BRIDGING ATTACK AND PROMPTING: AN ENHANCED VISUAL PROMPTING AT THE PIXEL LEVEL

Anonymous authors
Paper under double-blind review

ABSTRACT

In this paper, we study the problem of the visual prompt at the pixel level. Recent works demonstrate flexibility and generalization of visual-only prompt. However, it still cannot achieve superior results compared with linear probe in terms of accuracy and parameter efficiency. We believe that the full power of visual prompt remains to be harnessed through a novel perspective, which bridges adversarial attack and visual prompt considering the high similarity in both formats and objective functions. Bringing in the "old ideas" in adversarial attacks to enhance visual prompt is promising since there are extensive theoretical and empirical solutions to improve the performance of adversarial attack. Therefore, we propose a novel and concise visual prompting method incorporating simple and effective training strategies inspired by ideas from adversarial attack. Specifically, we introduce the input diversity and gradient normalization into visual prompt learning to obtain better generalization ability. Moreover, to avoid disruptions to the original image caused by perturbation without changing the spatial size of inputs, we separate the prompt and image by shrinking and then padding the image with learnable visual prompts, which can significantly improve the performance further without increasing FLOPs. Extensive experiments are conducted on various large-scale pre-trained models across several downstream datasets under different scenarios. We show that with a CLIP-based model, our enhanced visual prompt can successfully outperform linear probe by **1.9%** across 12 datasets on average with a comparable number of parameters, and can even match fully fine-tuning paradigm in some settings by training with only **0.04%** parameters.

1 INTRODUCTION

Deep learning models have witnessed pre-training on increasingly large-scale data as a general and more effective path to success (He et al., 2022; Radford et al., 2021; Bao et al., 2021; Devlin et al., 2018). At the same time, the model's size is getting larger along with the scale of the data. These large foundation models can achieve state-of-the-art performance in both vision (He et al., 2022; Radford et al., 2021; Bao et al., 2021) and natural language processing (Devlin et al., 2018) domains for various downstream tasks. One de-facto standard tuning paradigm of these large-scale models is fully fine-tuning, which not only introduces extra parameters, e.g., linear layer, but also requires the whole access to the model's parameters and enormous space to store them. Hence, for these ever-growing models, researchers devote more efforts to designing parameter-efficient turning pipelines.

In NLP, *prompting* method is one of the effective and efficient strategies, which can only modify input space to adapt the model for the downstream task (Gao et al., 2021; Lester et al., 2021; Li & Liang, 2021). The text prompt can match the performance of fully fine-tuning (Liu et al., 2021b). In the visual field, whether or not prompt can replicate such success has attracted the attention of many researchers. Recently, authors in (Jia et al., 2022) added a small amount of learnable parameters as tokens into large vision transformers to adapt specific downstream tasks. Concurrently, inspired by adversarial reprogramming, researchers in (Bahng et al., 2022) found that adding learnable perturbation at the pixel level can be an alternative way to utilize large-scale pre-trained models in specific downstream tasks. Both works demonstrate the potential representation power of visual prompt. However, if we take a close look at the trade-off between performance and parameter efficiency in these works as shown in Fig 1, current state-of-art visual prompting VPT (Jia et al., 2022) cannot achieve superior performance in terms of accuracy (**77.2% v.s. 80.7%**) compared to linear probe

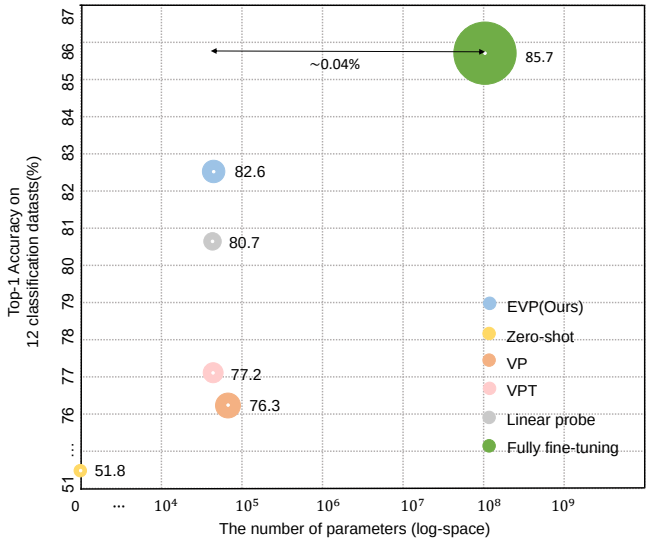


Figure 1: The trade-off between the number of parameters and accuracy. Our method outperforms linear probe and other prompting methods by a large margin with a similar amount of parameters.

paradigm. It is natural to ask *if visual prompt can be a preferable alternative of a simple linear layer in different scenarios empirically.*

With this curiosity, we delved into how to make universal visual prompts gain a stronger representation of learning capability without compromising efficiency. More concretely, the question is how visual prompt can improve the generalization ability across different datasets and models by only using the gradient information through backward propagation. To answer this, researchers in VP (Bahng et al., 2022) have built an early ***bridge*** between visual prompt and adversarial attack because of similarities in both format and objective function. Without accessing and storing the whole model, the VP achieves promising results with a small amount of parameters. However, it is still far behind the linear probe method as shown in Fig 1. We believe that the full power of visual prompt at the pixel level remains to be harnessed. The ***bridge*** can be strengthened through more advanced tools in adversarial attack, which has tons of well-established theoretical and empirical solutions to improve generalization and transferability of adversarial examples (Xie et al., 2019; Dong et al., 2017). Inspired by these works, we investigate how to integrate these on-the-fly adversarial attack tools into visual prompt deeply. Surprisingly, we find that gradient normalization (Goodfellow et al., 2015) and input diversity (Xie et al., 2019), which play an important role in adversarial examples, can also significantly improve the generalization ability of visual prompt at the pixel level. Furthermore, we notice that additive visual prompt on the original image may cover up the original image information. Thus, it becomes a new burden to improve classification performance further. To alleviate this problem, we separate the visual prompt and image by simply shrinking the original image into a smaller size and then padding it with learnable parameters back to the original size. Such separation not only preserves the image information, which can remarkably improve the performance, but also guarantees that the model FLOPs remain the same. Moreover, such a natural padding approach can make full use of positional embeddings, which we also find critical in prompting design.

Driven by these important findings, we design a novel and concise visual prompting method incorporating a simple and effective training strategy. To demonstrate the effectiveness, we conduct extensive experiments on three large-scale pre-trained models across 17 datasets. Specifically, on the CLIP-based model (Radford et al., 2021), our enhanced visual prompt can successfully outperform linear probe across 12 downstream tasks by $\mathbf{1.9}\%$ and beat the previous state-of-the-art visual prompt methods by $\mathbf{5.5}\%$ on average with similar or fewer parameters. Our method can also improve performance by a large margin on other pre-trained models that contains a specific linear classification layer. Attributed to the flexibility of visual prompting, we can further explore the potential of visual prompt in few-shot learning and out-of-distribution scenario. Surprisingly, our proposed visual prompt can achieve superior performance over linear probe with limited labeled data. On the out-of-distribution benchmark, our method even matches fully fine-tuning paradigm. We hope our enhanced visual prompt can inspire both vision and language prompt learning study in the future.

2 RELATED WORKS

Prompt Learning in NLP The key idea of prompting is to reformulate the input text in downstream tasks so that the frozen language models can *"understand"* the downstream task (Liu et al., 2021a). GPT-3 (Brown et al., 2020) demonstrates that manually operating text prompts can achieve remarkable representation capacity in few-shot or even zero-shot learning paradigm. Following this, recent works (Petroni et al., 2019; Cui et al., 2021) show that delicately hand-designed text prompt can further improve the generalization ability. Since designing the text prompts manually needs specific domain knowledge, more and more researchers devote efforts to prompt tuning (Li & Liang, 2021; Liu et al., 2021b; Lester et al., 2021), which directly optimizes the continuous prompt vector via gradient information. Compared with these works, we focus on visual prompt learning at the pixel level, which is a different type of signal from human language that contains high-level semantic information. Thus, visual prompt is more challenging than prompt learning in NLP.

Visual Prompt Learning After witnessing the success of prompt learning in language models, some prior works (Zhou et al., 2022; Bahng et al., 2022) aim at investigating the prompts on vision-language models like CLIP (Radford et al., 2021). For example, CoOp (Zhou et al., 2022) applies the prompt tuning to the vision-language models, which learns the soft prompt via minimizing the classification loss on the downstream tasks. Due to the different modalities between vision and language, there are few works (Bahng et al., 2022; Jia et al., 2022) to prompt with images. VP (Bahng et al., 2022) aims at optimizing the prompts in pixel space, which builds an early connection between attack and visual prompt learning. VPT (Jia et al., 2022) proposes visual prompts specific to ViT (Dosovitskiy et al., 2020) architecture. It adopts deep prompt tuning (Lester et al., 2021) by inserting a set of learnable tokens into each Transformer encoder layer. Although these works show the potential of visual-only prompt learning, we find their performance is still not promising compared with linear probe. We enhance the visual prompt by exploring the cooperation power of advanced adversarial attack tools and successfully outperform linear probe baseline by a notable margin on a wide range of datasets and tasks.

Adversarial Attack Previous works (Dalvi et al., 2004; Biggio et al., 2013; Huang et al., 2011) show that machine learning models are vulnerable to adversarial attacks. Researchers have devoted significant efforts to designing adversarial examples for several years. Goodfellow *et al.* (Goodfellow et al., 2015) proposed the fast gradient sign method to perturb a normal example for one step along the gradient direction. The methods were extended by Madry *et al.* to Projected Gradient Descent (Madry et al., 2018), which perturbs normal examples for several steps with a smaller step size. Adversarial reprogramming (Elsayed et al., 2018) tries to build class-agnostic and universal adversarial examples. Although different from adversarial goals, to our surprise, we find that pixel-level visual prompting is amenable to using the same optimization strategy to improve performance. There are also many works that focus on improving transferability in adversarial attacks. Xie *et al.* (Xie et al., 2019) proposed a method by creating diverse input patterns. Momentum-based iterative algorithms (Dong et al., 2017), proposed by Dong *et al.*, can build robust adversarial example. We also find that applying the input diversity, i.e., augmentation in visual prompt, can further improve performance.

3 METHODOLOGY

We propose a visual prompting method at the pixel level with a simple but effective training strategy inspired by the adversarial attack for adapting large pre-trained models to downstream tasks, especially for the pre-trained CLIP model. We first review previous visual prompting methods, including VP (Bahng et al., 2022) and VPT (Jia et al., 2022). Then, we present our approach with technical details, including prompt design and the training strategy.

3.1 PRELIMINARIES

VP (Bahng et al., 2022) aims to adapt the pre-trained models to downstream tasks by modifying some learnable pixels in original images. The key idea of VP is to learn an input-agnostic, task-specific visual prompt v_ϕ on the border of the images (as shown in Fig. 2 (b)), so as to maximize

the likelihood of the correct label y in the training stage: $\max_\phi P(y|x+v_\phi)$. Then during inference stage, optimized visual prompt is added to all test images: $X_{test} = \{x_1 + v_\phi, \ldots, x_n + v_\phi\}$.

Though VP builds an early connection between attack and prompt, we believe the full power of attack is unexplored very well in visual prompt. We first enhance the representation ability by incorporating a simple training strategy (i.e., gradient normalization and augmentation) inspired by attack. Secondly, we find that learnable pixels may obscure information from the original image(e.g., in 2(b), cat ears are obscured by learnable pixels). Hence, we shrink original images to preserve complete information and pad learnable pixels around the image. Both methods can significantly improve the performance over the VP baseline. More results can be found in Section 4.

VPT (Jia et al., 2022) Since we aim to explore the efficacy of visual prompting at the input space, we consider VPT-SHALLOW for a fair comparison, which only inserts prompts into the first transformer layer, namely as VPT. After patch embedding layer, the input $x \in \mathbb{R}^{N\times D}$ contains a learnable class token [CLS] of D-dimension and a sequence of patch embeddings $E = \{e_i | e_i \in \mathbb{R}^D, i = 1, \ldots, N-1\}$. VPT inserts a set of continuous embeddings between CLS and image patch embeddings. Formally, a collection of p prompts is denoted as $P = \{p^k \in \mathbb{R}^d | k \in \mathbb{N}, 1 \leq k \leq p\}$, and the input to Transformer layer can be formulated as: $x = [CLS, P, E]$.

Though VPT achieves competitive performance to linear probe, we notice that the visual prompt in VPT lacks positional information since their learnable tokens are inserted after positional embedding. Importantly, We find that positional embeddings are critical in visual prompting at both pixel and token levels. Results are demonstrated in Section 5.1.

3.2 PROMPT DESIGN

As shown in Fig. 2(c), we first resize the input image to an appropriate size to preserve the origin information. Let $\hat{x} \in \mathbb{R}^{k\times k\times 3}$ be images after shrinking, and $\hat{X} \in \mathbb{R}^{K\times K\times 3}$ be images generated from $\hat{x}$ by padding zero around the image to restore the input size of pre-trained model, where $k < K$. We denote our **E**nhanced **V**isual **P**rompt (EVP) as $V_e \in \mathbb{R}^{K\times K\times 3}$, and value in the location corresponding to $\hat{x}$ is always zero. Therefore, the number of parameters is $(K^2 - k^2) \times 3$. During training, our goal is to maximize the probability $P(y|\hat{X} + V_e)$ of the correct label y. During testing, the optimized visual prompt EVP is added to all test images.

3.3 TRAINING STRATEGY INSPIRED BY ADVERSARIAL ATTACK

In adversarial attack, given an image x_i, the aim is to learn an imperceptible pixel perturbation g_i to mislead the network, which can be formulated as: $\min_{g_i} P(y_i|x_i+g_i)$. In contrast, visual prompting aims to learn a visual prompt template v to maximize the likelihood of the correct label y, which can be considered as the inverse process of adversarial attack: $\max_v P(y|\hat{X} + v)$. Though the attack is input-specific while visual prompting is input-agnostic, there may be some training strategies in adversarial attack which is helpful in visual prompting.

Augmentation Previous work (Xie et al., 2019) shows that input diversity can improve the transferability of adversarial examples. In visual prompting, transferability is essential since the prompt template is input-agnostic. Therefore, we explore various data augmentation to increase input diversity. The range of augmentation is {RandomHorizontalFlip, RandAug (Cubuk et al., 2020), Mixup (Zhang et al., 2018), Cutmix (Yun et al., 2019) }. We find that RandomHorizontalFlip can achieve satisfactory results. More details are shown in the ablation section.

Normalization In adversarial attack, there are various normalization ways (Madry et al., 2018), e.g., L_1 norm, L_2 norm, and L_∞ norm. For example, in L_2 norm, the gradient of x is divided by its L_2 norm, which is shown in Eq. 1.

$$x^{t+1} = x^t + \gamma \frac{\nabla_{x^t} J(x, y)}{||\nabla_{x^t} J(x, y)||_2} \tag{1}$$

where γ is learning rate, J is the loss function, $\nabla_{x^t} J$ is the gradient of the loss function w.r.t. x^t.

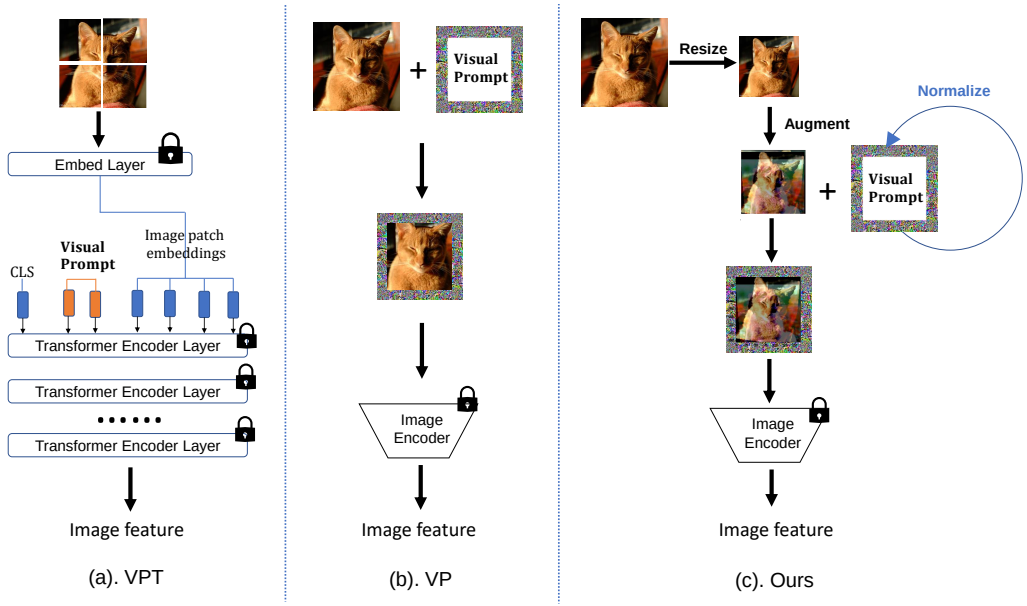


Figure 2: **Overview of different designs of visual prompting.** (a). VPT: Inject some learnable parameters into token space (b): VP: Modify learnable pixels on the border of original images. (c): Our method: Shrink images to preserve the complete semantic information, then apply data augmentation and pad learnable pixel perturbation around the image. The learnable pixels are updated by normalization strategies inspired by the adversarial attack.

Borrowing the normalization strategy in attack, we find that L_2 norm can stabilize the training stage and improve the generalization of visual prompting.

In practice, we define our EVP as $V_e = W \odot M$, where $W \in \mathbb{R}^{K \times K \times 3}$ are parameters that need gradient, and M is a mask matrix where is 1 for the location that corresponds to the prompt location. Then, we find that dividing the gradient of EVP by the L_2 norm of the gradient of W can achieve the best performance:

$$V_e^{t+1} = V_e^t - \gamma \frac{\nabla_{V_e^t} J}{||\nabla_W J||_2} \tag{2}$$

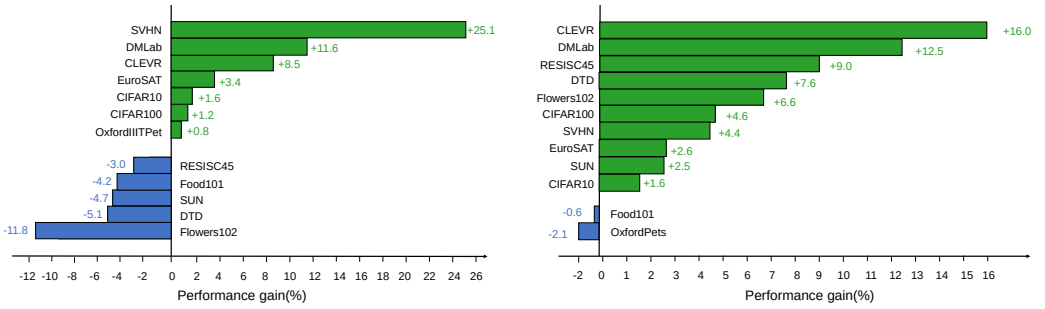
where γ is learning rate, J is the loss function, $\nabla_{V_e^t} J$ and $\nabla_W J$ are the gradient of the loss function w.r.t. V_e^t and the gradient of the loss function w.r.t W, respectively. More details about different normalization strategies are provided in the ablation section.

4 Experiments

4.1 Experimental Setting

Datasets We evaluate our prompt on 17 downstream classification datasets, including 12 classical classification datasets (CIFAR100, CIFAR10 (Krizhevsky et al., 2009), Flowers102 (Nilsback & Zisserman, 2008), Food101 (Bossard et al., 2014), EuroSAT (Helber et al., 2019), SUN397 (Xiao et al., 2010), SVHN (Netzer et al., 2011), DTD (Cimpoi et al., 2014), OxfordPets (Parkhi et al., 2012), Resisc45 (Cheng et al., 2017), CLEVR (Johnson et al., 2017), and DMLab (Beattie et al., 2016)), 3 out-of-distribution datasets (Koh et al., 2021) (Camelyon17, FMoW, and iWildCAM), and 2 corruption datasets (Hendrycks & Dietterich, 2018) (CIFAR100-C and CIFAR10-C).

Baselines We compare with other commonly used fine-tuning protocols: TP (text prompt), VP, VPT, LP (Linear Probe), and FT (Fully fine-tuning). Text prompt is equivalent to zero-shot in CLIP. Linear probe inserts a linear layer as the classification head. Fully fine-tuning updates all backbone and classification head parameters.



(a). EVP vs Linear Probe (a). EVP vs VPT

Figure 3: **Performance gain** of our approach compared to linear probe and VPT on each downstream dataset. The bars indicate the gain (or loss) in accuracy compared to linear probe and VPT, respectively. (a) Compared with linear probe, EVP outperforms linear probe on 7 out of 12 datasets. (b) Compared with VPT, EVP beats the VPT on 10 out of 12 datasets by **5.4%** on average.

4.2 The effectiveness of visual prompt on CLIP

Table 1 presents the full results on 12 classical classification datasets. We can see that:

1 **Our visual prompt outperforms all previous parameter-efficient prompt protocols** with similar or fewer parameters. Specifically, our method outperforms VP and VPT on average by 6.3% and 5.4%, respectively.

2 **Our visual prompt outperforms linear probe**. The table shows that the performance of our methods is higher than the linear probe on 7 out of 12 datasets, and our average accuracy is 82.6%, which is 1.9% higher than the linear probe. In addition, our method is more flexible compared with linear probe, since the number of parameters of our method is basically the same across different datasets, while the number of parameters of linear probe depends on the number of downstream classes.

3 **Our method is more parameter-efficient compared with fully fine-tuning method**. The number of parameters of our prompt is only 0.04% of fully fine-tuning, while the performance is competitive.

Table 1: **Comparisons with previous prompting methods** across 12 datasets on CLIP. EVP outshines the linear probe 7 out of 12 with similar number of parameters. The results where EVP outperforms linear probe are shown in **bold**.

Adaptation	CIFAR100	CIFAR10	Flowers	Food	EuroSAT	SUN	DMLab	SVHN	Pets	DTD	RESISC	CLEVR	Average
TP	63.1	89.0	61.8	83.2	34.1	58.0	30.2	11.0	85.9	42.8	42.4	20.2	51.8
VP	75.3	94.2	62.0	83.2	97.4	60.6	41.9	88.4	85.0	57.1	89.0	81.4	76.3
VPT	76.6	95.0	76.2	84.7	96.1	69.0	48.4	86.1	92.1	60.8	83.3	58.6	77.2
EVP(Ours)	**81.2**	**96.6**	82.3	84.1	**99.0**	71.5	**60.9**	**90.5**	**90.0**	68.4	92.3	**74.6**	82.6
LP	80.0	95.0	94.1	88.3	95.6	76.2	49.3	65.4	89.2	73.5	95.3	66.1	80.7
FT	82.1	95.8	97.4	87.8	95.8	65.3	63.5	95.7	88.5	72.3	89.7	94.4	85.7

4.3 The effectiveness of visual prompt on other models

In general, the last layer of the visual pre-trained model is fixed to a set of predefined classes and requires a separate task-specific head to adapt to the downstream tasks. In our experiments, we aim to explore whether purely visual prompting in input space can adapt pre-trained models to downstream tasks without any modification to pre-trained architecture and weights. VP arbitrarily maps downstream classes to pre-trained classes and discards unassigned classes. However, we hypothesize some similarity between pre-trained and downstream classes, but the correspondence is unknown. Therefore, we design a simple and efficient pre-processing stage before visual prompting.

For each downstream class, we feed downstream images in this class into pre-trained model and investigate the prediction in pre-trained classes. Then, we choose the pre-trained class with the highest prediction frequency as the corresponding class of this downstream class. After pre-processing, we fix the correspondence and train our visual prompting.

Table 2 shows that when using arbitrary mapping, EVP outperforms VP a little. When we apply our pre-processing stage, the performance outperforms EVP and VP significantly. It even matches the

Table 2: **Performance on other models except CLIP**. EVP* indicates that we train EVP using classes after preprocessing stage. EVP slightly exceeds VP and EVP* outperforms EVP and VP by a large margin. The **bold** indicates cases that the performance of EVP* is competitive with linear probe.

Model	Adaptation	CIFAR100	CIFAR10	Flowers	Food	EuroSAT	SUN	SVHN	Pets	DTD	RESISC	CLEVR	Average
Instagram	VP	16.7	62.1	4.8	6.5	85.8	2.2	53.8	18.6	29.1	41.4	30.9	32.0
Instagram	EVP	13.6	67.2	9.2	7.1	87.2	7.9	50.8	16.3	29.0	40.0	48.1	34.2
Instagram	EVP*	**60.3**	**93.5**	11.4	8.4	88.0	19.6	**55.3**	74.4	44.4	48.1	**50.5**	50.4
Instagram	LP	64.0	90.1	92.7	65.8	95.5	58.1	48.0	94.5	70.9	95.7	30.2	73.2
Instagram	FT	77.8	77.8	94.5	75.6	97.4	57.6	96.8	93.9	73.5	93.4	89.3	84.1
RN50	VP	10.1	54.5	4.7	5.1	82.8	1.1	57.1	10.8	8.2	29.9	29.5	26.9
RN50	EVP	9.2	55.9	6.6	3.9	76.2	5.1	48.6	10.5	18.7	26.0	35.5	26.7
RN50	EVP*	38.0	77.0	11.9	7.0	82.5	14.7	47.8	72.0	41.2	40.8	**37.2**	42.7
RN50	LP	67.7	87.7	92.7	62.5	95.8	57.5	60.3	91.1	66.7	92.2	32.6	73.3
RN50	FT	79.9	94.1	96.9	73.2	96.5	55.9	96.9	92.3	66.7	93.4	89.3	84.3

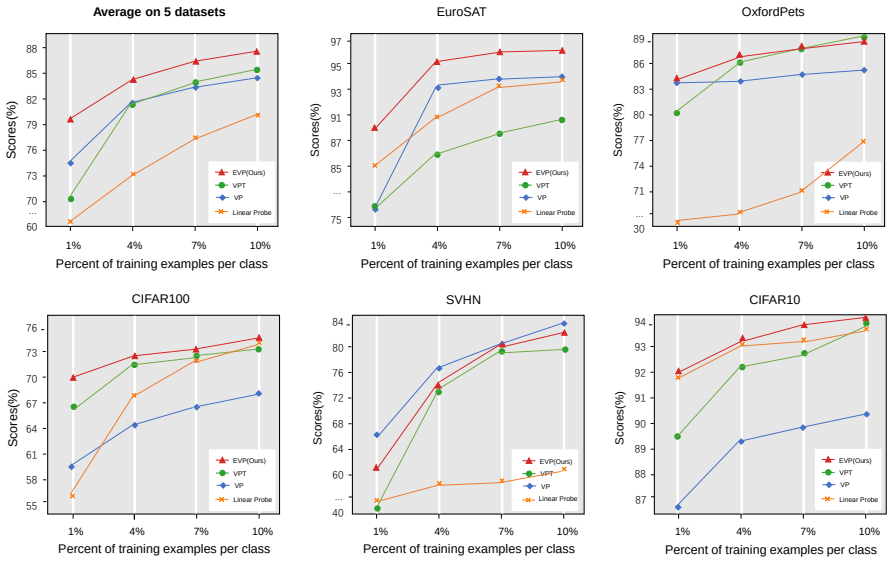


Figure 4: **Results of few-shot learning** on five visual recognition datasets. Each figure shows the few-shot results trained on 1%, 4%, 7%, 10% data respectively. All visual prompt methods show clear dominance compared with linear probe. EVP (red line) outperforms other methods by a large margin on average.

performance of the linear probe on some datasets, indicating that our pre-processing stage chooses some classes that are similar to downstream classes, and the visual prompting can modify the mapping from the pre-trained class to a downstream class. However, our EVP* fails on datasets of fine-grained datasets, like Flowers102 and Food101. We think it is very difficult to find such fine-grained classes from pre-trained classes.

4.4 The effectiveness of visual prompt on few-shot learning

We are interested in whether visual prompting has a good few-shot learning ability. To verify it, we train our prompt using only 1%, 4%, 7%, and 10% data for each class in the training datasets, which is sufficient to observe the trend.

The results are summarized in Fig 4. The results show that: (1). Visual prompting(VP, VPT, EVP) shows clear dominance in few-shot settings compared to linear probe, which shows that visual prompting has a stronger learning ability with limited labeled data. (2). Our method achieves the best performance on average among visual prompting methods, indicating that our strategy of normalization improves the generalization ability.

4.5 The robustness of visual prompt to distribution shift

Since the parameters of the pre-trained model are frozen, visual prompting only provides information about the downstream tasks. The backbone preserves the general pre-trained knowledge, reducing the possibility of overfitting the downstream training set. Hence, we are interested in how our visual prompt compares with other fine-tuning methods in terms of out-of-distribution robust-

Table 3: Robustness comparison on **out-of-distribution** and **corruption** datasets. Left: out-of-distribution datasets. Right: corruption datasets. Compared with **fully fine-tuning**, EVP even achieves better or comparable results on both out-of-distribution setting and corruption setting, which shows the strong robustness of EVP.

Model	Adaptation	iwildcam	camelyon17	fmow	Average
CLIP	TP	12.5	47.3	13.7	24.5
CLIP	VP	57.3	91.4	62.2	37.8
CLIP	VPT	58.8	91.9	29.7	60.1
CLIP	Ours	64.9	**95.1**	40.2	**66.7**
CLIP	LP	**66.7**	86.0	36.3	63.0
CLIP	FT	64.0	84.3	**49.7**	66.0

Model	Adaptation	CIFAR100-C	CIFAR10-C	Average
CLIP	TP	43.5	72.1	57.8
CLIP	VP	52.5	84.1	68.3
CLIP	VPT	54.0	75.8	64.9
CLIP	Ours	58.6	84.3	71.5
CLIP	LP	56.9	78.8	67.9
CLIP	FT	61.1	82.7	71.9

ness. Following prior studies, we use the WILDS benchmark (Koh et al., 2021). We train visual prompts from training datasets from a specific domain. Then we test the model on datasets from a different domain(e.g., images from different regions, cameras, and hospitals). As Table 3 shows, the performance of our method outperforms linear probe on 2 of 3 datasets. The above results verify that keeping the backbone fixed and providing task information with prompt can avoid overfitting.

4.6 The robustness of visual prompt on corruption datasets

We also test the robustness of visual prompt on corruption datasets, CIFAR100-C and CIFAR10-C. These two corruption datasets introduce a set of 19 common visual corruptions and apply them to the object recognition datasets, CIFAR100 and CIFAR10, respectively. These datasets serve as general datasets for benchmarking robustness to image corruptions. We train the visual prompting on datasets CIFAR100 and CIFAR10, then test the performance on corresponding corruption datasets. Results are shown in Table 3. We can see that our EVP outperforms other prompt methods and linear probe, showing the strong robustness of EVP. Since we shrink original images to preserve complete information, our prmopt can focus on the global semantic information rather than the corruption.

5 Ablation on Prompt Design Variants

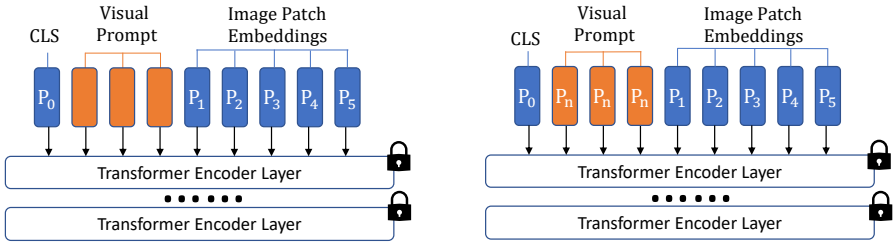


Figure 5: **Ablation on positional embedding at token level.** (a). Visual Prompting Tuning(VPT): Inject learnable tokens between CLS and image patch embedding without positional embedding (b): VP$_n$T: Inject learnable tokens between CLS and image patch embedding with same positional embedding P$_n$ (i.e, n-th positional embedding (n = 1, 2, . . . , 5)).

5.1 Prompt Positional Embedding

A significant distinction between VPT and our method is the positional embeddings(**PE**) of learnable visual prompting. We apply positional embeddings to both image patch embeddings and visual prompting, while VPT only adds positional embeddings to image patch embeddings. We ablate different choices to show that positional embeddings of visual prompting are important, at both pixel and token levels.

At the pixel level, denoting our main method as **EVP-small w/ PE**, then we define two other choices: 1. **EVP-big w/ PE**: We pad learnable pixels around the original image, and interpolate the original positional embeddings to the appropriate size, then we add the positional embeddings to both image patch embeddings and learnable pixels. 2. **EVP-big w/o PE**: We pad learnable pixels around the original image and add positional embeddings only to the image patch embeddings.

Table 4 shows the effect of the positional embedding of visual prompting at the pixel level. We can see that **EVP-big w/ PE** achieves the best performance, while **EVP-big w/o PE** is the worst. **EVP-small w/ PE** outperforms **EVP-big w/o PE** though the number of parameters is less and the resolution is lower, demonstrating the efficacy of positional embedding.

Table 4: **Ablation on positional embedding at the pixel level**. EVP-small shrinks the image and pad it with learnble pixels back to the origin size, while EVP-big pads pixel patches around the origin image. EVP-small w/ PE even can beat EVP-big w/o PE with fewer the number of parameters and smaller input resolution, indicating that the positional embedding are crucial.

Methods	CIFAR100	CIFAR10	DTD	RESISC	EuroSAT	Average
EVP-small w/ PE	81.2	96.6	68.4	92.3	99.0	87.5
EVP-big w/ PE	81.4	96.9	68.9	93.3	99.0	87.9
EVP-big w/o PE	73.4	93.7	64.6	83.1	92.0	82.4

Table 5: **Ablation on positional embedding at token level**. VPT only adds positional embeddings to the image patch embeddings, while VP_1T, VP_{25}T, VP_{50}T indicate that we add the *1*-st, *25*-th, *50*-th positional embedding to the learnable tokens, respectively. Simply adding positional embeddings to learnable tokens can improve the performance significantly.

Methods	CIFAR100	CIFAR10	DTD	RESISC	EuroSAT	Average
VPT	76.6	95.0	60.8	83.3	96.1	82.4
VP_1T	**77.3**	**96.0**	**67.7**	**88.3**	**96.7**	**85.2**
VP_{25}T	76.8	95.5	66.6	88.1	96.1	84.6
VP_{50}T	77.0	**96.0**	66.4	87.2	96.3	84.6

Table 6: **Ablation on augmentation.** We use CLIP-Base/32 as pre-trained model and evaluate on CIFAR100. RandomFilp works the best. Stronger augmentation degrades performance.

Augmentation				Performance
Flip	Mixup	RandAug	CutMix	
✗	✗	✗	✗	80.5
✔	✗	✗	✗	**81.2**
✔	✔	✗	✗	79.6
✔	✗	✔	✗	79.4
✔	✗	✗	✔	79.7

Table 7: **Ablation on gradient normalization**. Applying L2 norm on gradient can significant improve performance. Adapting whole image's gradient to normalize can improve further.

Gradient Normalization				Performance
L_1	L_∞	L_2-partial	L_2-whole	
✗	✗	✗	✗	77.5
✔	✗	✗	✗	77.2
✗	✔	✗	✗	71.9
✗	✗	✔	✗	79.4
✗	✗	✗	✔	**81.2**

At the token level, we find that simply adding positional embeddings to learnable tokens can improve performance. Since the positional information of Transformer architecture is only dependent on the positional embeddings, adding different positional embeddings to learnable tokens indicates inserting learnable tokens into different positions. Based on this, we design different prompt choices at the token level by adding different positional embeddings to the learnable tokens. Specifically, we denote prompt choices as VP_nT, which means that we add the *n*-th positional embeddings to the learnable tokens, which is shown in Fig. 5.

Table 5 shows the results. We can see that adding the positional embeddings to learnable tokens in different ways can improve the performance significantly, indicating that the positional embeddings are significant in visual prompting.

5.2 Ablation on training strategy

In adversarial attack, data augmentation can improve the transferability of adversarial examples. We aim to borrow this strategies to improve visual prompting. We ablate augmentation methods, like RandomHorizontalFlip, mixup, randAug, and cutmix on CIFAR100 dataset. As shown in Table 6, we find that simple RandomHorizontalFlip can achieve satisfactory results, and some strong augmentation like mixup or randAug may decrease the performance.

In adversarial attack, there are many normalization strategies, e.g. L_1 norm, L_2 norm, and L_∞ norm. Table 7 shows the L_2 norm achieves the best performance among all strategies on CIFAR100. In addition, We explored what the optimal L_2 norm consists of. We find that using the whole gradient of the image to calculate L_2 norm (L_2-whole) is better than using the gradient of the visual prompting pixels (L_2-partial).

6 Conclusion

We propose EVP, a new parameter-efficient visual prompting method at the pixel level to adapt large pre-trained vision model to downstream tasks. EVP builds the connection between adversarial attack and visual prompting, then borrows training strategies from attack to improve the transferability. In addition, EVP preserves the complete information and keeps the same FLOPs with origin images. We show that EVP can outperform other visual prompting methods at the input space and surpass linear probe on many settings.

REFERENCES

Hyojin Bahng, Ali Jahanian, Swami Sankaranarayanan, and Phillip Isola. Exploring Visual Prompts for Adapting Large-Scale Models. In *arxiv*, 2022.

Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. BEiT: BERT Pre-Training of Image Transformers. In *ICLR*, 2021.

Charles Beattie, Joel Z Leibo, Denis Teplyashin, Tom Ward, Marcus Wainwright, Heinrich Küttler, Andrew Lefrancq, Simon Green, Víctor Valdés, Amir Sadik, et al. Deepmind lab. In *arXiv*, 2016.

Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In *ECML-PKDD*, 2013.

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

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. In *NeurIPS*, 2020.

Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote sensing image scene classification: Benchmark and state of the art. In *Proc. IEEE*, 2017.

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

Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *CVPR workshops*, pp. 702–703, 2020.

Leyang Cui, Yu Wu, Jian Liu, Sen Yang, and Yue Zhang. Template-Based Named Entity Recognition Using BART. In *ACL-IJCNLP*, 2021.

Nilesh Dalvi, Pedro Domingos, Sumit Sanghai, and Deepak Verma. Adversarial classification. In *KDD*, 2004.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *arxiv*, 2018.

Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Xiaolin Hu, and Jun Zhu. Discovering Adversarial Examples with Momentum. In *CVPR*, 2017.

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.

Gamaleldin F Elsayed, Ian Goodfellow, and Jascha Sohl-Dickstein. Adversarial Reprogramming of Neural Networks. In *ICLR*, 2018.

Tianyu Gao, Adam Fisch, and Danqi Chen. Making Pre-trained Language Models Better Few-shot Learners. In *ACL*, 2021.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and Harnessing Adversarial Examples. In *ICLR*, 2015.

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

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross B. Girshick. Masked Autoencoders Are Scalable Vision Learners. In *CVPR*, 2022.

Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. In *IEEE J-STARS*, 2019.

Dan Hendrycks and Thomas Dietterich. Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. In *ICLR*, 2018.

Ling Huang, Anthony D Joseph, Blaine Nelson, Benjamin IP Rubinstein, and J Doug Tygar. Adversarial machine learning. In *Proceedings of the 4th ACM workshop on Security and artificial intelligence*, 2011.

Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual Prompt Tuning. In *ECCV*, 2022.

Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*, 2017.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *ICML*, 2021.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. In *Citeseer*, 2009.

Brian Lester, Rami Al-Rfou, and Noah Constant. The Power of Scale for Parameter-Efficient Prompt Tuning. In *EMNLP*, 2021.

Xiang Lisa Li and Percy Liang. Prefix-Tuning: Optimizing Continuous Prompts for Generation. In *ACL-IJCNLP*, 2021.

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. In *arxiv*, 2021a.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Zhengxiao Du, Zhilin Yang, and Jie Tang. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. In *arXiv*, 2021b.

Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *ICLR*, 2017.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards Deep Learning Models Resistant to Adversarial Attacks. In *ICLR*, 2018.

Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens Van Der Maaten. Exploring the limits of weakly supervised pretraining. In *ECCV*, 2018.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *NeurIPS*, 2011.

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *ICVGIP*, 2008.

Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *CVPR*, 2012.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. Language Models as Knowledge Bases? In *EMNLP-IJCNLP*, 2019.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision. In *PMLR*, 2021.

Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010.

Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L Yuille. Improving transferability of adversarial examples with input diversity. In *CVPR*, 2019.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, 2019.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond Empirical Risk Minimization. In *ICLR*, 2018.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. In *IJCV*, 2022.

A Appendix

Implementation details We implement all experiments in Python using Pytorch (Paszke et al., 2019) framework. We use CLIP-B/32, Instagram (Mahajan et al., 2018), and ResNet50 (He et al., 2016) as our pre-trained model, and the batch size is 256, 32, 128, respectively. All visual prompts in our experiments are trained for 1000 epochs. For EVP, we use SGD with a learning rate of 70, which decayed using cosine schedule (Loshchilov & Hutter, 2017). The prompt size is 30 pixels by default. To compare fairly with VP, we follow the text prompt as VP (Bahng et al., 2022) in CLIP model. Specifically, we use "This is a photo of a [LABEL]" as default for the text prompt. For CLEVR datasets, we use "This is a photo of [LABEL] objects", for DMLab datasets, we use "The distance is [LABEL1], and the reward is [LABEL2]", and for Camelyon17, the text prompt template is "a tissue region [LABEL] tumor".

Prompt Size The prompt size is a hyper-parameter for EVP. We define prompt size $p = \frac{K-k}{2}$, where k is the image size after shirinking, and K is the input size of pre-trained model. Therefore, the number of parameters is $12p(K-p)$, which only depends on p since K is fixed for a given model. In our experiment, the optimal prompt size varies across datasets, as shown in Fig. 6. Since we shrink the original image and pad learnable pixels around it, the image resolution and the number of parameters are traded off essentially. In some datasets with low resolution(e.g., CIFAR100), the prompt of p=30 achieves the best performance. However, in some datasets wth high resolution, decreased resolution may lead to decreased performance. We find that p=5 is the best in Food101 with resolution 512×512.

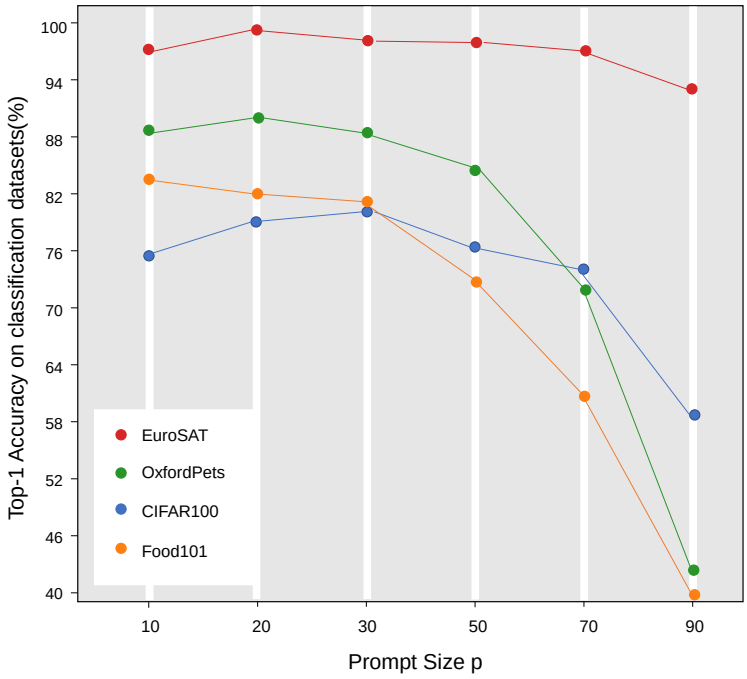


Figure 6: **Ablation on prompt size.** The pre-trained model is CLIP-B/32. We vary the prompt size, which determines the number of parameters, and show the performance on four datasets. The best prompt size varies across datasets.