Rethinking Visual Prompt Learning as Masked Visual Token Modeling

Ning Liao[1], Bowen Shi[1], Min Cao[2], Xiaopeng Zhang[3], Qi Tian[3], Junchi Yan[1]
[1]Shanghai Jiao Tong University [2]Soochow University [3]Huawei Cloud
{liaoning, sjtu_shibowen, yanjunchi}@sjtu.edu.cn mcao@suda.edu.cn
{zhangxiaopeng12, tian.qi1}@huawei.com

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, thus narrowing down the gap between them and improving the performance stably. However, when transferring it to the vision area, current visual prompt learning methods are all 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 into the pre-trained masked visual token prediction. 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, and the first to achieve 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 [21, 5, 8, 18]. 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 PMs.

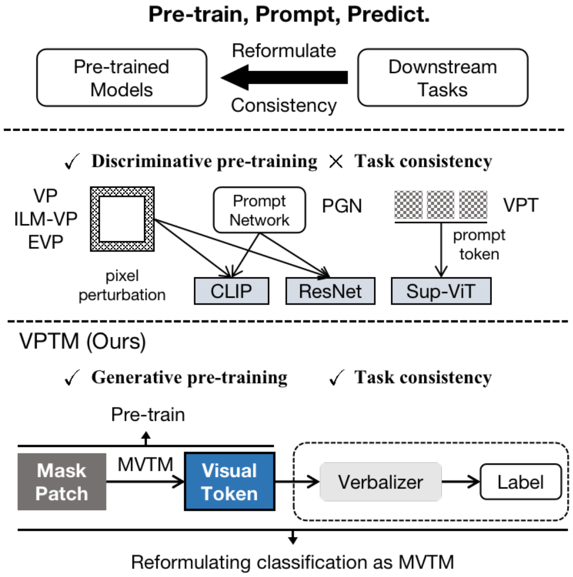


Figure 1. **Top.** Prompt learning in NLP reformulates downstream tasks as *generative* pre-training tasks. **Mid.** Visual prompt methods designed on discriminative pre-trained models concentrate on adding prompts to input space (VPT [26], VP [1], ILM-VP [6], EVP [57]) or learning prompt network (PGN [38]), while ignoring task consistency. **Bottom.** Our method aims at reformulating the downstream visual classification task as the generative masked visual token modeling (MVTM) pre-training task in BEITv2 [44].

To mitigate the issue in the paradigm, namely "pre-train, then fine-tune" [47, 14, 58, 34] in NLP, a new paradigm, namely "pre-train, prompt, then predict" [36] has been proposed [48, 45, 49, 17]. Based on the generative pre-trained models, e.g., GPT-3 [4], *the core technology is to reformulate downstream tasks to be the same form as the pre-training language modeling tasks*, as shown on the top of Fig. 1. In this way, when PMs are applied to downstream tasks, the knowledge of PMs could be naturally exploited with same objectives as in pre-training tasks, and

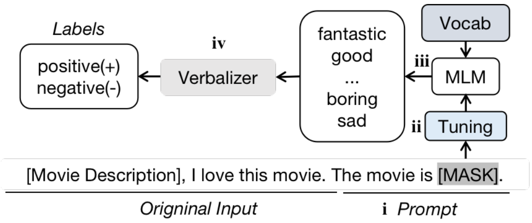


Figure 2. The prompt tuning paradigm in NLP. The text classification task is reformulated by the "cloze prompt" to keep the consistency with MLM pre-training.

contributes to better performance stably.

Taking the masked language modeling (MLM) pre-training task as an example [28, 62], classification tasks in NLP are usually reformulated by "cloze prompt", which follows four steps [32, 39] as shown in Fig. 2: (i) adding prompts with masks to the original input; (ii) performing prompt tuning; (iii) predicting the word in the masked place from the vocabulary by MLM; (iv) mapping the predicted words to downstream labels using verbalizer. Specifically, the predicted words are usually not the downstream labels, thus verbalizer [25, 51, 24, 17] is devised to establish connections between them, e.g., the verbalizer in Fig. 2 maps "fantastic, good" to "positive". As such, prompt learning can be well applied in solving tasks such as text classification, named entity recognition, and so on.

Witnessing the success of prompt learning in NLP, researchers introduce prompt learning into vision applications [41, 38, 54]. VPT [26] prepends a few parameters as prompts to the input sequence of ViT [15], which has been supervised pre-trained on ImageNet-21k [13] in a discriminative way. Visual prompting (VP) [1] modifies the pixel space with learnable parameters to perform visual prompt learning on CLIP [46], which has been pre-trained by contrastive learning. Till now, the current visual prompting methods [38, 54, 57, 41] are all designed on discriminative pre-trained models shown in the middle of Fig. 1. *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 [26, 1, 57, 6], learning prompt networks [38] or designing prompt blocks [41], *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 generative visual prompt learning by inheriting the generative pre-training task to achieve consistency.

For this, based on the generative model BEITv2 [44], in which the visual tokens of masked patches are predicted from the codebook in pre-training, we propose Visual Prompt learning as masked visual Token Modeling (VPTM) for the visual classification task, as shown in the bottom of Fig. 1. Specifically, we concatenate continuous prompts and pre-trained mask token to the input sequence in prompt tuning. *The 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 [56, 11].

Experimentally, VPTM outperforms other visual prompt learning methods [26, 1, 6, 38, 57] 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. To the best of our knowledge, it is the first visual prompt learning method on generative pre-trained models and achieves a close match between the forms of downstream and pre-training tasks.

2) For mapping from 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.

2. Related Work

2.1. Prompt Learning in NLP

Reformulating downstream tasks as pre-training task. Considering a limited application of fine-tuning the entire large-scale pre-trained model [47, 48, 4, 28] for downstream tasks, and the impaired performance due to the gap between pre-trained and fine-tuning tasks, researchers in NLP initially proposed prompt learning [48, 45, 49, 17], 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 [48, 45, 49, 17]. Based on the generative masked language modeling pre-training task, the text classification [33, 53, 51], named entity recognition [12] and commonsense reasoning [16] are transformed into "cloze prompt", in which prompts with the mask token are added to the original input. The tokens predicted in masked places by the pre-training task are then mapped to the answers by the verbalizer. The question answering [29, 27], text generation [4, 52, 35] and automatic evaluation of text generation [59] are reformulated as "prefix prompt", in which a prefix string is prepended to the original input for answer text generation. *Our method is inspired by the core idea in prompt learning, i.e., reformulating the downstream tasks as the pre-training task.*

Verbalizer. In the "cloze prompt", the predicted words in masked places are usually not the labels. To map the predicted words to labels, handcrafted verbalizer [51, 53] was initially proposed by manually designed rules. To avoid the expert dependence and prediction bias in handcrafted verbalizer, the method [37] uses gradient descent to search the mapping. KPT [25] incorporates external knowledge bases into verbalizer for text classification. Soft verbalizer [19, 61] regards each label as a trainable token, which is optimized together with prompt tuning. Prototypical verbalizer [56, 11] 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 relationships between visual tokens with implicit meaning and downstream labels.

2.2. Unimodal Visual Prompt Learning

As an effective and efficient alternative technology of fine-tuning, prompt learning has been introduced into unimodal vision area [38, 54]. VPT [26] 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. Visual prompting (VP) [1] introduces learnable pixel perturbation on the image encoder of CLIP [46] as prompts to be optimized. Pro-tuning [41] adapts pre-trained ResNets [22] to downstream tasks by introducing lightweight prompt blocks. However, these methods are all designed on the discriminative pre-trained models. Prompting on generative pre-trained models has not been studied. Keeping consistency between pre-training and downstream applications also 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 NLP [28, 62]. With masking pieces of the inputted 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 [20, 63, 7]. BEITv1 [2] was proposed to predict the visual tokens, which are tokenized by the codebook of DALL-E [50], of the masked patches. However, BEITv1 is limited to learn the low-level features in the codebook of DALL-E. To solve the drawback of semantic-less representations of BEITv1 [2], BEITv2 [44] was then proposed with a semantic-rich tokenizer guided by CLIP [46] or DINO [5]. The pre-training strategies of BEITv1 and BEITv2 are the same. As the visual tokens are supposed to be equipped with high-level semantics as the language words, we propose the visual prompt learning in consistency with masked visual token modeling on BEITv2.

3. 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 preliminary, i.e., the MVTM pre-training task of BEIT-series models [2, 44] in Sec.3.1. The proposed method is presented in Sec. 3.2. The prototypical verbalizer for mapping the predicted visual token to downstream labels is devised in Sec. 3.3.

3.1. Masked Visual Token Modeling Pre-training

Following BERT [28], BEITv1 [2] and BEITv2 [44] are pre-trained by generative masked visual token modeling. 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:

$$\mathcal{L} = -\sum_{\boldsymbol{x}\in\mathcal{D}}\sum_{i\in\mathcal{M}}\log p(z_i|\boldsymbol{x}^{\mathcal{M}}). \tag{1}$$

3.2. Visual Prompt Learning as MVTM

Based on the masked visual token modeling (MVTM) pre-training, we propose the visual prompt learning method VPTM, as shown in Fig. 3 (b). Our method reformulates the visual classification task as MVTM task. 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 [44] through an embedding function $\texttt{Embed}_{pre}$:

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

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. To preserve the image information and reformulate the classification task as

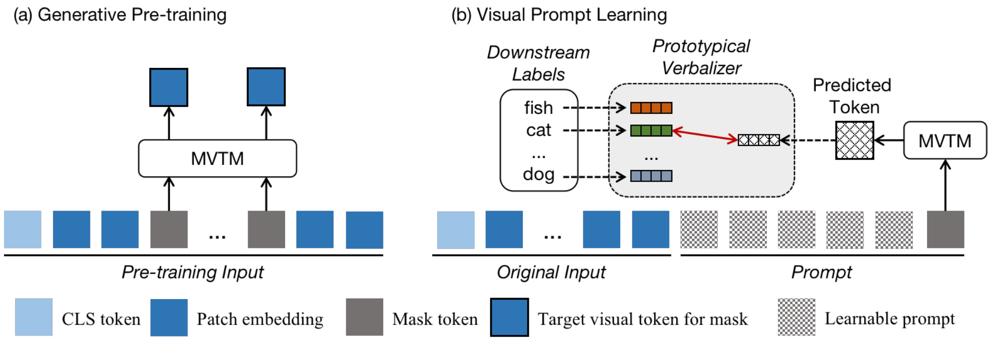


Figure 3. (a) BEIT-series models [2, 44] are pre-trained by predicting the visual tokens of the masked patches. (b) The proposed VPTM reformulates the visual classification task as the generative masked visual token modeling (MVTM) task with pre-trained BEITv2. The proposed prototypical verbalizer constructs 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 is ablating studied in experiments.

the MVTM task, we concatenate the `[MASK]` token $\boldsymbol{e}_{\texttt{[MASK]}}$, which is also from the pre-trained model, to the original input sequence. For continuously tuning towards downstream datasets, we insert additional N_p learnable prompts $\boldsymbol{p}_i, i \in [1, N_p]$ into the sequence. Then, we obtain the final input sequence $\boldsymbol{H}_{vp}$:

$$\boldsymbol{H}_{vp} = [\boldsymbol{e}_{\texttt{[CLS]}}, \boldsymbol{e}_1, ..., \boldsymbol{e}_N, \boldsymbol{p}_1, ..., \boldsymbol{p}_{N_p}, \boldsymbol{e}_{\texttt{[MASK]}}]. \tag{3}$$

The positions of the prompts $\boldsymbol{p}_i$ and `[MASK]` token $\boldsymbol{e}_{\texttt{[MASK]}}$ is ablating studied 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 to be predicted. 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. 3.3.

3.3. 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 [56, 11].

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 $\boldsymbol{h}_{\texttt{[MASK]}}$, we project it into the prototype space using linear function $\mathcal{F} : \mathbb{R}^d \rightarrow \mathbb{R}^t$, then get vector $\boldsymbol{u}_{\texttt{[MASK]}}$:

$$\boldsymbol{u}_{\texttt{[MASK]}} = \mathcal{F}(\boldsymbol{h}_{\texttt{[MASK]}}). \tag{4}$$

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:

$$\texttt{sim}(\boldsymbol{x}_i, C_i) = \boldsymbol{u}^{\boldsymbol{i}}_{\texttt{[MASK]}} \cdot \boldsymbol{c}_i^T, \tag{5}$$

where T is the transpose manipulation.

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. For a batch of N images, the loss is:

$$\begin{aligned} \mathcal{L}_{vp} &= -\frac{1}{N}\sum_{i=1}^{N} \log \frac{\exp(\texttt{sim}(\boldsymbol{x}_i, C_i))}{\sum_{k=1}^{N_C} \exp(\texttt{sim}(\boldsymbol{x}_i, C_k))} \\ &= -\frac{1}{N}\sum_{i=1}^{N} \log \frac{\exp(\boldsymbol{u}^{\boldsymbol{i}}_{\texttt{[MASK]}} \cdot \boldsymbol{c}_i^T)}{\sum_{k=1}^{N_C} \exp(\boldsymbol{u}^{\boldsymbol{i}}_{\texttt{[MASK]}} \cdot \boldsymbol{c}_k^T)}. \end{aligned} \tag{6}$$

In the prompt tuning phase, all parameters in the pre-trained vision model are kept frozen.

4. Experiments

4.1. Datasets

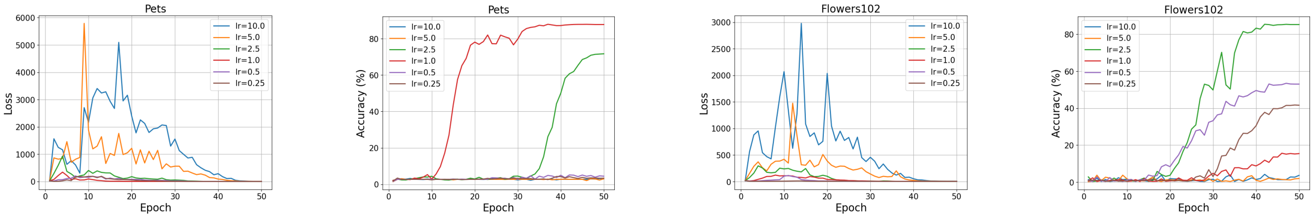
To evaluate the performance of the proposed VPTM, we select 10 datasets in our experiments. The visual prompts are learned on the training sets and evaluated on the test sets. In the datasets, CIFAR100 [31] includes 100 classes, CIFAR10 [31] includes 10 classes, Oxford Flowers102 [42] includes 102 classes, Food101 [3] includes 101 classes, EuroSAT [23] includes 10 classes, SVHN [40] includes 10 classes, Oxford Pets [43] includes 37 classes, DTD [10] includes 47 classes, Resisc45 [9] includes 45 classes, Patch Camelyon (PatchCame) [55] includes 2 classes.

4.2. Baseline Methods

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

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 [1]	82.10	95.80	97.40	80.50	97.90	95.70	88.50	72.30	94.40	N.R.	89.40
LP+CLIP [1]	80.00	95.00	96.90	84.60	95.30	65.40	89.20	74.60	66.00	N.R.	83.00
TP+CLIP [1]	63.10	89.00	61.90	79.80	40.00	5.10	85.90	43.00	42.40	N.R.	56.69
TP+VP [1]+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 [38]+CLIP	79.30	96.10	**94.00**	82.50	98.00	94.20	**91.50**	71.50	92.10	N.R.	88.80
EVP [57]	**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 [6]	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 [26]+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**



(a) Loss curve on Pets. (b) Accuracy curve on Pets. (c) Loss curve on Flowers. (d) Accuracy curve on Flowers.

Figure 4. Loss and accuracy curves with different hyperparameters in VPT [26]. 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.

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

(2) *Linear probe (FP).* 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) [1], which adds perturbation on the pixel; e) TP + PGN [38], which generates prompts for input; f) EVP [57], which adds prompts on the pixel with improved generalization; g) ILM-VP [6], which adds prompts on the pixel and learns a label mapping.

(4) *Visual prompt tuning (VPT)* [26]. 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*, we could not compare with multi-modal prompt methods performed on texts, such as CoOp [65] and CoCoOp [64]. We also could not compare with multi-modal prompt methods that the visual and textual prompts are learned jointly and could not be separated, including UPT [60], MaPLe [30].

4.3. Implementation Details

We experiment with the BEITv2 [44], which has been pre-trained on ImageNet-1k [13] by masked visual token modeling and the codebook is guided by CLIP [46]. Main experiments of our method are performed for 50 epochs 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 all zeros. Fine-tuning and linear probing on BEITv2 [44] are performed for 50 epochs following the official code [1]. We implement VPT on the weights of BEITv2 for a fair comparison [2].

4.4. Comparison to Baseline Methods

The results measured by accuracy are shown in Table 1.

(1) Comparison with FT & LP on BEITv2. Our method is inferior to fine-tuning the entire model. We infer the reason as the size of parameters that are optimized. Our method only optimizes 0.17% parameters of the entire model on average, which is discussed in the next section. The fewer parameters are optimized, the lower the performance upper

[1] https://github.com/microsoft/unilm/tree/master/beit2.

[2] VPT is officially performed on supervised pre-trained ViT [15].

Table 2. The comparisons between the proposed VPTM and VPT [26] on prototype dimension t, prompt length N_p, GFLOPs, and the ratio of the amount of tuned parameters to the entire parameters in the optimal setting.

	Index	CIFAR100	CIFAR10	Flowers	Food101	EuroSAT	SVHN	Pets	DTD	Resisc45	PatchCame	Average
VPT [26]	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**
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

Table 3. 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 (Ours)	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**

bound is. On the other hand, our method consistently surpasses linear probe on all datasets more than about 5% on average. Particularly, our method outperforms 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 in linear probe.

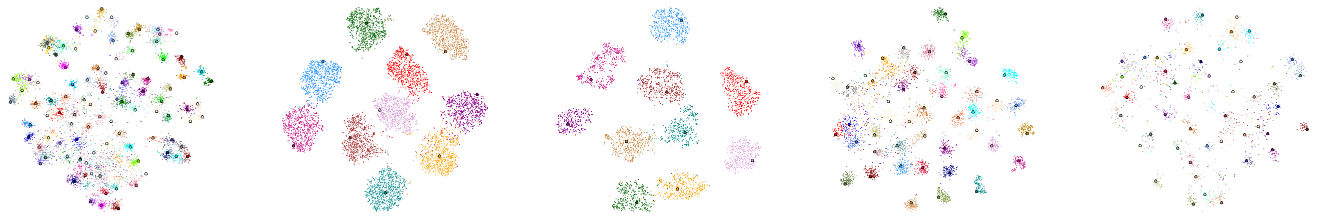
(2) Comparison with prompting methods on the image encoder of CLIP. Our method achieves competitive performance compared with fine-tuning the entire CLIP [46]. Concerning the results of VP [1] and PGN [38] combined with textual prompts, our method achieves the best average performance even without the assistance of texts. In addition, compared with EVP [57] and ILM-VP [6], 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.

(3) Comparison with VPT [26]. For a fair comparison, we implement VPT-shallow on BEITv2 for 100 epochs following the official setting of VPT [26]. 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 comparable average performance to that of VPT under 100 epochs. The accuracy of VPTM are 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 described in the paper of VPT [26], different datasets adopt different parameters of the learning rate and the weight decay. Given that, when implementing 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.**

4.5. 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, and the ratio of the amount of tuned parameters to the entire parameters $Tuned/Total$. Besides, the prototype vectors are also counted as optimized parameters in VPTM, we show



(a) CIFAR100. (b) CIFAR10. (c) EuroSAT. (d) Resisc45. (e) DTD.

Figure 5. The 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.

Table 4. 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

the prototype dimension t together in Table 2.

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. Therefore, by inheriting the pre-training task to keep consistency, VPTM is proved to be more efficient.

4.6. Ablation Studies

4.6.1 Effectiveness 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. 3.

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 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.

4.6.2 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. The position relationships are represented by the order of their abbreviations. Based on the optimal setting in Table 1, the ablation results are shown in Table 4.

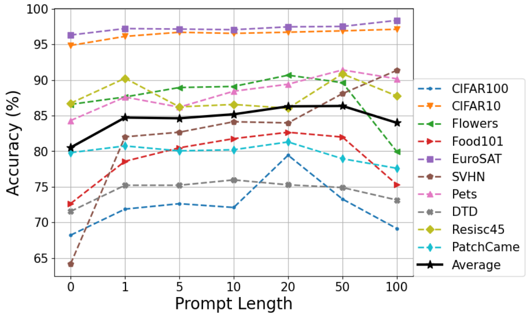


Figure 6. Accuracy under different settings of prompt length.

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.

4.6.3 Prompt Length N_p

We compare the results under different prompt length $N_p \in [0, 1, 5, 10, 20, 50, 100]$. The results are shown in Fig. 6.

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. 6, 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 long prompts, i.e., more parameters that can be optimized.

4.6.4 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. 7. 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.

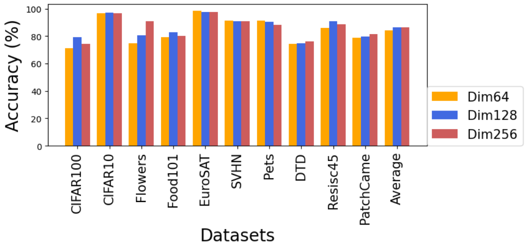


Figure 7. Accuracy under different prototype dimensions.

5. 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. Specifically, *the large language models in NLP [28, 62] almost take the language modeling as pre-training task, thus the cloze prompt could be applied on most of them for keeping the task consistency*. In comparison, the pre-training tasks in vision area are various, e.g., supervised pre-training [15], masked image modeling (MIM) [20, 7], and masked visual token modeling (MVTM) [2, 44]. 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. Thus, VPTM with a specific cloze prompt could not be applied on all pre-trained vision models.

On the other hand, the codebook also plays a key role. *To achieve classification by mapping the visual tokens to downstream labels, visual tokens are supposed to be equipped with high-semantics, as the words in vocabulary in NLP*. Taking a step further, given the significant similarity between MLM and MVTM pre-training, also between cloze prompt in NLP and VPTM, it is expected to achieve unified multimodal prompting by inheriting the generative mask modeling pre-training in the vision-language area.

6. Conclusions

In this paper, we proposed 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.

References

[1] Hyojin Bahng, Ali Jahanian, Swami Sankaranarayanan, and Phillip Isola. Visual prompting: Modifying pixel space to adapt pre-trained models. *arXiv preprint arXiv:2203.17274*, 2022. 1, 2, 3, 5, 6

[2] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. In *ICLR*, 2021. 3, 4, 8

[3] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *ECCV*, pages 446–461. Springer, 2014. 4

[4] 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. *NIPS*, 33:1877–1901, 2020. 1, 2

[5] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, pages 9650–9660, 2021. 1, 3

[6] Aochuan Chen, Yuguang Yao, Pin-Yu Chen, Yihua Zhang, and Sijia Liu. Understanding and improving visual prompting: A label-mapping perspective. *arXiv preprint arXiv:2211.11635*, 2022. 1, 2, 5, 6

[7] Xiaokang Chen, Mingyu Ding, Xiaodi Wang, Ying Xin, Shentong Mo, Yunhao Wang, Shumin Han, Ping Luo, Gang Zeng, and Jingdong Wang. Context autoencoder for self-supervised representation learning. *arXiv preprint arXiv:2202.03026*, 2022. 3, 8

[8] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *ICCV*, pages 9640–9649, 2021. 1

[9] Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883, 2017. 4

[10] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *CVPR*, pages 3606–3613, 2014. 4

[11] Ganqu Cui, Shengding Hu, Ning Ding, Longtao Huang, and Zhiyuan Liu. Prototypical verbalizer for prompt-based few-shot tuning. In *ACL*, pages 7014–7024, 2022. 2, 3, 4

[12] Leyang Cui, Yu Wu, Jian Liu, Sen Yang, and Yue Zhang. Template-based named entity recognition using bart. In *ACL/IJCNLP (Findings)*, 2021. 2

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. Ieee, 2009. 2, 5

[14] 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. *NIPS*, 32, 2019. 1

[15] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2020. 2, 5, 8

[16] Allyson Ettinger. What bert is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *TACL*, 8:34–48, 2020. 2

[17] Tianyu Gao, Adam Fisch, and Danqi Chen. Making pre-trained language models better few-shot learners. In *IJCNLP*, pages 3816–3830, 2021. 1, 2

[18] 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. *NIPS*, 33:21271–21284, 2020. 1

[19] Karen Hambardzumyan, Hrant Khachatrian, and Jonathan May. Warp: Word-level adversarial reprogramming. In *IJCNLP*, pages 4921–4933, 2021. 3

[20] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, pages 16000–16009, 2022. 3, 8

[21] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, pages 9729–9738, 2020. 1

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 3

[23] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019. 4

[24] Ari Holtzman, Peter West, Vered Shwartz, Yejin Choi, and Luke Zettlemoyer. Surface form competition: Why the highest probability answer isn't always right. In *EMNLP*, pages 7038–7051, 2021. 2

[25] Shengding Hu, Ning Ding, Huadong Wang, Zhiyuan Liu, Jingang Wang, Juanzi Li, Wei Wu, and Maosong Sun. Knowledgeable prompt-tuning: Incorporating knowledge into prompt verbalizer for text classification. In *ACL*, pages 2225–2240, 2022. 2, 3

[26] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *ECCV*, 2022. 1, 2, 3, 5, 6

[27] Zhengbao Jiang, Frank F Xu, Jun Araki, and Graham Neubig. How can we know what language models know? *TACL*, 8:423–438, 2020. 2

[28] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186, 2019. 2, 3, 8

[29] Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. Unifiedqa: Crossing format boundaries with a single qa system. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1896–1907, 2020. 2

[30] Muhammad Uzair Khattak, Hanoona Rasheed, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. Maple: Multi-modal prompt learning. *arXiv preprint arXiv:2210.03117*, 2022. 5

[31] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 4

[32] Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. Ask me anything: Dynamic memory networks for natural language processing. In *ICML*, pages 1378–1387. PMLR, 2016. 2

[33] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *EMNLP*, pages 3045–3059, 2021. 2

[34] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *ACL*, pages 7871–7880, 2020. 1

[35] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *IJCNLP*, pages 4582–4597, 2021. 2

[36] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*, 2021. 1

[37] Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. Gpt understands, too. *arXiv preprint arXiv:2103.10385*, 2021. 3

[38] Jochem Loedeman, Maarten C Stol, Tengda Han, and Yuki M Asano. Prompt generation networks for efficient adaptation of frozen vision transformers. *arXiv preprint arXiv:2210.06466*, 2022. 1, 2, 3, 5, 6

[39] Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. The natural language decathlon: Multitask learning as question answering. *arXiv preprint arXiv:1806.08730*, 2018. 2

[40] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011. 4

[41] Xing Nie, Bolin Ni, Jianlong Chang, Gaomeng Meng, Chunlei Huo, Zhaoxiang Zhang, Shiming Xiang, Qi Tian, and Chunhong Pan. Pro-tuning: Unified prompt tuning for vision tasks. *arXiv preprint arXiv:2207.14381*, 2022. 2, 3

[42] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729. IEEE, 2008. 4

[43] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *CVPR*, pages 3498–3505. IEEE, 2012. 4

[44] Zhiliang Peng, Li Dong, Hangbo Bao, Qixiang Ye, and Furu Wei. Beit v2: Masked image modeling with vector-quantized visual tokenizers. *arXiv preprint arXiv:2208.06366*, 2022. 1, 2, 3, 4, 5, 8

[45] Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. Language models as knowledge bases? In *EMNLP-IJCNLP*, pages 2463–2473, 2019. 1, 2

[46] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763. PMLR, 2021. 2, 3, 5, 6

[47] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018. 1, 2

[48] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. 1, 2

[49] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*, 21(140):1–67, 2020. 1, 2

[50] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *ICML*, pages 8821–8831. PMLR, 2021. 3

[51] Timo Schick, Helmut Schmid, and Hinrich Schütze. Automatically identifying words that can serve as labels for few-shot text classification. In *COLING*, pages 5569–5578, 2020. 2, 3

[52] Timo Schick and Hinrich Schütze. Few-shot text generation with pattern-exploiting training. *arXiv preprint arXiv:2012.11926*, 2020. 2

[53] Timo Schick and Hinrich Schütze. Exploiting cloze-questions for few-shot text classification and natural language inference. In *ACL*, pages 255–269, 2021. 2, 3

[54] Kihyuk Sohn, Yuan Hao, José Lezama, Luisa Polania, Huiwen Chang, Han Zhang, Irfan Essa, and Lu Jiang. Visual prompt tuning for generative transfer learning. *arXiv preprint arXiv:2210.00990*, 2022. 2, 3

[55] Bastiaan S Veeling, Jasper Linmans, Jim Winkens, Taco Cohen, and Max Welling. Rotation equivariant cnns for digital pathology. In *International Conference on Medical image computing and computer-assisted intervention*, pages 210–218. Springer, 2018. 4

[56] Yinyi Wei, Tong Mo, Yongtao Jiang, Weiping Li, and Wen Zhao. Eliciting knowledge from pretrained language models for prototypical prompt verbalizer. *arXiv preprint arXiv:2201.05411*, 2022. 2, 3, 4

[57] Junyang Wu, Xianhang Li, Chen Wei, Huiyu Wang, Alan Yuille, Yuyin Zhou, and Cihang Xie. Unleashing the power of visual prompting at the pixel level. *arXiv preprint arXiv:2212.10556*, 2022. 1, 2, 5, 6

[58] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *NIPS*, 32, 2019. 1

[59] Weizhe Yuan, Graham Neubig, and Pengfei Liu. Bartscore: Evaluating generated text as text generation. *NIPS*, 34:27263–27277, 2021. 2

[60] Yuhang Zang, Wei Li, Kaiyang Zhou, Chen Huang, and Chen Change Loy. Unified vision and language prompt learning. *arXiv preprint arXiv:2210.07225*, 2022. 5

[61] Ningyu Zhang, Luoqiu Li, Xiang Chen, Shumin Deng, Zhen Bi, Chuanqi Tan, Fei Huang, and Huajun Chen. Differentiable prompt makes pre-trained language models better few-shot learners. In *ICLR*, 2021. 3

[62] Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. Ernie: Enhanced language representation with informative entities. In *ACL*, pages 1441–1451, 2019. 2, 3, 8

[63] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. Image bert pre-training with online tokenizer. In *ICLR*, 2021. 3

[64] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *CVPR*, pages 16816–16825, 2022. 5

[65] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *IJCV*, 130(9):2337–2348, 2022. 5