimized to construct the contextual background targeting at image classification.

5.11. *Significance of semantics of codebook*

To demonstrate the significance of the codebook’s semantics, we compare the results of our method by using the weights of BEIT [32] or BEITv2 [28]. BEIT [32] has been pre-trained on ImageNet-21k [31] with the codebook from DALL-E [65], while BEITv2 [28] has been pre-trained on ImageNet-1k [31] with the codebook guided by CLIP [22]. The results are given in Table 7.

Even though BEIT has been pre-trained on ImageNet-21k, the results of it are far worse than those of BEITv2, which has been pre-trained on ImageNet-1k. The codebook of pre-trained BEIT is from DALL-E, which enforces the model to recover low-level information of the image patches. In contrast, BEITv2 aims at exploring a semantic-rich visual tokenizer, which promotes the pre-training from learning the low-level pixel-wise features to the high-level semantic-wise features. The results validate the significance of the high-level semantics of the codebook.

Aiming at achieving the excellent performance of image classification by inheriting the masked visual token modeling pre-training, the visual tokens are supposed to be like the words in the vocabulary in NLP and carry high semantics.

Taking a step further, there are only 8192 visual tokens in codebook in BEITv2, which are far fewer than words in vocabulary in NLP. To enrich the granularity of the visual tokens, pre-training with a larger codebook is required in the future.

6. Discussions on the backbone dependence

Following the core design of keeping consistency between downstream tasks and pre-training ones, VPTM is exactly suitable for BEITv2 [28]. Specifically, the large language models in NLP [13,14] almost take the language modeling as pre-training task. Therefore, the *cloze prompt* can be applied on language models that have been pre-trained by the *masked language modeling* task, the *prefix prompt* can be applied on language models that have been pre-trained by the *casual (autoregressive) language modeling* task, for keeping the task consistency. In comparison, the pre-training tasks in vision area are various, e.g., supervised pre-training [20], masked image

Table 8
The performance of applying the proposed VPTM on SimMIM and EVA model.

Pre-train Model	CIFAR100	CIFAR10	Flowers	Food101	EuroSAT	SVHN	Pets	DTD	Resisc45	PatchCame	Average
SimMIM [61]	22.68	67.44	11.12	34.66	95.63	44.81	14.16	23.62	46.51	80.40	44.08
EVA [63]	83.64	99.20	74.19	88.66	98.52	94.84	79.19	73.19	84.40	86.74	86.26

modeling (MIM) [58,60], and masked visual token modeling (MVTM) [32,28]. The supervised pre-trained models are not equipped with generative task, and MIM pre-training task recovers each patch in pixel space, which lacks semantic-rich representations. They are not consistent with the cloze prompt.

Seeing the high similarity between BEIT and other MIM models, it is worthy applying our method on other MIM methods to strengthen our analysis. Therefore, we perform further study on applying the proposed VPTM to more MIM methods such as MAE [58], SimMIM [61] and EVA [63].

Because the `[MASK]` token in MAE [58][3] is utilized in the decoding stage, while only the weight of the encoder in MAE is released publicly. Thus, we conduct supplement experiments on SimMIM [61] and EVA [63]. The selected SimMIM [61][4] model is the ViT-Base pre-trained on ImageNet-1K for 800 epochs with the resolution as 224 × 224. The training target is raw pixel value regression, which focuses on learning low-level features. The architecture and pre-training data are both the same as the BEITv2 model adopted in the manuscript, and they both contain 86M parameters. The selected EVA [63][5] model has been pre-trained on 30M image data for 150 epochs. It contains 1.0B parameters, which is much larger than the SimMIM and BEITv2. EVA is pre-trained by reconstructing the masked out CLIP vision features conditioned on visible image patches for learning high-level semantics.

Relaxing the strict restriction on whether adopts visual tokenization in the MIM models, we perform prompt tuning for 50 epochs following the same setting in the Table 1. Results are given in Table 8. Because the pre-training process of SimMIM targets at raw pixel value regression, thus the `[MASK]` token tend to learn low-level features rather than high-level semantics. In consistent with the results of applying the VPTM on BEITv1 in Table 7, the performance is poor. As for EVA, the performance is great even without any hyperparameter optimization. In EVA, the authors hold that an ideal vision pretext task needs the abstraction of not only the low-level geometry and structure information, but also high-level semantics. Therefore, the `[MASK]` token in EVA model is equipped with high semantics through CLIP guided pre-training. However, EVA does not conduct tokenization, thus there still exists a gap between pre-training and masked token prediction in the proposed VPTM.

According to the experiments, on the one hand, the significance of semantic learning in MIM models is further strengthened. On the other hand, even with little gap caused by the lack of visual tokenization in pre-training, *the proposed VPTM also works well on MIM pre-training models targeting at learning high-level semantics*.

7. Conclusions

In this paper, we propose the Visual Prompt learning as masked visual Token Modeling (VPTM), which is the first visual prompt method designed on generative pre-trained visual models and achieves consistency between pre-training and visual classification by task reformulation. Extensive experiments show that VPTM outperforms linear probe and CLIP-based visual prompt baselines. Compared with VPT, we also achieve the best average accuracy. The proposed VPTM is revealed to be parameter-efficient and easy to be deployed uniformly. Further ablation studies validate the effectiveness of the prototypical verbalizer, and exhibit the robustness of our method against the positions of prompts and `[MASK]` token, prompt length and prototype dimensions. It demonstrates the rationality and efficacy of reformulating downstream tasks as the pre-training one to fulfill prompt learning in vision with task consistency. Moreover, it is validated that the proposed VPTM could also work well on other MIM pre-training models aiming at learning high-level semantics, even without specific visual tokenization.

CRediT authorship contribution statement

Ning Liao: Writing – original draft, Methodology, Conceptualization, Writing – review & editing, Visualization, Data curation. **Bowen Shi:** Validation. **Xiaopeng Zhang:** Project administration. **Min Cao:** Writing – review & editing. **Junchi Yan:** Funding acquisition, Formal analysis. **Qi Tian:** Supervision, Project administration.

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.

[3] mae_pretrain_vit_base.pth.

[4] simmim_pretrain_vit_base_img224_800ep.pth.

[5] eva_psz14.pt.

Acknowledgements

This work was partly supported at the university side, by National Science Foundation of China (62222607, 62002252), and Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102).

Data availability

There is no new dataset included in this paper. The datasets in our experiments are all publicly available, they are: CIFAR100 [66], CIFAR10 [66], Oxford Flowers102 [67], Food101 [68], EuroSAT [69], SVHN [70], Oxford Pets [71], DTD [72], Resisc45 [73], Patch Camelyon [74].

References

[1] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, Ross Girshick, Momentum contrast for unsupervised visual representation learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 9729–9738.

[2] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, Armand Joulin, Emerging properties in self-supervised vision transformers, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 9650–9660.

[3] Xinlei Chen, Saining Xie, Kaiming He, An empirical study of training self-supervised vision transformers, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 9640–9649.

[4] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al., Bootstrap your own latent-a new approach to self-supervised learning, Adv. Neural Inf. Process. Syst. 33 (2020) 21271–21284.

[5] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al., Improving Language Understanding by Generative Pre-Training, 2018.

[6] Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, Hsiao-Wuen Hon, Unified language model pre-training for natural language understanding and generation, Adv. Neural Inf. Process. Syst. 32 (2019).

[7] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R. Salakhutdinov, Quoc V. Le, Xlnet: generalized autoregressive pretraining for language understanding, Adv. Neural Inf. Process. Syst. 32 (2019).

[8] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, Luke Zettlemoyer, Bart: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 7871–7880.

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

[10] Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander Miller, Language models as knowledge bases?, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, 2019.

[11] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, J. Mach. Learn. Res. 21 (140) (2020) 1–67.

[12] Tianyu Gao, Adam Fisch, Danqi Chen, Making pre-trained language models better few-shot learners, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 2021, pp. 3816–3830.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, Bert: pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), 2019, pp. 4171–4186.

[14] Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, Qun Liu, Ernie: enhanced language representation with informative entities, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, 2019, p. 1441.

[15] Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, Richard Socher, Ask me anything: dynamic memory networks for natural language processing, in: International Conference on Machine Learning, PMLR, 2016, pp. 1378–1387.

[16] Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, Richard Socher, The natural language decathlon: multitask learning as question answering, arXiv preprint, arXiv:1806.08730, 2018.

[17] Shengding Hu, Ning Ding, Huadong Wang, Zhiyuan Liu, Jingang Wang, Juanzi Li, Wei Wu, Maosong Sun, Knowledgeable prompt-tuning: incorporating knowledge into prompt verbalizer for text classification, in: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, 2022.

[18] Timo Schick, Helmut Schmid, Hinrich Schütze, Automatically identifying words that can serve as labels for few-shot text classification, in: Proceedings of the 28th International Conference on Computational Linguistics, 2020, pp. 5569–5578.

[19] Ari Holtzman, Peter West, Vered Shwartz, Yejin Choi, Luke Zettlemoyer, Surface form competition: why the highest probability answer isn't always right, in: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, 2021, pp. 7038–7051.

[20] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias 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.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[22] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al., Learning transferable visual models from natural language supervision, in: International Conference on Machine Learning, PmLR, 2021, pp. 8748–8763.

[23] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, Ser-Nam Lim, Visual prompt tuning, in: European Conference on Computer Vision, Springer, 2022, pp. 709–727.

[24] Hyojin Bahng, Ali Jahanian, Swami Sankaranarayanan, Phillip Isola, Visual prompting: modifying pixel space to adapt pre-trained models, arXiv preprint, arXiv:2203.17274, 3 (11–12) (2022) 3.

[25] Aochuan Chen, Yuguang Yao, Pin-Yu Chen, Yihua Zhang, Sijia Liu, Understanding and improving visual prompting: a label-mapping perspective, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 19133–19143.

[26] Junyang Wu, Xianhang Li, Chen Wei, Huiyu Wang, Alan Yuille, Yuyin Zhou, Cihang Xie, Unleashing the power of visual prompting at the pixel level, Trans. Mach. Learn. Res. (2024).

[27] Jochem Loedeman, Maarten C. Stol, Tengda Han, Yuki M. Asano, Prompt generation networks for input-space adaptation of frozen vision transformers, arXiv preprint, arXiv:2210.06466, 2022.

[28] Zhiliang Peng, Li Dong, Hangbo Bao, Qixiang Ye, Furu Wei, Beit v2: masked image modeling with vector-quantized visual tokenizers, arXiv preprint, arXiv:2208.06366, 2022.

[29] Xing Nie, Bolin Ni, Jianlong Chang, Gaofeng Meng, Chunlei Huo, Shiming Xiang, Qi Tian, Pro-tuning: unified prompt tuning for vision tasks, IEEE Trans. Circuits Syst. Video Technol. 34 (6) (2023) 4653–4667.

[30] Kihyuk Sohn, Huiwen Chang, José Lezama, Luisa Polania, Han Zhang, Yuan Hao, Irfan Essa, Lu Jiang, Visual prompt tuning for generative transfer learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 19840–19851.

[31] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, Li Fei-Fei, Imagenet: a large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, Ieee, 2009, pp. 248–255.

[32] Hangbo Bao, Li Dong, Songhao Piao, Furu Wei, BEiT: BERT pre-training of image transformers, in: International Conference on Learning Representations, 2022.

[33] Yinyi Wei, Tong Mo, Yongtao Jiang, Weiping Li, Wen Zhao, Eliciting knowledge from pretrained language models for prototypical prompt verbalizer, in: International Conference on Artificial Neural Networks, Springer, 2022, pp. 222–233.

[34] Ganqu Cui, Shengding Hu, Ning Ding, Longtao Huang, Zhiyuan Liu, Prototypical verbalizer for prompt-based few-shot tuning, in: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2022, pp. 7014–7024.

[35] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al., Language models are few-shot learners, Adv. Neural Inf. Process. Syst. 33 (2020) 1877–1901.

[36] Yongliang Wu, Xu Yang, A glance at in-context learning, Front. Comput. Sci. 18 (5) (2024) 185347.

[37] Brian Lester, Rami Al-Rfou, Noah Constant, The power of scale for parameter-efficient prompt tuning, in: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2021.

[38] Timo Schick, Hinrich Schütze, Exploiting cloze-questions for few-shot text classification and natural language inference, in: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, Association for Computational Linguistics, 2021.

[39] Leyang Cui, Yu Wu, Jian Liu, Sen Yang, Yue Zhang, Template-based named entity recognition using bart, in: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, 2021, pp. 1835–1845.

[40] Allyson Ettinger, What bert is not: lessons from a new suite of psycholinguistic diagnostics for language models, Trans. Assoc. Comput. Linguist. 8 (2020) 34–48.

[41] Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, Hannaneh Hajishirzi, Unifiedqa: crossing format boundaries with a single qa system, in: Findings of the Association for Computational Linguistics: EMNLP 2020, 2020, pp. 1896–1907.

[42] Zhengbao Jiang, Frank F. Xu, Jun Araki, Graham Neubig, How can we know what language models know?, Trans. Assoc. Comput. Linguist. 8 (2020) 423–438.

[43] Timo Schick, Hinrich Schütze, Few-shot text generation with pattern-exploiting training, arXiv preprint, arXiv:2012.11926, 2020.

[44] Xiang Lisa Li, Percy Liang, Prefix-tuning: optimizing continuous prompts for generation, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Association for Computational Linguistics, 2021.

[45] Weizhe Yuan, Graham Neubig, Pengfei Liu, Bartscore: evaluating generated text as text generation, Adv. Neural Inf. Process. Syst. 34 (2021) 27263–27277.

[46] Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, Jie Tang, Gpt understands, too, AI Open 5 (2024) 208–215.

[47] Karen Hambardzumyan, Hrant Khachatrian, Jonathan May, Warp: word-level adversarial reprogramming, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 2021, pp. 4921–4933.

[48] Ningyu Zhang, Luoqiu Li, Xiang Chen, Shumin Deng, Zhen Bi, Chuanqi Tan, Fei Huang, Huajun Chen, Differentiable prompt makes pre-trained language models better few-shot learners, in: International Conference on Learning Representations, 2021.

[49] Aodi Li, Liansheng Zhuang, Shuo Fan, Shafei Wang, Learning common and specific visual prompts for domain generalization, in: Proceedings of the Asian Conference on Computer Vision, 2022, pp. 4260–4275.

[50] Yunhe Gao, Xingjian Shi, Yi Zhu, Hao Wang, Zhiqiang Tang, Xiong Zhou, Mu Li, Dimitris N. Metaxas, Visual prompt tuning for test-time domain adaptation, arXiv preprint, arXiv:2210.04831, 2022.

[51] Zangwei Zheng, Xiangyu Yue, Kai Wang, Yang You, Prompt vision transformer for domain generalization, arXiv preprint, arXiv:2208.08914, 2022.

[52] Ziqing Yang, Zeyang Sha, Michael Backes, Yang Zhang, From visual prompt learning to zero-shot transfer: mapping is all you need, arXiv preprint, arXiv:2303.05266, 2023.

[53] Lingbo Liu, Jianlong Chang, Bruce X.B. Yu, Liang Lin, Qi Tian, Chang-Wen Chen, Prompt-matched semantic segmentation, arXiv preprint, arXiv:2208.10159, 2022.

[54] Qidong Huang, Xiaoyi Dong, Dongdong Chen, Weiming Zhang, Feifei Wang, Gang Hua, Nenghai Yu, Diversity-aware meta visual prompting, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 10878–10887.

[55] Yuanhan Zhang, Kaiyang Zhou, Ziwei Liu, Neural prompt search, IEEE Trans. Pattern Anal. Mach. Intell. 47 (7) (2025) 5268–5280.

[56] Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, et al., Dualprompt: complementary prompting for rehearsal-free continual learning, in: European Conference on Computer Vision, Springer, 2022, pp. 631–648.

[57] Bowen Dong, Pan Zhou, Wangmeng Zuo, et al., Lpt: long-tailed prompt tuning for image classification, in: The Eleventh International Conference on Learning Representations, 2022.

[58] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, Ross Girshick, Masked autoencoders are scalable vision learners, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 16000–16009.

[59] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, Tao Kong, Image bert pre-training with online tokenizer, in: International Conference on Learning Representations, 2021.

[60] Xiaokang Chen, Mingyu Ding, Xiaodi Wang, Ying Xin, Shentong Mo, Yunhao Wang, Shumin Han, Ping Luo, Gang Zeng, Jingdong Wang, Context autoencoder for self-supervised representation learning, Int. J. Comput. Vis. 132 (1) (2024) 208–223.

[61] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, Han Hu, Simmim: a simple framework for masked image modeling, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 9653–9663.

[62] Jiahao Xie, Wei Li, Xiaohang Zhan, Ziwei Liu, Yew-Soon Ong, Chen Change Loy, Masked frequency modeling for self-supervised visual pre-training, in: The Eleventh International Conference on Learning Representations, 2023.

[63] Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, Yue Cao, Eva: exploring the limits of masked visual representation learning at scale, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 19358–19369.

[64] Yang Liu, Xinlong Wang, Muzhi Zhu, Yue Cao, Tiejun Huang, Chunhua Shen, Masked channel modeling for bootstrapping visual pre-training, Int. J. Comput. Vis. 133 (2) (2025) 760–780.

[65] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, Ilya Sutskever, Zero-shot text-to-image generation, in: International Conference on Machine Learning, Pmlr, 2021, pp. 8821–8831.

[66] A. Krizhevsky, Learning multiple layers of features from tiny images, Master's thesis, University of Tront, 2009.

[67] Maria-Elena Nilsback, Andrew Zisserman, Automated flower classification over a large number of classes, in: 2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing, IEEE, 2008, pp. 722–729.

[68] Lukas Bossard, Matthieu Guillaumin, Luc Van Gool, Food-101–mining discriminative components with random forests, in: European Conference on Computer Vision, Springer, 2014, pp. 446–461.

[69] Patrick Helber, Benjamin Bischke, Andreas Dengel, Damian 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.

[70] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Baolin Wu, Andrew Y. Ng, et al., Reading digits in natural images with unsupervised feature learning, in: NIPS Workshop on Deep Learning and Unsupervised Feature Learning, Granada, vol. 2011, 2011, p. 7.

[71] Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, C.V. Jawahar, Cats and dogs, in: 2012 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2012, pp. 3498–3505.

[72] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, Andrea Vedaldi, Describing textures in the wild, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 3606–3613.

[73] Gong Cheng, Junwei Han, Xiaoqiang Lu, Remote sensing image scene classification: benchmark and state of the art, Proc. IEEE 105 (10) (2017) 1865–1883.

[74] Bastiaan S. Veeling, Jasper Linmans, Jim Winkens, Taco Cohen, Max Welling, Rotation equivariant cnns for digital pathology, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2018, pp. 210–218.

[75] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, Ziwei Liu, Learning to prompt for vision-language models, Int. J. Comput. Vis. 130 (9) (2022) 2337–2348.

[76] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, Ziwei Liu, Conditional prompt learning for vision-language models, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 16816–16825.

[77] Ning Liao, Xiaopeng Zhang, Min Cao, Junchi Yan, M-tuning: prompt tuning with mitigated label bias in open-set scenarios, IEEE Trans. Circuits Syst. Video Technol. 35 (6) (2025) 5885–5899.

[78] Yuhang Zang, Wei Li, Kaiyang Zhou, Chen Huang, Chen Change Loy, Unified vision and language prompt learning, arXiv preprint, arXiv:2210.07225, 2022.

[79] Muhammad Uzair Khattak, Hanoona Rasheed, Muhammad Maaz, Salman Khan, Fahad Shahbaz Khan, Maple: multi-modal prompt learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 19113–19122.