

One Prompt Word is Enough to Boost Adversarial Robustness for Pre-trained Vision-Language Models

Lin Li^{1*}, Haoyan Guan^{1*}, Jianing Qiu², Michael Spratling¹

¹King’s College London, ²Imperial College London

{lin.3.li, haoyan.guan, michael.spratling}@kcl.ac.uk, jianing.qiu17@imperial.ac.uk

Abstract

Large pre-trained Vision-Language Models (VLMs) like CLIP, despite having remarkable generalization ability, are highly vulnerable to adversarial examples. This work studies the adversarial robustness of VLMs from the novel perspective of the text prompt instead of the extensively studied model weights (frozen in this work). We first show that the effectiveness of both adversarial attack and defense are sensitive to the used text prompt. Inspired by this, we propose a method to improve resilience to adversarial attacks by learning a robust text prompt for VLMs. The proposed method, named Adversarial Prompt Tuning (APT), is effective while being both computationally and data efficient. Extensive experiments are conducted across 15 datasets and 4 data sparsity schemes (from 1-shot to full training data settings) to show APT’s superiority over hand-engineered prompts and other state-of-the-art adaptation methods. APT demonstrated excellent abilities in terms of the in-distribution performance and the generalization under input distribution shift and across datasets. Surprisingly, by simply adding one learned word to the prompts, APT can significantly boost the accuracy and robustness ($\epsilon = 4/255$) over the hand-engineered prompts by +13% and +8.5% on average respectively. The improvement further increases, in our most effective setting, to +26.4% for accuracy and +16.7% for robustness. Code is available at <https://github.com/TreeLLi/APT>.

1. Introduction

Large pre-trained Vision-Language Models (VLMs) such as CLIP [51], ALIGN [27], BLIP [33], etc. have emerged as general-purpose (a.k.a. foundation) models [5], fostering ecosystems across numerous sectors within the realm of artificial intelligence [5, 50]. As more research and applications build upon these foundation models, any failures or vulnerabilities inherent in them can cause cascading impacts on the performance and reliability of the down-

*equal contribution

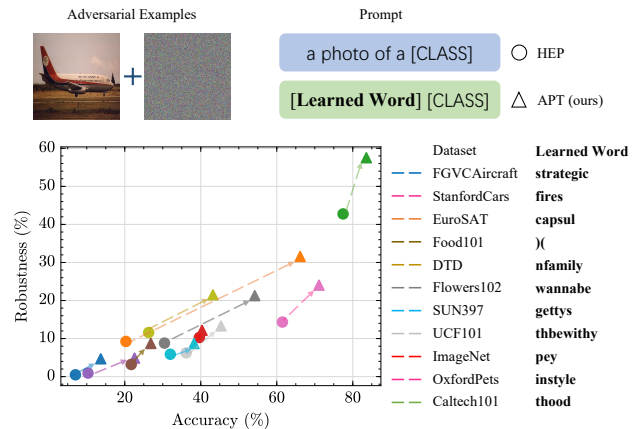


Figure 1. Adding a learned “word” to prompts boosts both accuracy and robustness ($\epsilon = 4/255$) substantially over hand-engineered prompts (HEP) across 11 datasets. The dashed arrows indicate the performance boost. A “word” is a learnable vector, which is interpreted in the last column of the figure.

stream tasks. A critical issue unveiled by the recent studies [26, 45, 54, 70] is that these VLMs, like vision models, are highly vulnerable to adversarial examples [57]. Their output can be manipulated by human-imperceptible perturbations to the image [45, 54], posing substantial safety implications and thereby raising serious concerns about the reliability and security of these models.

A prevalent paradigm [5] for the deployment of contemporary VLMs involves the initial pre-training of large models on large-scale datasets, followed by adapting for specific downstream tasks. Adaption is vital as it can often largely boost the performance for downstream tasks. A well-established approach to adaptation is fine-tuning the model weights [66]. However, fine-tuning all the model weights becomes prohibitively costly as pre-trained models scale up to tens or hundreds of billions of parameters, and can be even more unaffordable if adversarial training [43] is applied to improve adversarial robustness. Besides, fine-tuning may distort pre-trained features, and thus, hurt the out-of-distribution generalization performance [32]. There-

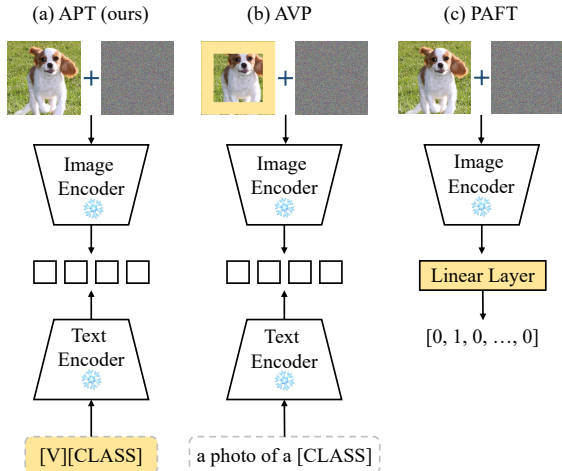


Figure 2. A high-level architectural comparison between our method Adversarial Prompt Tuning (APT), Adversarial Visual Prompting (AVP), and Partial Adversarial Fine-Tuning (PAFT). The learnable parameters are highlighted in yellow. Note that PAFT discards the entire text branch of CLIP.

fore, parameter-efficient adaption methods [15] that freeze all or most model weights become a promising solution.

This work studies the problem of *parameter-efficient adaption of pre-trained VLMs for adversarial robustness*. Current adaption methods for adversarial robustness focus on the model weights, *i.e.*, adversarial fine-tuning [10, 22, 28, 30, 41] or image pixels, *i.e.*, adversarial visual prompting [9, 25]. The text input to VLMs, a.k.a. prompt, has been rarely studied before for adversarial robustness despite its significant impact on the accuracy of VLMs [72] and advantages such as naively supported by VLMs (so no need to modify architecture), parameter efficiency, *etc.* This work aims to fill this gap by studying the effect of text prompt in adversarial robustness and proposing a new method to tune text prompt for improving adversarial robustness (see Fig. 2). We focus on a category of VLMs resembling CLIP [51] as it represents a quintessential vision-language foundation model and has been used in many applications.

We start by investigating how the text prompt influences adversarial attack and defense on CLIP. Our key findings include: 1) the strength of adversarial attack is sensitive to the prompt used for generating adversarial examples; 2) the strongest adversarial examples are almost all generated when the prompt used for attack is the same as the prompt used by the victim model during inference; 3) the adversarial robustness of CLIP is sensitive to the prompt used for inference. The former two findings shed light on how to prompt for strong adversarial attack.

The last finding leads us to propose **Adversarial Prompt Tuning (APT)** to learn robust text prompts for CLIP based on adversarial examples to improve its adversarial robustness. APT parameterizes prompts in the form of soft prompts [39], *i.e.*, concatenating the class embedding with

a sequence of learnable vectors (illustrated in Fig. 3). These vectors constitute the *context* description of the data and class. They can be unified to be shared by all classes or specific to each class. Three different prompting strategies are then proposed to generate training adversarial examples on which the learnable vectors are optimized to minimize the predictive loss like CrossEntropy. Ultimately, the best prompting strategy we adopted is to generate training adversarial examples based on the latest updated prompts.

Extensive experiments are conducted to benchmark APT across 15 datasets and 4 data sparsity schemes, 1-, 4- and 16-shot learning and training with the entire training set. APT is compared against the hand-engineered prompts proposed in CLIP [51] and the state-of-the-art adaption methods other than text prompting. APT is found to outperform these alternative methods in terms of the in-distribution performance and the generalization ability under distribution shift (the same classes yet different input distribution) and across datasets (different classes). Three promising properties of APT are highlighted below:

- **Parameter-efficient:** one prompt word is enough to boost performance substantially (see Fig. 1).
- **Data-efficient:** one shot is enough to boost performance considerably.
- **Effective:** large performance boost and excellent trade-off between accuracy and robustness.

Overall, our work paves a new way for enhancing adversarial robustness for VLMs through text prompting.

2. Related Works

Adapting pre-trained models for accuracy. In contrast to the traditional approach to fine-tune the entire model’s parameters [58], parameter-efficient adaptation methods are investigated. Current parameter-efficient methods mainly contain three categories: prompt tuning [72], adapter tuning [24, 38] and linear probing [32]. Prompt tuning modifies the input to adapt the model. According to the modality of the input, prompt tuning can be categorized into visual prompting [7, 40, 67, 69, 73] for image input and text prompting [37, 42, 47, 49, 59] and for text input. Adapter tuning inserts a small learnable module in the model to be trained for downstream tasks. Linear probing is performed by training only a linear layer attached to the end of the model. In this paper, we investigate the application of text-driven prompt learning as a strategy for defending against adversarial attacks in the context of image recognition.

Adversarial training [19] has been so far the most effective defense against adversarial examples [2]. It replaces the clean examples with adversarial examples generated on-the-fly during training. Adversarial training is well known to be expensive [43] and prone to overfitting [53, 62]. Numerous methods have been proposed to improve the efficiency [1, 29, 34, 55, 62] and/or the effective-

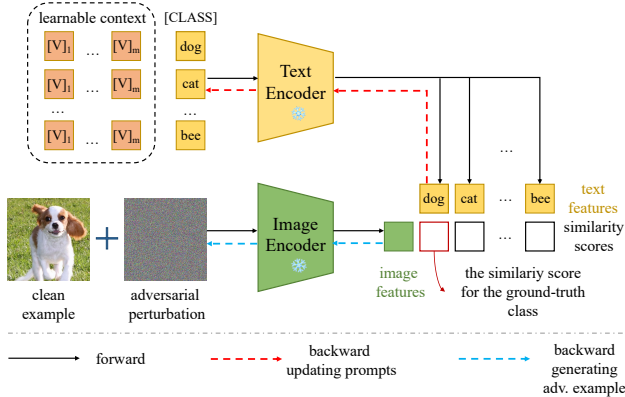


Figure 3. An overview of the proposed Adversarial Prompt Tuning (APT) method on CLIP-like VLMs. Both image and text encoders are frozen and only the prompt contexts are learnable. The learnable context can be unified for all classes or specific to each class.

ness [35, 43, 61, 63, 68] of the algorithm. However, most of them train models from scratch, while the adaption of pre-trained models for adversarial robustness is less studied. A line of works [10, 22, 28, 30, 41] adapt pre-trained models for adversarial robustness by adversarial fine-tuning: fine-tuning model weights by adversarial training. According to the amount of parameters to be tuned, those methods are categorized as full adversarial fine-tuning and partial adversarial fine-tuning [10]. Alternatively, Chen et al. [9] and Huang et al. [25] explore adversarial visual prompting [3] as a test-time defense to enhance adversarial robustness for pre-trained models. Our method aims at adapting pre-trained models by tuning text prompts, differing from the above works that adapt model weights or input images. More works on the adversarial robustness of VLMs are reviewed in Appendix A.

3. Text Prompt for Adversarial Robustness

3.1. Review of CLIP

As shown in Fig. 3, CLIP consists of two primary components: an image encoder and a text encoder, parameterized by θ_v and θ_t respectively. They are used to extract the features from images and text respectively. Given an input image \mathbf{x}_i and text \mathbf{t}_j , the respective features \mathbf{z}_v^i and \mathbf{z}_t^j are computed as:

$$\mathbf{z}_v^i = f(\mathbf{x}_i; \theta_v), \quad \mathbf{z}_t^j = f(\mathbf{t}_j; \theta_t) \quad (1)$$

A cosine similarity score is then calculated for each pair of image and text features to measure their alignment:

$$s_{i,j} = \cos(\mathbf{z}_v^i, \mathbf{z}_t^j) \quad (2)$$

These similarity scores are analogous to the logit output of the classical vision model like ResNet [20]. The probability of \mathbf{x}_i aligning with \mathbf{t}_j is:

$$p_{i,j} = p(\mathbf{x}_i, \mathbf{t}_j) = \frac{\exp(s_{i,j})}{\sum_j \exp(s_{i,j})} \quad (3)$$

Algorithm 1 Pseudo-code for ℓ_∞ adversarial attack on CLIP. Text is perturbed if *perturb_t* is true. K is the step number. α (α') is the step size for perturbing image (text).

```

1: function ATTACK( $\mathbf{x}, \mathbf{y}, \mathbf{t}, \text{perturb}_t$ )
2:    $\delta = \text{uniform}(-\epsilon, \epsilon)$   $\triangleright$  perturbation at image pixels
3:    $\delta' = \mathbf{0}$   $\triangleright$  perturbation at word embeddings
4:   for  $1 \rightarrow K$  do
5:      $\mathbf{x}' = \min(0, \max(\mathbf{x} + \delta, 1))$ 
6:      $L = \mathcal{L}(\mathbf{x}', \mathbf{t} + \delta', \mathbf{y}; \theta_v, \theta_t)$ 
7:      $\delta = \min(-\epsilon, \max(\delta + \alpha \cdot \text{SIGN}(\nabla_{\mathbf{x}} L), \epsilon))$ 
8:     if perturb_t then  $\triangleright$  jointly perturb prompt
9:        $\delta' = \delta' + \alpha' \cdot \nabla_{\mathbf{t}} L$ 
10:    end if
11:  end for
12:  return  $\min(0, \max(\mathbf{x} + \delta, 1))$ 
13: end function

```

Two encoders are jointly pre-trained by maximizing the similarity scores for true image-text pairs, *i.e.*, $i = j$ while minimizing the similarity scores for false pairs. Once pre-trained, CLIP can be applied to perform zero-shot image classification by using the text description of the classes in the target dataset as the text prompts and predicting the most probable class:

$$\arg \max_j p_{i,j} \quad (4)$$

By default, CLIP constructs the prompt for each class using a template of “a photo of a [CLASS]” where [CLASS] is the name of a class. We define the content in the prompt other than [CLASS] as the *context*. A prompt can be formulated as:

$$\mathbf{t}_j = [\text{context}_{\text{front}}][\text{CLASS}_j][\text{context}_{\text{end}}] \quad (5)$$

Theoretically, the context can be arbitrary which provides a new dimension for adapting a frozen, pre-trained, VLM. Empirically, it has been shown that tuning the text prompt context can significantly impact the performance on the target dataset [71, 72]. Note that some specific details are ignored in the above review for simplicity. Please refer to the original work of CLIP [51] for the complete specification.

3.2. The Sensitivity of Robustness to Prompts

A common strategy [45, 70] to generate adversarial examples for VLMs is to search for a perturbation δ_i for input \mathbf{x}_i to maximize the (cosine) dissimilarity between the image feature, \mathbf{z}_v^i , and the text feature of the corresponding ground-truth class prompt, $\mathbf{z}_t^{y_i}$. Assuming δ is bounded by the ϵ -ball of the p -norm, it can be formulated as:

$$\arg \max_{\|\delta_i\|_p \leq \epsilon} \mathcal{L}(\mathbf{x}_i + \delta_i, \mathbf{t}', y_i; \theta_v, \theta_t) \quad (6)$$

This differs from the conventional formulation [34] due to the presence of the text encoder, θ_t , and text prompt, \mathbf{t}' (which can be different from the one used for inference, \mathbf{t} ,

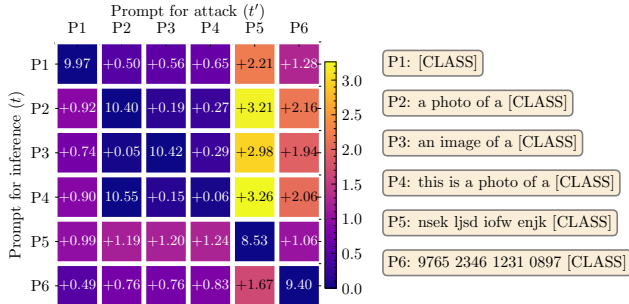


Figure 4. The robustness averaged over 11 datasets of pre-trained CLIP as varied prompts are used for inference, t , (rows) and adversarial attack, t' , (columns). The image encoder backbone is ViT-B/32. Robustness is evaluated against PGD100. Prompts 1 to 4 are manually constructed. Prompts 5 and 6 are randomly sampled from English characters and numbers respectively. For each row, the cell of the most malicious t' , *i.e.*, with the lowest robustness is annotated by the absolute robustness while the rest are annotated by the relative robustness, *i.e.*, the amount exceeding the row minimum. Cells are colored according to the relative robustness.

in Eq. (3)). The effectiveness of adversarial examples generated by Eq. (6) is dependent on the text encoder and text prompt since the gradients used for constructing adversarial examples are dependent (due to Eq. (2)) on the text features. Nevertheless, the influence of the text encoder is fixed and can be ignored as in this work its weights are frozen after pre-training. An implementation of the above attack algorithm is illustrated in Algorithm 1.

Now the question is how t' should be selected to maximize the strength of the attack. A common choice [45, 70] is to use the same prompt as the one used for inference assuming that the attackers have access to this information, *i.e.* a white-box threat model. To validate this, we fix the prompt for inference and vary the prompt for attack. It is observed (see Fig. 4) that the strength of the attack is sensitive to t' . The robustness can vary a lot when different t' are used to attack the same t . Importantly, the lowest robustness is achieved when $t' = t$ in all cases except for the inference prompt P4. Nevertheless, in that case, the gap between the robustness when using the attack prompt P4 (*i.e.* the same inference and attack prompts) and the lowest robustness (produced by the attack prompt P2) is very small, 0.06%. It is therefore vital for attackers to have access to the prompts used by the model users to construct strong attack.

Another intriguing observation in Fig. 4 is that the (lowest) robustness varies with the prompt for inference. For instance, by simply changing the inference prompt from P5 (“nsek ljsd iofw enjk [CLASS]”) to P4 (“this is a photo of a [CLASS]”), the worst-case robustness (evaluated by $t' = t$) increases from 8.53% to 10.55%.

4. Adversarial Prompt Tuning (APT)

Motivated by the above observation, we hypothesize that the adversarial robustness of VLMs is sensitive to the text prompt used for inference, t . Therefore, we propose to improve the adversarial robustness of VLMs through adversarially tuning the prompt. Specifically, we aim at learning text prompt contexts that make the model more robust to adversarial attacks.

4.1. Prompt Parameterization

We first parameterize the context in a text prompt (Eq. (5)) to be learnable. Following Zhou et al. [72], the context of a class C_j is formulated by a sequence of M vectors, $[V]_{m,j}$ ($m \in 1, \dots, M$), defined in the word embedding space rather than as raw text. This enables the parameters to be continuous for more flexibility compared to the discrete ones of the textual formulation. Each vector has the same dimension as a word embedding, *i.e.*, 512 for CLIP. The final input to the text encoder is the concatenation of the context vectors and the word embedding of the class or the class embedding for short, $[C_j]$, as

$$t_j = [V]_{1,j} \dots [V]_{m,j} [C_j] \quad (7)$$

Theoretically, $[C_j]$ can be placed at an arbitrary position inside the sequence of context vectors. For simplicity, we only test three positions: *front*, *middle* and *end*. Empirically, no distinction is observed among the results for these three positions (see Appendix C.3.2), so *end* is used by default.

Furthermore, we employ two variants of context parameterization: *Unified Context* (UC) and *Class-Specific Context* (CSC). In UC, the same context vectors are shared by all classes so there is only one sequence of context vectors no matter how many classes are used. In contrast, CSC assigns separate context vectors to each class so different classes are allowed to have different, tailored, contexts. The number of parameters for CSC increases linearly with the number of classes, C , in the dataset. Given the same context length, the CSC variant has C times more parameters than the UC variant. This may benefit learning complicated tasks but at the expense of requiring more training data to mitigate overfitting. A detailed empirical comparison is given in Sec. 5.1, but in summary, UC (CSC) is more effective when the training data is limited (abundant).

4.2. Prompt Optimization

To improve adversarial robustness, we train the prompt contexts using adversarial training [43]:

$$\arg \min_t \mathbb{E}_{i \in B} \mathcal{L}(x_i + \delta_i, t, y_i; \theta_v, \theta_t) \quad (8)$$

Where the perturbation δ_i is generated on-the-fly by a training adversary as illustrated in Algorithm 2. Inside the prompt t , only the context vectors have learnable parameters while the class embeddings are constant so optimizing the prompt is essentially optimizing the context vectors.

Note that Eq. (8) can be easily extended to alternative adversarial training methods like TRADES [68].

Algorithm 2 Pseudo-code of APT. v is the learnable context vectors. $\text{ATTACK}(\cdot)$ is defined in Algorithm 1.

```

1: function TRAIN_ONE_ITERATION( $x, y$ )
2:    $t = G(\text{“[CLASS]”})$   $\triangleright$  text to word embeddings
3:    $t = [v, t]$   $\triangleright$  join context and class embedding
4:   if constant then
5:      $t' = G(\text{“a photo of a [CLASS]”})$ 
6:      $x = \text{ATTACK}(x, y, t', \text{false})$ 
7:   else if on-the-fly then
8:      $x = \text{ATTACK}(x, y, t, \text{false})$ 
9:   else if perturbed then
10:     $x = \text{ATTACK}(x, y, t, \text{true})$ 
11:  end if
12:   $L = \mathcal{L}(x, t, y; \theta_v, \theta_t)$ 
13:   $v = v - \ell \cdot \nabla_v L$   $\triangleright \ell$  is learning rate
14: end function

```

The key design choice in Eq. (8) is the algorithm for generating δ_i . As discussed in Sec. 3.2, δ_i is dependent on the text prompt t' used for attack that can be different from t in Eq. (8). We propose three potential prompting strategies for generating training adversarial examples: *constant*, *on-the-fly* and *perturbed* as formulated below respectively.

$$\arg \max_{\|\delta_i\|_p \leq \epsilon} \mathcal{L}(x_i + \delta_i, t^*, y_i; \theta_v, \theta_t) \quad (9)$$

$$\arg \max_{\|\delta_i\|_p \leq \epsilon} \mathcal{L}(x_i + \delta_i, t, y_i; \theta_v, \theta_t) \quad (10)$$

$$\arg \max_{\|\delta_i\|_p \leq \epsilon, \delta'} \mathcal{L}(x_i + \delta_i, t + \delta', y_i; \theta_v, \theta_t) \quad (11)$$

The strategy *constant* fixes the prompt for attack to a pre-defined one, “a photo of a [CLASS]” in this case. The perturbation generated by this strategy for each image is constant during training regardless of the inference prompts since both model weights and attack prompts are fixed. This enables the reuse of adversarial image features and thus accelerates the prompt tuning process. However, it may not benefit or even hurt the performance as the perturbation now is no longer dynamically adversarial. In contrast, the strategy *on-the-fly* generates adversarial examples based on the latest, updated, text prompts, t from Eq. (8). This is the exact method used for adversarial evaluation, as discussed in Sec. 3.2. Last, the strategy *perturbed*, a.k.a. multimodal adversarial attack [18], perturbs both images and text prompts (on top of the strategy *on-the-fly*) to further enlarge the adversarial loss and hopefully to generate stronger adversarial examples. This strategy was adopted before by Gan et al. [18] for adversarially training model weights. The algorithms for adversaries based on the above prompting strategies are illustrated in Algorithm 1.

Efficiency analysis. Analogous to adversarial training on model weights, the expense of APT is twofold: adversarial generation and prompt update. Supposing K iterations of inner maximization, the cost of adversarial generation for the strategies *constant* and *on-the-fly* is K times the number of forward and backward passes of the image encoder. The strategy *perturbed* costs an extra K forward and backward passes of the text encoder in addition to the expense of the strategy *on-the-fly* for perturbing text prompts. In practice, adversarial perturbation per image in the strategy *constant*, once generated, can be cached and reused throughout training since the text prompt is constant. In contrast, adversarial perturbation has to be generated on-the-fly in each iteration for the other two strategies because the text prompt is variable. The cost of prompt update is the same for all strategies: one forward pass of the image and text encoders plus one backward pass of the text encoder.

A performance comparison among the above strategies is conducted in Appendix C.3.3. Empirically, the strategy *on-the-fly* matches the effectiveness of strategy *perturbed* while being much more effective than strategy *constant*. The strategy *on-the-fly* is used by default as it achieves the best trade-off between effectiveness and efficiency.

5. Experiments

The experiments in this section were based on the following setup (more details in Appendix B) unless otherwise specified. Following Zhou et al. [72], 11 datasets were used to evaluate our method: ImageNet [14], Caltech101 [17], OxfordPets [48], StanfordCars [31], Flowers102 [46], Food101 [6], FGVCaircraft [44], SUN397 [64], DTD [11], EuroSAT [21] and UCF101 [56]. For each dataset, we evaluate with N -shots, meaning N examples per class are randomly sampled from the entire training set for training. N was either 1, 4, 16 or “all”, where the last means the entire training set was used. One exception was for ImageNet, where 100-shots was used instead of “all” because our computational resource is insufficient to run experiments on the full dataset. All methods are evaluated on the entire test set regardless of the training data scheme used.

Models. The default backbone for the image encoder is ViT-B/32 [16]. The weights of image encoders were pre-trained using the state-of-the-art zero-shot adversarial robustness method TeCoA [45]. The necessity of robust pre-training is discussed in Appendix C.6.

Adversarial training and evaluation. The PGD [43] attack is used for both training and evaluation. Two perturbation budgets, $\epsilon = 1/255$ and $4/255$, are used following Mao et al. [45] and Croce et al. [13] respectively. We use 3 steps with a step size of $2\epsilon/3$ for training and 100 steps with a step size of $\epsilon/4$ and random start for evaluation. The inference prompts are used for attack as discussed in Sec. 3.2.

Competitive methods. The proposed method is a text-

Table 1. The average performance for different ϵ and shots. The context length, M , for our methods is 16. The **best** and second best results are highlighted under each metric. HEP are manually tuned on the target dataset so no strict control on the number of shots used. The results of HEP are copied under different shots in the table for the convenience of comparison.

ϵ	Method	1 shot		4 shots		16 shots		All	
		Acc.	Rob.	Acc.	Rob.	Acc.	Rob.	Acc.	Rob.
1/255	HEP [51]	45.2	<u>32.1</u>	45.2	32.1	45.2	32.1	45.2	32.1
	AVP [9]	44.6	31.6	45.0	32.4	45.7	33.6	50.6	39.0
	PAFT [10]	30.6	21.7	46.9	34.4	66.4	51.0	<u>71.1</u>	<u>56.9</u>
	APT-UC	51.3	35.1	58.2	40.8	66.5	49.0	70.9	54.3
	APT-CSC	39.9	26.2	<u>54.3</u>	<u>37.8</u>	66.6	49.1	73.5	57.1
4/255	HEP [51]	<u>33.0</u>	10.3	33.0	10.3	33.0	10.3	33.0	10.3
	AVP [9]	32.2	<u>10.5</u>	32.4	10.8	32.7	11.3	34.4	13.1
	PAFT [10]	19.2	8.5	32.4	13.9	<u>51.5</u>	22.9	<u>54.9</u>	27.5
	APT-UC	33.1	11.4	<u>41.8</u>	15.2	51.1	20.2	<u>54.9</u>	24.8
	APT-CSC	28.1	8.1	41.9	<u>14.4</u>	54.2	<u>20.7</u>	59.4	<u>27.0</u>

prompting-based parameter-efficient adaption method so it is compared against two categories of related works: text prompting and parameter-efficient adaption methods. For text prompting, we compare our method against Hand-Engineered Prompts (HEP) which was originally proposed in CLIP and has been widely used subsequently [45, 70–72]. The specific prompts used for each dataset are described in Appendix B. For parameter-efficient adaption methods, we adopt Adversarial Visual Prompting (AVP) [9] and Partial Adversarial Fine-Tuning (PAFT) [10] for comparison. PAFT can be also viewed as the adversarial training variant of linear probing [51]. A high-level architectural comparison between these adaption methods is shown in Fig. 2. The specification of AVP and PAFT is described in Appendix A. All compared methods share the same frozen pre-trained image and text encoders.

5.1. In-Distribution Performance on 11 Datasets

This section benchmarks the proposed method and its competitors on the in-distribution performance, *i.e.*, the training and test data are drawn from the (approximately) same distribution. Specifically, models are adapted on the training set of a dataset and then evaluated on the test set of the same dataset. Below are the results for ViT-B/32 while the results for ResNet50 [20] are given in Appendix C.2.

Learned prompts vs. hand-engineered prompts. A comparison of different text prompting methods on the performance averaged over 11 datasets for various perturbation budgets, ϵ , and shots is shown in Tab. 1. Our method yields substantial improvement over HEP. Even for 1 shot, our method (UC variant) effectively boosts the accuracy and robustness over HEP and the improvement is remarkable for $\epsilon = 1/255$, *i.e.*, +6.1% and +3.0% for accuracy and robustness respectively. Furthermore, such improvement consistently increases with the number of shots. We highlight that when the entire training dataset is used, our method (CSC

variant) achieves a substantial boost over HEP by +28.3% (+26.4%) and +25.0% (+16.7%) for accuracy and robustness respectively when $\epsilon = 1/255$ (4/255).

For each specific dataset, our method shows improvement on all of them but the margin varies considerably. Fig. 5 (Fig. 7 in Appendix) depicts the results for $\epsilon = 4/255$ (1/255). The improvement is huge on some datasets such as Flowers102, EuroSAT, *etc.*, but relatively small on ImageNet. The reason why the result of ImageNet is small is because the model weights have been pre-trained with HEP on the entire training set of ImageNet using TeCoA [45] so HEP is supposed to be optimal in this setting. It is, therefore, promising that our method in this setting can still improve on this by an evident margin of, *e.g.*, +1.1% and +1.9% for accuracy and robustness respectively (UC variant) when trained with 16 shots.

APT vs. AVP and PAFT. A comparison of different types of adaption methods on the overall performance averaged over 11 datasets is given in Tab. 1. It is observed that our method substantially outperforms AVP and PAFT in terms of both accuracy and robustness for 1 and 4 shots, suggesting that our method is more data-efficient than these alternatives. Noticeably, PAFT is much inferior to our method and even considerably underperforms the baseline HEP method in the 1-shot setting. Furthermore, as more data is used for training, *i.e.*, 16 and all shots, the superiority of our method compared to AVP in terms of both accuracy and robustness becomes more significant suggesting our method is much more effective than AVP in leveraging more data. Meanwhile, compared to PAFT when using 16 and all shots, our method achieves a comparable robustness and a considerably higher accuracy, suggesting a much better trade-off between accuracy and robustness.

For performance on each individual dataset as shown in Fig. 5 (and Fig. 7 in the Appendix), we highlight that our methods exhibit a substantial improvement over PAFT regarding both accuracy and robustness on ImageNet when trained with 100 shots. We observe that PAFT suffered from underfitting on ImageNet with 100 shots for both ϵ settings. We tried training with more epochs (increased to 50 from 20 epochs) but found no effect. This issue is even severer when a logistic classifier is applied as originally done for linear probing (*i.e.* non-adversarial variant of PAFT) in CLIP [51]. Note that the linear probing is also observed by Zhou et al. [72] to perform worse than zero-shot CLIP on ImageNet.

Unified context vs. class-specific context. Two variants of our method (Sec. 4.1) are compared. The UC variant of our method in general outperforms the CSC variant when the training data is limited, *i.e.*, 1 and 4 shots in Tab. 1, but underperforms when the training data is relatively abundant, *i.e.*, 16 and all shots. This is because the CSC variant has more parameters, and thus, larger capacity than the UC variant to learn from relatively larger-scale data. Nevertheless,

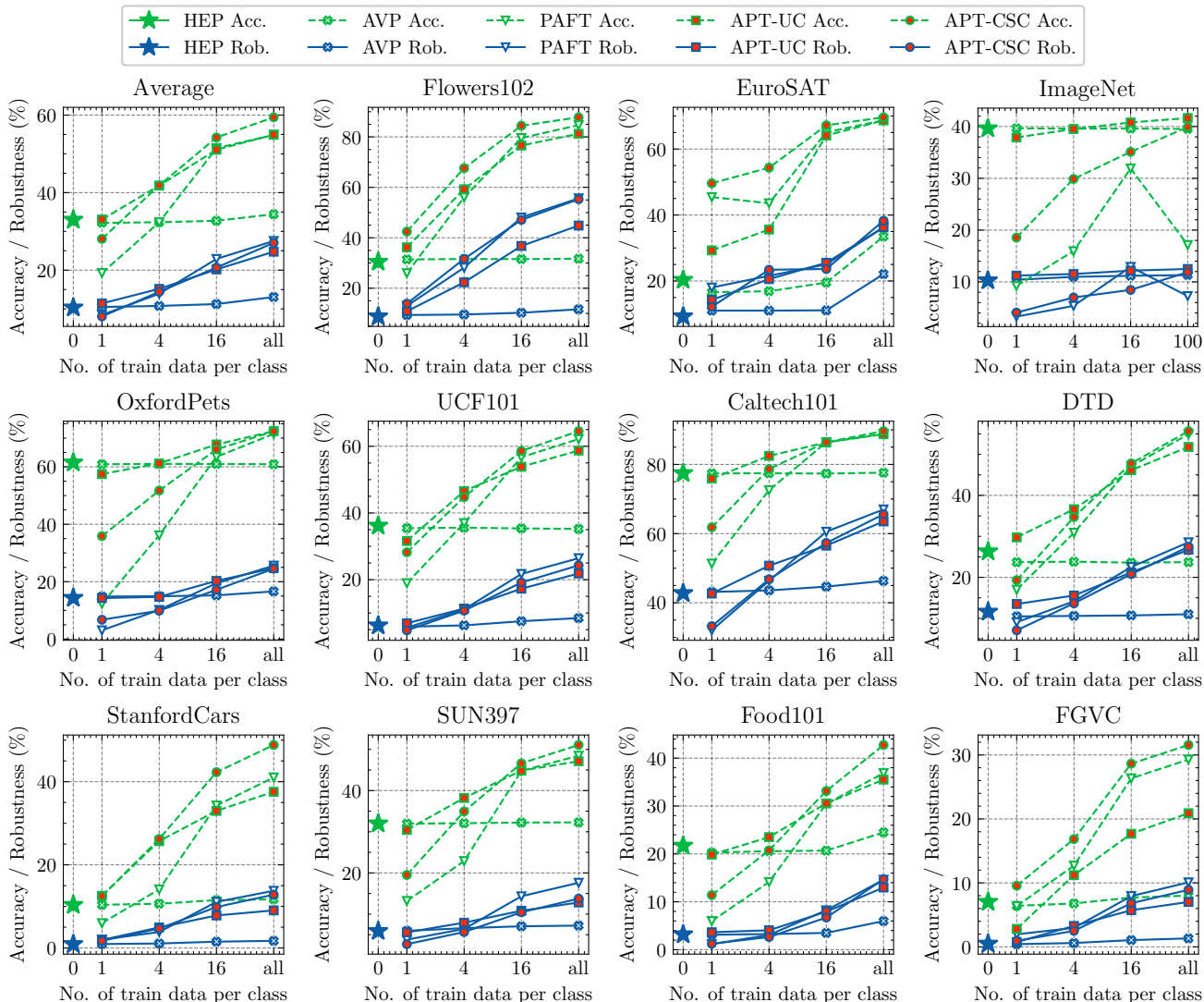


Figure 5. The in-distribution performance on 11 datasets and the averaged performance under different shots. $\epsilon = 4/255$ and $M = 16$.

the CSC variant also requires more data to mitigate overfitting due to the larger capacity. This likely accounts for the relatively poor performance of the CSC variant in 1- and 4-shot settings. The above trends hold for most of the evaluated datasets, but it is also observed that for some datasets one variant is consistently superior to the other, as shown in Fig. 5 (and Appendix Fig. 7). For instance, the CSC variant achieves higher (lower) performance than the UC variant across all four data schemes on Flowers102 (ImageNet).

5.2. Generalization of Learned Prompt Contexts

This section assesses the generalization ability beyond in-distribution performance of the text prompts learned by our method. Two types of generalization are evaluated: distribution shift and cross dataset. For both tests, we use ImageNet as the source dataset on which the model was adapted using the methods to be tested. We then evaluate the per-

formance of these ImageNet-adapted models on the target datasets with the same classes yet different data distributions (*i.e.* the distribution shift test) and the target datasets with different classes (*i.e.* the cross dataset test). Specifically, following the work of Li et al. [36], we use four ImageNet variant datasets, ImageNet-V2 [52], ImageNet-Sketch, ImageNet-R [23] and ObjectNet [4], to represent different kinds of distribution shift. We use the remaining ten of the original eleven datasets for the cross dataset test.

Our method exhibits the best overall generalization ability on both tests among all compared methods (Tab. 2). It achieves the highest accuracy and robustness on most target datasets. It is also noteworthy that PAFT, despite having the best robustness on the source dataset ImageNet, performs poorly under most distribution shifts, *e.g.*, its relative robustness improvement over ours (UC) drops from +0.57% on ImageNet to -2.99% on ImageNet-Sketch and -2.42% on

Table 2. The generalization of the prompts learned by our method on ImageNet to (1) datasets with input distribution shifts and (2) the other 10 datasets. “N/A” denotes that the corresponding method by design cannot generalize to the target setting. The results for both variants of our method are reported for the checkpoints trained with $M = 4$ and 16 shots. $\epsilon = 4/255$.

Method	Source		Distribution Shifts						Cross Datasets			
	ImageNet		ImageNet-V2		ImageNet-Sketch		ImageNet-R		ObjectNet		10 Datasets Avg.	
	Acc.	Rob.	Acc.	Rob.	Acc.	Rob.	Acc.	Rob.	Acc.	Rob.	Acc.	Rob.
HEP	39.86	10.28	32.74	7.49	17.40	7.21	21.46	5.80	9.16	1.15	32.32	10.34
AVP	39.61	11.18	32.68	8.12	17.39	7.69	21.47	6.25	9.08	1.31	31.37	11.13
PAFT	31.92	12.90	25.55	9.52	10.02	5.05	13.34	4.55	5.57	0.86	N/A	N/A
APT-CSC	37.18	9.49	28.93	6.65	12.72	4.83	15.06	3.57	7.17	0.61	N/A	N/A
APT-UC	40.80	<u>12.33</u>	33.20	<u>9.04</u>	18.35	8.04	22.66	6.97	9.31	1.49	<u>32.01</u>	11.84

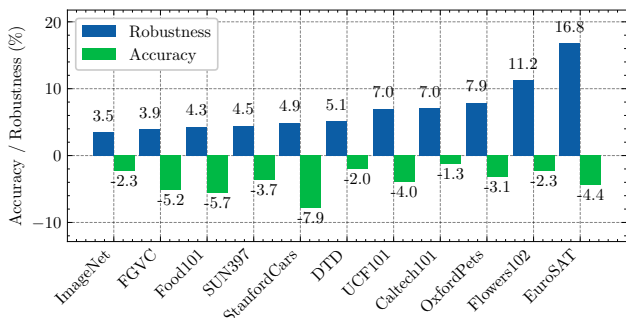


Figure 6. The performance improvement per dataset of our adversarially-trained prompt over the standardly-trained prompt for unified context ($M = 16$). The results are reported on the checkpoints trained on 16 shots. $\epsilon = 4/255$.

ImageNet-R. More importantly, PAFT, unlike AVP and our method, cannot deal with the novel classes that were unseen during training due to its rigid, hard-coded, linear layer. Hence, it is not applicable to the cross dataset test. The same issue also applies to the CSC variant of our method.

5.3. Trade-off Between Accuracy and Robustness

As shown in Fig. 6, we compare our adversarially-trained prompt contexts to the standardly-trained prompt contexts [72] based on the unified context. In general, the adversarially-trained prompts improve robustness at the expense of accuracy when compared to the standardly-trained prompts. This trade-off between accuracy and robustness is somewhat expected as it also happens to adversarially-trained vision models [60]. More importantly, we found that for most datasets the improvement in robustness surpasses the reduction in accuracy. Taking an example of Flowers102, a significant improvement of +11.2% in robustness is achieved with a sacrifice in accuracy of merely -2.3%. This observation suggests an attractive trade-off between accuracy and robustness of our method.

5.4. Reliability of Adversarial Evaluation

To verify that our evaluation of adversarial robustness is reliable, we first additionally evaluate the adversarial robustness of our methods using a diverse set of attacks including

TPGD [68], CW [8] and AutoAttack [12]. Next, to further exclude the possibility of our method masking the gradients for the particular prompts [2], we evaluate our methods using the adversarial examples transferred from other prompts with the same model as defined in Fig. 4. The results are described in Appendix C.4. In summary, the robustness advantage of our methods is not a consequence of overfitting to the particular attack or obfuscated gradients [2]

5.5. Ablation Study

To dissect the design choice of our method, we conduct ablation study on the length of prompt context, M , in Appendix C.3.1, the position of class embedding in Appendix C.3.2, and the prompting strategy for training adversarial generation in Appendix C.3.3.

6. Limitation

APT has two limitations. First, it is challenging to interpret the learned context vectors. The semantics of the learned context, when decoded by the nearest words, appears to be irrelevant to the data and sometimes even uninterpretable. Second, the effectiveness of APT depends on the pre-trained model weights. We observe that APT and other parameter-efficient adaption methods including AVP and PAFT cannot effectively boost adversarial robustness for the standardly-pre-trained overly vulnerable model. This is somewhat reasonable because the number of tunable parameters is dramatically limited compared to those of the image and text encoders esp. for the UC variant of our method. We discuss the above two issues in detail in Appendices C.5 and C.6.

7. Conclusion

This work studies the adversarial robustness of VLMs from the novel perspective of the text prompt. We first demonstrate that both adversarial attack and defense for VLMs are sensitive to the used text prompt. We then propose Adversarial Prompt Tuning (APT) to learn robust text prompts for CLIP based on adversarial examples to improve its adversarial robustness. Extensive experiments are conducted

to demonstrate the effectiveness of APT for both the in-distribution performance and the generalization ability under distribution shift and across datasets. APT is also parameter- and data-efficient. Given the promising performance of APT, our work paves a new way for enhancing adversarial robustness for VLMs through text prompting.

References

- [1] Maksym Andriushchenko and Nicolas Flammarion. Understanding and Improving Fast Adversarial Training. In *Advances in Neural Information Processing Systems*, page 12, 2020. [2](#)
- [2] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples. In *Proceedings of the 35th International Conference on Machine Learning*, pages 274–283. PMLR, 2018. ISSN: 2640-3498. [2](#), [8](#), [16](#)
- [3] Hyojin Bahng, Ali Jahani, Swami Sankaranarayanan, and Phillip Isola. Exploring Visual Prompts for Adapting Large-Scale Models, 2022. arXiv:2203.17274 [cs]. [3](#), [13](#)
- [4] Andrei Barbu, David Mayo, Julian Alverio, William Luo, Christopher Wang, Dan Gutfreund, Josh Tenenbaum, and Boris Katz. ObjectNet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2019. [7](#)
- [5] Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Muniyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. On the Opportunities and Risks of Foundation Models, 2022. arXiv:2108.07258 [cs]. [1](#)
- [6] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101—mining discriminative components with random forests. In *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part VI 13*, pages 446–461. Springer, 2014. [5](#)
- [7] Benjamin Bowman, Alessandro Achille, Luca Zancato, Matthew Trager, Pramuditha Perera, Giovanni Paolini, and Stefano Soatto. a-la-carte prompt tuning (apt): Combining distinct data via composable prompting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14984–14993, 2023. [2](#)
- [8] Nicholas Carlini and David Wagner. Towards Evaluating the Robustness of Neural Networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57, 2017. ISSN: 2375-1207. [8](#), [16](#)
- [9] Aochuan Chen, Peter Lorenz, Yuguang Yao, Pin-Yu Chen, and Sijia Liu. Visual Prompting for Adversarial Robustness. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, 2023. [2](#), [3](#), [6](#), [13](#), [17](#)
- [10] Tianlong Chen, Sijia Liu, Shiyu Chang, Yu Cheng, Lisa Amini, and Zhangyang Wang. Adversarial Robustness: From Self-Supervised Pre-Training to Fine-Tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 699–708, 2020. [2](#), [3](#), [6](#), [13](#)
- [11] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3606–3613, 2014. [5](#)
- [12] Francesco Croce and Matthias Hein. Reliable Evaluation of Adversarial Robustness with an Ensemble of Diverse Parameter-free Attacks. In *Proceedings of the 37th International Conference on Machine Learning*, page 11, 2020. [8](#), [16](#)
- [13] Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. RobustBench: a standardized adversarial robustness benchmark. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. [5](#), [16](#)
- [14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. [5](#)
- [15] Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, Jing Yi, Weilin Zhao, Xiaozhi Wang, Zhiyuan Liu, Hai-Tao Zheng, Jianfei Chen, Yang Liu, Jie Tang, Juanzi Li, and Maosong Sun. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence*, 5(3):220–235, 2023. Number: 3 Publisher: Nature Publishing Group. [2](#)
- [16] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Syl-

- vain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 5
- [17] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *2004 conference on computer vision and pattern recognition workshop*, pages 178–178. IEEE, 2004. 5
- [18] Zhe Gan, Yen-Chun Chen, Linjie Li, Chen Zhu, Yu Cheng, and Jingjing Liu. Large-Scale Adversarial Training for Vision-and-Language Representation Learning. In *Advances in Neural Information Processing Systems*, pages 6616–6628. Curran Associates, Inc., 2020. 5, 13
- [19] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and Harnessing Adversarial Examples. In *International Conference on Learning Representations*, 2015. 2
- [20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, Las Vegas, NV, USA, 2016. IEEE. 3, 6, 17
- [21] 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. 5
- [22] Dan Hendrycks, Kimin Lee, and Mantas Mazeika. Using Pre-Training Can Improve Model Robustness and Uncertainty. In *Proceedings of the 36th International Conference on Machine Learning*, pages 2712–2721. PMLR, 2019. ISSN: 2640-3498. 2, 3
- [23] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The Many Faces of Robustness: A Critical Analysis of Out-of-Distribution Generalization. In *International Conference on Computer Vision*, page 10, 2021. 7
- [24] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR, 2019. 2
- [25] Qidong Huang, Xiaoyi Dong, Dongdong Chen, Yinpeng Chen, Lu Yuan, Gang Hua, Weiming Zhang, and Nenghai Yu. Improving Adversarial Robustness of Masked Autoencoders via Test-time Frequency-domain Prompting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1600–1610, 2023. 2, 3
- [26] Nathan Inkawhich, Gwendolyn McDonald, and Ryan Luley. Adversarial Attacks on Foundational Vision Models, 2023. arXiv:2308.14597 [cs]. 1
- [27] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling Up Visual and Vision-Language Representation Learning With Noisy Text Supervision. In *Proceedings of the 38th International Conference on Machine Learning*, pages 4904–4916. PMLR, 2021. ISSN: 2640-3498. 1
- [28] Ziyu Jiang, Tianlong Chen, Ting Chen, and Zhangyang Wang. Robust Pre-Training by Adversarial Contrastive Learning. In *Advances in Neural Information Processing Systems*, pages 16199–16210. Curran Associates, Inc., 2020. 2, 3
- [29] Pau de Jorge, Adel Bibi, Riccardo Volpi, Amartya Sanyal, Philip Torr, Grégory Rogez, and Puneet K. Dokania. Make Some Noise: Reliable and Efficient Single-Step Adversarial Training. In *Advances in Neural Information Processing Systems*, 2022. 2
- [30] Minseon Kim, Jihoon Tack, and Sung Ju Hwang. Adversarial Self-Supervised Contrastive Learning. In *Advances in Neural Information Processing Systems*, pages 2983–2994. Curran Associates, Inc., 2020. 2, 3
- [31] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013. 5
- [32] Ananya Kumar, Aditi Raghunathan, Robbie Matthew Jones, Tengyu Ma, and Percy Liang. Fine-Tuning can Distort Pretrained Features and Underperform Out-of-Distribution. 2022. 1, 2
- [33] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation. In *Proceedings of the 39th International Conference on Machine Learning*, pages 12888–12900. PMLR, 2022. ISSN: 2640-3498. 1
- [34] Lin Li and Michael Spratling. Understanding and combating robust overfitting via input loss landscape analysis and regularization. *Pattern Recognition*, 136:109229, 2023. 2, 3
- [35] Lin Li and Michael W. Spratling. Data augmentation alone can improve adversarial training. In *The Eleventh International Conference on Learning Representations*, 2023. 3
- [36] Lin Li, Yifei Wang, Chawin Sitawarin, and Michael Spratling. Oodrobustbench: benchmarking and analyzing adversarial robustness under distribution shift. *arXiv preprint arXiv:2310.12793*, 2023. 7
- [37] Junfan Lin, Jianlong Chang, Lingbo Liu, Guanbin Li, Liang Lin, Qi Tian, and Chang-wen Chen. Being comes from not-being: Open-vocabulary text-to-motion generation with wordless training. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 23222–23231, 2023. 2
- [38] Zhaojiang Lin, Andrea Madotto, and Pascale Fung. Exploring versatile generative language model via parameter-efficient transfer learning. *arXiv preprint arXiv:2004.03829*, 2020. 2
- [39] 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, 2021. arXiv:2107.13586 [cs]. 2
- [40] 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. 2
- [41] Rundong Luo, Yifei Wang, and Yisen Wang. Rethinking the Effect of Data Augmentation in Adversarial Contrastive

- Learning. In *The Eleventh International Conference on Learning Representations*, 2023. 2, 3
- [42] Chengcheng Ma, Yang Liu, Jiankang Deng, Lingxi Xie, Weiming Dong, and Changsheng Xu. Understanding and mitigating overfitting in prompt tuning for vision-language models. *IEEE Transactions on Circuits and Systems for Video Technology*, 2023. 2
- [43] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards Deep Learning Models Resistant to Adversarial Attacks. In *International Conference on Learning Representations*, 2018. 1, 2, 3, 4, 5
- [44] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 5
- [45] Chengzhi Mao, Scott Geng, Junfeng Yang, Xin Wang, and Carl Vondrick. Understanding Zero-shot Adversarial Robustness for Large-Scale Models. In *The Eleventh International Conference on Learning Representations*, 2023. 1, 3, 4, 5, 6, 13, 16
- [46] 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. 5
- [47] Jihye Park, Sunwoo Kim, Soohyun Kim, Seokju Cho, Jaeeun Yoo, Youngjung Uh, and Seungryong Kim. Lanit: Language-driven image-to-image translation for unlabeled data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23401–23411, 2023. 2
- [48] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3498–3505. IEEE, 2012. 5
- [49] Fang Peng, Xiaoshan Yang, Linhui Xiao, Yaowei Wang, and Changsheng Xu. Sgva-clip: Semantic-guided visual adapting of vision-language models for few-shot image classification. *IEEE Transactions on Multimedia*, 2023. 2
- [50] Jianing Qiu, Lin Li, Jiankai Sun, Jiachuan Peng, Peilun Shi, Ruiyang Zhang, Yinzhaohong, Kyle Lam, Frank P-W Lo, Bo Xiao, et al. Large ai models in health informatics: Applications, challenges, and the future. *IEEE Journal of Biomedical and Health Informatics*, 2023. 1
- [51] 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 *Proceedings of the 38th International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. ISSN: 2640-3498. 1, 2, 3, 6, 13
- [52] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do ImageNet Classifiers Generalize to ImageNet? In *Proceedings of the 36th International Conference on Machine Learning*, page 12, 2019. 7
- [53] Leslie Rice, Eric Wong, and J Zico Kolter. Overfitting in adversarially robust deep learning. In *Proceedings of the 37th International Conference on Machine Learning*, page 12, 2020. 2
- [54] Christian Schlarman and Matthias Hein. On the Adversarial Robustness of Multi-Modal Foundation Models, 2023. arXiv:2308.10741 [cs]. 1
- [55] Ali Shafahi, Mahyar Najibi, Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S. Davis, Gavin Taylor, and Tom Goldstein. Adversarial Training for Free! In *Advances in Neural Information Processing Systems*, 2019. 2
- [56] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 5
- [57] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014. 1
- [58] Nima Tajbakhsh, Jae Y. Shin, Suryakanth R. Gurudu, R. Todd Hurst, Christopher B. Kendall, Michael B. Gotway, and Jianming Liang. Convolutional neural networks for medical image analysis: Full training or fine tuning? *IEEE Transactions on Medical Imaging*, 35(5):1299–1312, 2016. 2
- [59] Ming Tao, Bing-Kun Bao, Hao Tang, and Changsheng Xu. Galip: Generative adversarial clips for text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14214–14223, 2023. 2
- [60] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness May Be at Odds with Accuracy. In *International Conference on Learning Representations*, 2019. 8
- [61] Zekai Wang, Tianyu Pang, Chao Du, Min Lin, Weiwei Liu, and Shuicheng Yan. Better Diffusion Models Further Improve Adversarial Training, 2023. arXiv:2302.04638 [cs]. 3
- [62] Eric Wong, Leslie Rice, and J. Zico Kolter. Fast is better than free: Revisiting adversarial training. In *International Conference on Learning Representations*, 2020. 2
- [63] Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial Weight Perturbation Helps Robust Generalization. In *Advances in Neural Information Processing Systems*, pages 2958–2969, 2020. 3
- [64] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 3485–3492. IEEE, 2010. 5
- [65] Karren Yang, Wan-Yi Lin, Manash Barman, Filipe Condessa, and Zico Kolter. Defending Multimodal Fusion Models Against Single-Source Adversaries. pages 3340–3349, 2021. 13
- [66] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems*, 2014. 1
- [67] Shengming Yu, Zhaopeng Dou, and Shengjin Wang. Prompting and tuning: A two-stage unsupervised domain adaptive

- person re-identification method on vision transformer backbone. *Tsinghua Science and Technology*, 28(4):799–810, 2023. [2](#)
- [68] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically Principled Trade-off between Robustness and Accuracy. In *International Conference on Machine Learning*, pages 7472–7482. PMLR, 2019. ISSN: 2640-3498. [3](#), [5](#), [8](#), [16](#)
- [69] Yuanhan Zhang, Kaiyang Zhou, and Ziwei Liu. Neural prompt search. *arXiv preprint arXiv:2206.04673*, 2022. [2](#)
- [70] Yunqing Zhao, Tianyu Pang, Chao Du, Xiao Yang, Congxuan Li, Ngai-Man Cheung, and Min Lin. On Evaluating Adversarial Robustness of Large Vision-Language Models, 2023. arXiv:2305.16934 [cs]. [1](#), [3](#), [4](#), [6](#)
- [71] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional Prompt Learning for Vision-Language Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16816–16825, 2022. [3](#)
- [72] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to Prompt for Vision-Language Models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022. [2](#), [3](#), [4](#), [5](#), [6](#), [8](#), [13](#), [16](#)
- [73] Ziqin Zhou, Yinjie Lei, Bowen Zhang, Lingqiao Liu, and Yifan Liu. Zegclip: Towards adapting clip for zero-shot semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11175–11185, 2023. [2](#)

A. Related works

A.1. Adversarial Defense for VLMs

Research on adversarial defense has concentrated on vision models, and few previous works consider VLMs [18, 45, 65]. Gan et al. [18] first employ adversarial training to train VLMs from scratch for improved clean performance. Yang et al. [65] equip standard VLMs with an auxiliary network and a robust feature fusion layer and performs adversarial training on these new modules. Mao et al. [45] uses full adversarial fine-tuning to adapt a pre-trained CLIP model for zero-shot adversarial robustness. All of the above works study the adversarial robustness of VLMs through the lens of model weights, which leaves unexplored the question of how the text prompt affect adversarial robustness. To fill this blank, we analyze the sensitivity of adversarial robustness to text prompts and propose a new method to optimize the text prompt to increase adversarial robustness.

A.2. Adversarial Visual Prompting

AVP [9] combines visual prompting [3] with adversarial training to counteract adversarial perturbations. The visual prompt perturbation, parameterized by ϕ , is applied to the input image, x , so that the prompted image is given by $x_{vp} = x + \phi$. AVP optimizes visual prompt ϕ to jointly minimize both clean and adversarial losses:

$$\arg \min_{\phi} \mathcal{L}(x + \phi, t, y; \theta_v, \theta_t) + \mathcal{L}(x + \delta + \phi, t, y; \theta_v, \theta_t) \quad (12)$$

where δ is generated by Algorithm 1 independent of ϕ .

A.3. Partial Adversarial Fine-Tuning

PAFT [10] discards the text encoder branch of CLIP and attaches an extra linear layer, parameterized by θ_l , on top of the frozen image encoder, θ_v , to form a new classifier. The output of the linear classifier has the same dimension as the number of classes. PAFT optimizes θ_l to minimize the adversarial loss:

$$\arg \min_{\theta_l} \mathcal{L}(x + \delta, y; \theta_v, \theta_l) \quad (13)$$

where δ is generated using the conventional PGD attack on the new classifier.

B. Experiment Setting

Data. The 11 datasets were selected to constitute a comprehensive benchmark, encompassing a diverse array of vision tasks including generic objects classification, scene recognition, action classification, fine-grained classification, textures recognition and satellite imagery analysis. They were split into training and test sets in the same way as Zhou et al. [72]. The N -shot images, once sampled, are fixed so that all

Table 3. Datasets statistics and their prompts

Dataset	Classes	Hand-Engineered Prompt
ImageNet	1000	a photo of a [CLASS]
Caltech101	100	a photo of a [CLASS]
OxfordPets	37	a photo of a [CLASS], a type of pet
StanfordCars	196	a photo of a [CLASS]
Flowers102	102	a photo of a [CLASS], a type of flower
Food101	101	a photo of a [CLASS], a type of food
FGVCAircraft	100	a photo of a [CLASS], a type of aircraft
SUN397	397	a photo of a [CLASS]
DTD	47	[CLASS] texture
EuroSAT	10	a centered satellite photo of a [CLASS]
UCF101	101	a photo of a person doing [CLASS]
ImageNetV2	1000	a photo of a [CLASS]
ImageNet-Sketch	1000	a photo of a [CLASS]
ImageNet-A	200	a photo of a [CLASS]
ImageNet-R	200	a photo of a [CLASS]

evaluated methods are trained on the exactly same data for a fair comparison.

Model. The text encoder is the default pre-trained model from CLIP [51]. For both image and text branches, we adopt the same data pre-processing as CLIP [51].

B.1. General Training Set-ups

APT was trained by Stochastic Gradient Descent (SGD) using a cosine learning rate schedule with an initial learning rate of 0.002. The batch size was 32. For ImageNet, the number of epochs was 20, 20, 50 and 20 for 1, 4, 16 and all shots respectively, and for other datasets was 50, 100, 200, 200. The number of epochs used with ImageNet was reduced due to limited computational resources. Results were reported for the last checkpoint.

AVP was implemented in the mode of padding with a prompt size of 52 to match the number of parameters of our method for the unified context. Although a class-specific variant of AVP was proposed in [9] and was observed to outperform its unified variant on CIFAR10 with ResNet18, we found that the class-specific variant of AVP generalized poorly to our experimental set-ups, *i.e.*, CLIP plus 11 diverse datasets. We therefore decided to use the unified variant of AVP. Following [9], a cosine learning rate schedule with an initial learning rate of 0.1 was used. The number of epochs, corresponding to 1/4/16/all shots, was 20/50/100/100 for non-ImageNet datasets and 20/20/50/20 for ImageNet. The implementation was based on the open-source code of Chen et al. [9].

PAFT was trained, following Chen et al. [10], for 20/50/100/100 epochs with an initial learning rate of 0.1 decayed by 0.1 at epochs 5/15/30/30 and 10/25/50/50 for 1/4/16/all shots, respectively, for non-ImageNet datasets. For ImageNet, the number of training epochs was 20/20/50/20 for 1/4/16/100 shots respectively.

Table 4. The average performance for ResNet50 with $\epsilon = 1/255$. $M = 16$ and 16 shots.

Method	Accuracy	Robustness
HEP	37.44	22.35
AVP	37.62	25.04
PAFT	50.14	39.14
APT-UC	58.38	38.32
APT-CSC	60.21	39.26

Table 5. The average performance of our methods with different number of context vectors, M . $\epsilon = 4/255$ and the number of shots is 16.

M	UC		CSC	
	Acc.	Rob.	Acc.	Rob.
1	43.63	16.75	53.77	20.77
4	48.09	18.59	54.07	20.53
8	50.19	19.65	54.21	20.52
16	51.06	20.19	54.21	20.68

C. Additional Results

C.1. In-Distribution Performance for $\epsilon = 1/255$

Fig. 7 depicts the results for $\epsilon = 1/255$ on each dataset. The trends are generally consistent to those observed for $\epsilon = 4/255$ in Fig. 5.

C.2. Generalization to Other Architectures

As shown in Tab. 4, the results of our method with the ResNet50 vision encoder demonstrates a performance advantage consistent with that achieved based on ViT. Our method exhibits substantial enhancements over the zero-shot baseline. Furthermore, our approach attains robustness comparable to that achieved by LP, while achieving significantly higher accuracy.

C.3. Ablation Study

The ablation study was conducted using the following setup unless otherwise specified. The number of context vectors, M , was 16, the class embedding was placed at the *end* of the prompt, the prompting strategy for training the adversary was *on-the-fly*. Models were trained with 16 shots for $\epsilon = 4/255$.

C.3.1 Prompt Context Length

As shown in Tab. 5, it is notable that as the number of context vectors increases, a significant enhancement in performance becomes evident for variant UC. In the case of CSC, while there is a marginal increase in accuracy associated

Table 6. The average performance of our methods with the class embedding located at the different positions. $M = 16$, $\epsilon = 4/255$ and 16 shots.

Class Embedding Position	UC		CSC	
	Acc.	Rob.	Acc.	Rob.
front	51.16	20.04	54.09	20.56
middle	51.91	20.16	54.21	20.68
end	51.06	20.19	54.21	20.68

Table 7. The average performance of our methods using different prompting strategies for generating training adversarial examples. $M = 16$, $\epsilon = 4/255$ and the number of shots is 16.

t'	α'	UC		CSC	
		Acc.	Rob.	Acc.	Rob.
<i>constant</i>	-	44.42	12.30	43.52	11.13
<i>on-the-fly</i>	-	51.06	<u>20.19</u>	54.21	<u>20.68</u>
<i>perturbed</i>	0.1	53.31	17.99	55.10	19.42
<i>perturbed</i>	0.01	<u>52.01</u>	19.99	<u>54.25</u>	20.66
<i>perturbed</i>	0.001	51.60	20.25	54.24	20.69

with the number of context vectors, there is no improvement in robustness. A longer context has a greater number of parameters and, consequently, it is expected to possess a larger capacity for improved performance. However, the observation that robustness is not notably improved for CSC may be attributed to a fact that even when the context length is one (the minimum), CSC inherently incorporates more parameters than UC with a context length of 16, provided that the number of classes exceeds 16. This phenomenon exists in the majority of our test datasets. The default context length was set to 16 based on these results, due to its better performance.

C.3.2 Class Embedding Position

The performance of our method was evaluated with the class embedded at different locations in the context (Tab. 6). This experiment reveals that no statistically significant differences are observed for both UC and CSC. As a result, *end* was selected as the default position due to its slightly better performance and simplicity.

C.3.3 Prompting for Training Adversarial Generation

Tab. 7 compares the performance of different variants of our method when training adversarial examples are generated by the different prompting strategies introduced in Sec. 4.2. We observe that the prompting strategies *on-the-fly* and *perturbed*, if α' is properly configured, achieve very close ac-

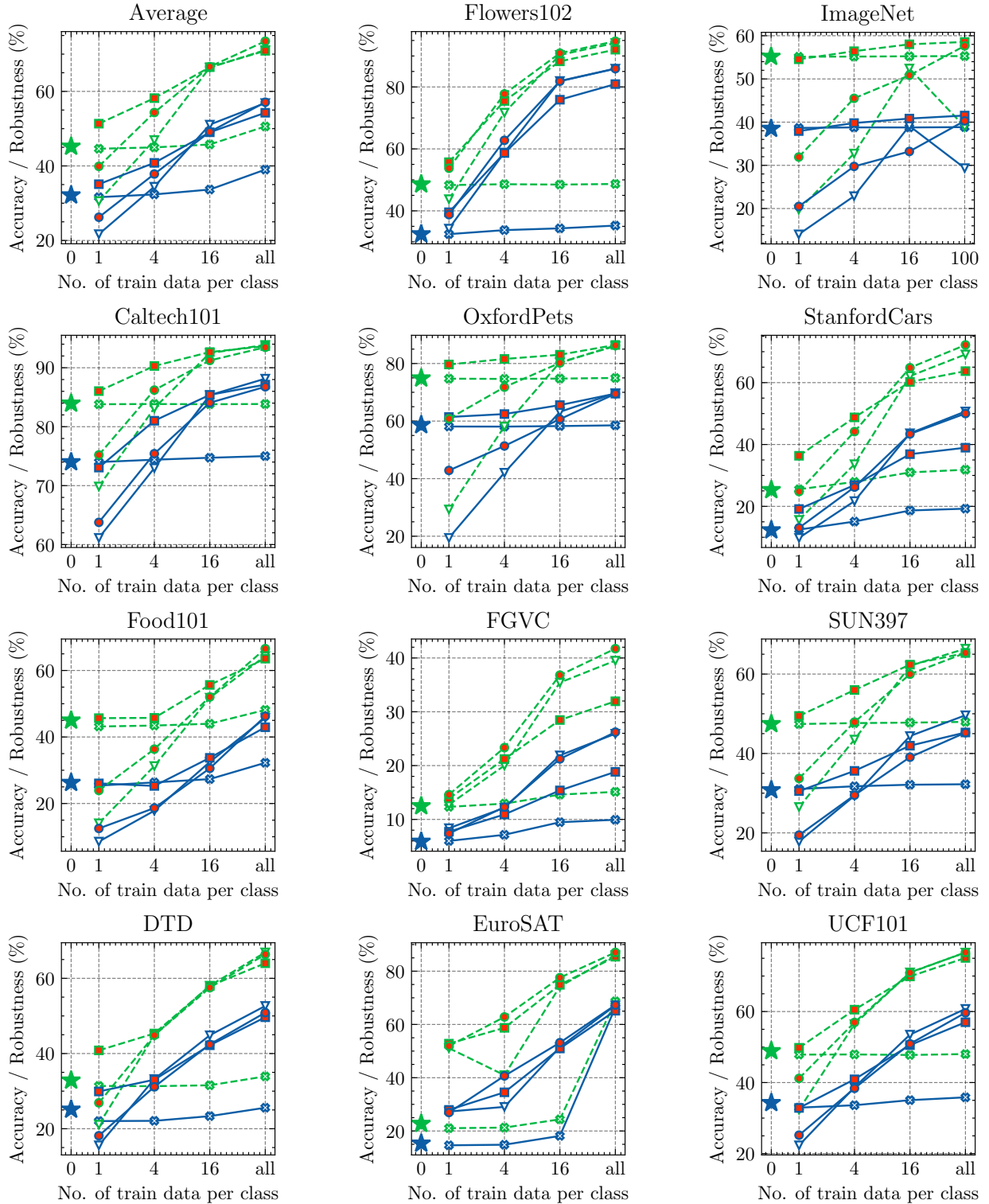


Figure 7. The in-distribution performance on 11 datasets and the averaged performance under different shots. $\epsilon = 1/255$ and $M = 16$.

Table 8. The average robustness under various attacks. The results of our methods are reported on the checkpoints trained on all shots. $M = 16$ and $\epsilon = 4/255$.

Method	Accuracy	PGD	TPGD	CW	AutoAttack
HEP	32.98	10.34	23.34	10.02	8.55
APT-UC	54.95	24.76	39.40	22.17	19.69
APT-CSC	59.43	27.05	42.80	24.69	22.47

Table 9. The average robustness of our methods against attacks based on the different attack prompts (columns). The **lowest robustness** is highlighted for each variant of our method (row). ‘‘Learned’’ denotes the prompts learned by our method. P1-P6 are defined in Fig. 4. The results of our methods are reported for the checkpoints trained on all shots. $M = 16$ and $\epsilon = 4/255$.

Method	Accuracy	Robustness						
		Learned	P1	P2	P3	P4	P5	P6
APT-UC	54.95	24.76	35.78	35.30	35.37	35.26	37.92	36.58
APT-CSC	59.43	27.05	42.43	42.09	42.10	42.04	44.17	43.35

curacy and robustness, while *perturbed* seems to be slightly advantageous regarding accuracy under the unified context. Both of them perform much better than the strategy *constant*. Inside the strategy *perturbed*, the perturbation magnitude increases as α' increases so that the perturbed prompts deviate more from the original ones, which results in the drop of performance as shown in the data.

C.4. Verification

To verify that our evaluation of adversarial robustness is reliable, we first additionally evaluated the adversarial robustness of our methods using a diverse set of attacks including TPGD [68], CW [8] and AutoAttack [12]. AutoAttack, particularly, is widely accepted as a reliable attack. Note that for AutoAttack only we reduce the size of the ImageNet test set to 5000 following the practice used in RobustBench [13], since running AutoAttack on the full ImageNet test set is very expensive. As shown in Tab. 8, the performance of our methods degrade by a reasonable amount but still improve over the baseline method by a large margin under the stronger attacks CW and AutoAttack. This suggests that the robustness advantage of our methods is not a consequence of overfitting to the particular attack or to obfuscated gradients [2].

To further exclude the possibility of our method masking the gradients for the particular attack prompts [2], we evaluated our methods using adversarial examples transferred from other prompts with the same model as defined in Fig. 4. It is shown in Tab. 9 that our method achieves higher robustness against the adversarial examples generated by six other different prompts. This suggests that the robustness achieved by our method is not specific to the at-

Table 10. The nearest word for each unified context vector learned by our method with 16 shots on different datasets. N/A means non-Latin characters.

No.	ImageNet		Food101		DTD	
	Word	Distance	Word	Distance	Word	Distance
1	optimizing	1.30	yourselves	1.18	vid	0.82
2	dent	1.63	aux	0.91	N/A	0.96
3	daisies	1.07	ester	0.94	industries	0.67
4	hashtags	1.06	blocker	0.82	lala	0.98
5	ents	1.01	obe	1.26	tice	0.97
6	sph	1.14	each	0.94	met	1.02
7	swirl	1.04	legally	1.24	stan	1.27
8	ridge	1.20	frock	1.21	washingtondc	1.33
9	respectively	1.10	notte	0.92	press	1.31
10	stigma	1.02	ur	0.91	ely	1.52
11	spike	1.09	offic	1.17	Ł	1.18
12	pas	0.83	councillor	1.17	soil	0.85
13	conquering	1.40	dgers	1.17	popefrancis	2.21
14	turk	1.19	charging	1.35	honourable	1.49
15	compares	1.21	sk	1.09	perth	1.50
16	artillery	1.39	mindy	1.38	methodology	1.76

tack from a particular prompting source and can generalize well to attacks using different prompting methods.

C.5. Interpretability of the Learned Context

This section studies the interpretability of the context vectors learned by APT through the lens of the nearest words. The nearest word is, following the implementation of Zhou et al. [72], the vocabulary whose embedding vector is closest to the learned vector based on the Euclidean distance. Tab. 10 shows the decoded nearest words for sixteen learned, unified, vectors ($M = 16$) on the selected datasets. We find that, in contrast to the hand-engineered prompt contexts (Tab. 3), the decoded nearest words appear to lack semantic relevance with the respective dataset. It seems that APT may have learned something beyond current human vocabulary in order to defend against human-imperceptible adversarial perturbations. On the other hand, the nearest words, despite being used in the previous works like [72], may not accurately reflect the semantic meaning of the learned vectors. To be more specific, there is a lack of well-established distance threshold above which the decoded words can be considered as faithful interpretation. Besides, some decoded words are non-Latin characters, *e.g.*, the second word for DTD in Tab. 10, which are uninterpretable to human.

C.6. Dependency on the Robust Visual Backbone

This section discusses the dependency of APT on the pre-trained adversarially robust image encoder. It was observed in our experiments that APT failed to improve robustness over the HEP baseline, or even collapsed during training, when the image encoder of CLIP is not pre-trained using a robust training methods like TeCoA [45] to attain adversarial robustness. This suggests that a robust visual backbone

is necessary for APT to be effective.

We note that it was reported in Chen et al. [9] that AVP achieved 27.7% for accuracy and 16.7% for robustness against PGD100 with a standardly-trained ResNet18 [20] on CIFAR10. The accuracy is dramatically compromised when compared to 94.9% accuracy of the pre-trained model without AVP. Although the robustness increases from 0 to 16.7%, the trade-off between accuracy and robustness seems unacceptable.