Visual Prompting for Adversarial Robustness

Aochuan Chen*
Michigan State University
chenaoch@msu.edu

Peter Lorenz*
Fraunhofer ITWM†
peter.lorenz@itwm.fhg.de

Yuguang Yao
Michigan State University
yaoyugua@msu.edu

Pin-Yu Chen
IBM Research
pin-yu.chen@ibm.com

Sijia Liu
Michigan State University
liusiji5@msu.edu

Abstract

In this work, we leverage visual prompting (VP) to improve adversarial robustness of a fixed, pre-trained model at testing time. Compared to conventional adversarial defenses, VP allows us to design universal (*i.e.*, data-agnostic) input prompting templates, which have plug-and-play capabilities at testing time to achieve desired model performance without introducing much computation overhead. Although VP has been successfully applied to improving model generalization, it remains elusive whether and how it can be used to defend against adversarial attacks. We investigate this problem and show that the vanilla VP approach is *not* effective in adversarial defense since a universal input prompt lacks the capacity for robust learning against sample-specific adversarial perturbations. To circumvent it, we propose a new VP method, termed Class-wise Adversarial Visual Prompting (C-AVP), to generate class-wise visual prompts so as to not only leverage the strengths of ensemble prompts but also optimize their interrelations to improve model robustness. Our experiments show that C-AVP outperforms the conventional VP method, with 2.1× standard accuracy gain and 2× robust accuracy gain. Compared to classical test-time defenses, C-AVP also yields a 42× inference time speedup. Code is available at `link`.

1 Introduction

Machine learning (ML) models,can easily be manipulated to output drastically different classifications. This process is known as *adversarial attack* [1, 2]. Thereby, model robustification against adversarial attacks is now a major focus of research. Yet, a large volume of existing works focused on advancing training recipes and/or model architectures to gain robustness. Adversarial training (AT) [3], one of the most effective defense, adopted min-max optimization to minimize the worst-case training loss induced by adversarial attacks. Extended from AT, various defense methods were proposed, ranging from supervised learning to semi-supervised learning, and further to unsupervised learning [4–12].

Although the design for robust training has made tremendous success in improving model robustness [13, 14], it typically takes an intensive computation cost with poor defense scalability to a fixed, pre-trained ML model. Towards circumventing this difficulty, the problem of test-time defense arises; see the seminal work in [15]. Test-time defense alters either a test-time input example or a small portion of the pre-trained model for adversarial defense. Examples include input (anti-adversarial) purification [16–18] and model refinement by augmenting the pre-trained model with auxiliary components [19–21]. However, these defense techniques inevitably raise the inference time and hamper the test-time efficiency [15]. Inspired by that, our work will advance the test-time defense technology by leveraging the idea of *visual prompting* (**VP**) [22], also known as model reprogramming [23–26].

Generally speaking, VP [22] creates a *universal* (*i.e.*, *data-agnostic*) input prompting template (in terms of input perturbations) in order to improve the generalization ability of a pre-trained model

*Equal contribution. † and Fraunhofer Center of Machine Learning

ML Safety Workshop, 36th Conference on Neural Information Processing Systems (NeurIPS 2022).

when incorporating such a visual prompt into test-time examples. It enjoys the same idea as model reprogramming [23–26] or unadversarial example [27], which optimizes the universal perturbation pattern to maneuver (*i.e.*, reprogram) the functionality of a pre-trained model towards the desired criterion, *e.g.*, cross-domain transfer learning [25], out-of-distribution generalization [27], and fairness [26]. However, it remains elusive whether or not VP could be designed as an effective solution to adversarial defense. We will investigate this problem, which we call *adversarial visual prompting* (**AVP**), in this work. Compared to conventional test-time defense methods, AVP will significantly reduce the inference time overhead since visual prompts can be designed offline over training data and have the plug-and-play capability applied to any testing data. We summarize our **contributions** below.

❶ We formulate and investigate the problem of AVP for the first time. We empirically show that the conventional data-agnostic VP design is incapable of gaining adversarial robustness.

❷ We propose a new VP method, termed class-wise AVP (**C-AVP**), which produces multiple, class-wise visual prompts with explicit optimization on their couplings to gain adversarial robustness.

❸ We provide insightful experiments to demonstrate the pros and cons of VP in adversarial defense.

2 Problem Statement

Visual prompting. We describe the problem setup of VP following [22, 24–26]. Let $\mathcal{D}_{\mathrm{tr}}$ denote a training set for supervised learning, where $(\mathbf{x}, y) \in \mathcal{D}_{\mathrm{tr}}$ signifies a training sample with feature $\mathbf{x}$ and label y. And let $\boldsymbol{\delta}$ be a visual prompt. The prompted input is then given by $\mathbf{x} + \boldsymbol{\delta}$ with respect to (w.r.t.) $\mathbf{x}$. VP drives $\boldsymbol{\delta}$ to minimize the performance loss ℓ of a pre-trained model $\boldsymbol{\theta}$. This leads to

$$\begin{array}{ll} \underset{\boldsymbol{\delta}}{\operatorname{minimize}} & \mathbb{E}_{(\mathbf{x},y)\in\mathcal{D}_{\mathrm{tr}}}[\ell(\mathbf{x}+\boldsymbol{\delta}; y, \boldsymbol{\theta})] \\ \text{subject to} & \boldsymbol{\delta} \in \mathcal{C}, \end{array} \tag{1}$$

where ℓ denotes prediction error given the training data $(\mathbf{x}, y)$ and base model $\boldsymbol{\theta}$, and $\mathcal{C}$ is a perturbation constraint. Following [22, 24, 25], $\mathcal{C}$ restricts $\boldsymbol{\delta}$ to let $\mathbf{x} + \boldsymbol{\delta} \in [0, 1]$ for any $\mathbf{x}$. Projected gradient descent (PGD) [3, 27] can then be applied to solving problem (1). In the evaluation, $\boldsymbol{\delta}$ is integrated into test data to improve the prediction ability of $\boldsymbol{\theta}$.

Adversarial visual prompting. Inspired by the usefulness of VP to improve model generalization [25, 22], we ask:

(AVP problem) Can VP (1) be extended to robustify $\boldsymbol{\theta}$ against adversarial attacks?

At the first glance, the AVP problem seems trivial only if we specify the performance loss ℓ as the adversarial training (AT) loss [3, 4]:

$$\ell_{\mathrm{adv}}(\mathbf{x}+\boldsymbol{\delta}; y, \boldsymbol{\theta}) = \underset{\mathbf{x}':\|\mathbf{x}'-\mathbf{x}\|_\infty \le \epsilon}{\operatorname{maximize}} \ \ell(\mathbf{x}'+\boldsymbol{\delta}; y, \boldsymbol{\theta}), \tag{2}$$

where $\mathbf{x}'$ denotes the adversarial input that lies in the ℓ_∞-norm ball centered at $\mathbf{x}$ with radius $\epsilon > 0$.

Recall from (1) that the conventional VP design requests $\boldsymbol{\delta}$ to be universal across training data. Thus, we term *universal AVP* (**U-AVP**) the following problem by integrating (1) with (2):

$$\underset{\boldsymbol{\delta}:\,\boldsymbol{\delta}\in\mathcal{C}}{\operatorname{minimize}} \quad \lambda \mathbb{E}_{(\mathbf{x},y)\in\mathcal{D}_{\mathrm{tr}}}[\ell(\mathbf{x}+\boldsymbol{\delta}; y, \boldsymbol{\theta})] + \mathbb{E}_{(\mathbf{x},y)\in\mathcal{D}_{\mathrm{tr}}}[\ell_{\mathrm{adv}}(\mathbf{x}+\boldsymbol{\delta}; y, \boldsymbol{\theta})] \tag{U-AVP}$$

where $\lambda > 0$ is a regularization parameter to strike a balance between generalization and adversarial robustness [4].

The problem (U-AVP) can be effectively solved using a standard min-max optimization method, which involves two alternating optimization routines: inner maximization and outer minimization. The former generates adversarial examples as AT, and the latter produces the visual prompt $\boldsymbol{\delta}$ like (1). At testing time, the effectiveness of $\boldsymbol{\delta}$ is measured from two aspects: (1) standard accuracy, *i.e.*, the accuracy of $\boldsymbol{\delta}$-integrated benign examples, and (2) robust accuracy, *i.e.*, the accuracy of $\boldsymbol{\delta}$-integrated adversarial examples (against the victim model $\boldsymbol{\theta}$). Despite the succinctness of (U-AVP), Fig. 1 shows its *ineffectiveness* to defend against adversarial attacks. Compared to the vanilla VP (1), it also suffers a significant standard accuracy drop (over 50% in Fig. 1 corresponding to 0 PGD attack steps) and robust accuracy is only enhanced by a small margin (around 18% against PGD attacks). The negative results in Fig. 1 are not quite surprising since a data-agnostic input prompt $\boldsymbol{\delta}$ has limited learning capacity to enable adversarial defense. Thus, it is non-trivial to tackle the problem of AVP.

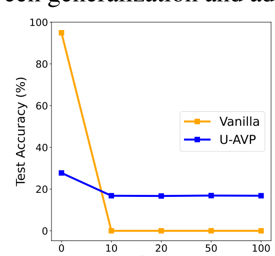


Fig. 1: Example of designing U-AVP for adversarial defense on (CIFAR-10, ResNet18), measured by robust accuracy against PGD attacks [3] of different steps. The robust accuracy of 0 steps is the standard accuracy.

3 Class-wise Adversarial Visual Prompting

No free lunch for class-wise visual prompts. A direct extension of (U-AVP) is to introduce multiple adversarial visual prompts, each of which corresponds to one class in the training set $\mathcal{D}_{\mathrm{tr}}$. If we split $\mathcal{D}_{\mathrm{tr}}$ into class-wise training sets $\{\mathcal{D}_{\mathrm{tr}}^{(i)}\}_{i=1}^{N}$ (for N classes) and introduce class-wise visual prompts $\{\boldsymbol{\delta}^{(i)}\}$, then the direct C-AVP extension from (U-AVP) becomes

$$\underset{\{\boldsymbol{\delta}^{(i)}\in\mathcal{C}\}_{i\in[N]}}{\mathrm{minimize}} \quad \frac{1}{N}\sum_{i=1}^{N}\left\{\lambda\mathbb{E}_{(\mathbf{x},y)\in\mathcal{D}_{\mathrm{tr}}^{(i)}}[\ell(\mathbf{x}+\boldsymbol{\delta}^{(i)};y,\boldsymbol{\theta})]+\mathbb{E}_{(\mathbf{x},y)\in\mathcal{D}_{\mathrm{tr}}^{(i)}}[\ell_{\mathrm{adv}}(\mathbf{x}+\boldsymbol{\delta}^{(i)};y,\boldsymbol{\theta})]\right\} \quad \text{(C-AVP-v0)}$$

where $[N]$ denotes the set of class labels $\{1,2,\ldots,N\}$. It is worth noting that C-AVP-v0 is *decomposed* over class labels. Although the class-wise separability facilitates numerical optimization, it introduces challenges **(C1)**-**(C2)** when applying class-wise visual prompts for adversarial defense.

• **(C1)** *Test-time prompt selection*: After acquiring the visual prompts $\{\boldsymbol{\delta}^{(i)}\}$ from (C-AVP-v0), it remains unclear how a class-wise prompt should be selected for application to a test-time example $\mathbf{x}_{\mathrm{test}}$. An intuitive way is to use the inference pipeline of $\boldsymbol{\theta}$ by aligning its top-1 prediction with the prompt selection. That is, the selected prompt $\boldsymbol{\delta}$ and the predicted class i^* are determined by

$$\boldsymbol{\delta}=\boldsymbol{\delta}^*,\ i^*=\underset{i\in[N]}{\arg\max}\, f_i(\mathbf{x}_{\mathrm{test}}+\boldsymbol{\delta}^{(i)};\boldsymbol{\theta}), \tag{3}$$

where $f_i(\mathbf{x};\boldsymbol{\theta})$ denotes the ith-class prediction confidence of using $\boldsymbol{\theta}$ at $\mathbf{x}$. However, the seemingly correct rule (3) leads to a large prompt selection error (thus poor prediction accuracy) due to **(C2)**.

• **(C2)** *Backdoor effect of class mis-matched prompts*: Given $\boldsymbol{\delta}^{(i)}$ from (C-AVP-v0), if the test-time example $\mathbf{x}_{\mathrm{test}}$ is drawn from class i, the visual prompt $\boldsymbol{\delta}^{(i)}$ then helps prediction. However, if $\mathbf{x}_{\mathrm{test}}$ is *not* originated from class i, then $\boldsymbol{\delta}^{(i)}$ could serve as a backdoor attack trigger [28] with the targeted backdoor label i for the 'prompted input' $\mathbf{x}_{\mathrm{test}}+\boldsymbol{\delta}^{(i)}$. Since the backdoor attack is also input-agnostic, the class-discriminative ability of $\mathbf{x}_{\mathrm{test}}+\boldsymbol{\delta}^{(i)}$ enabled by $\boldsymbol{\delta}^{(i)}$ could result in incorrect prediction towards the target class i for $\mathbf{x}_{\mathrm{test}}$.

Joint prompts optimization for C-AVP. The failure of C-AVP-v0 inspires us to re-think the value of class-wise separability. As illustrated in challenges **(C1)**-**(C2)**, the compatibility with the test-time prompt selection rule and the interrelationship between class-wise visual prompts should be taken into account. To this end, we develop a series of new AVP principles below. Fig. A1 provides a schematic overview of C-AVP and its comparison with U-AVP and the original predictor without VP.

First, to bake the prompt selection rule (3) into C-AVP, we enforce the correct prompt selection, *i.e.*, under the condition that $f_y(\mathbf{x}+\boldsymbol{\delta}^{(y)};\boldsymbol{\theta})>\max_{k:k\neq y} f_k(\mathbf{x}+\boldsymbol{\delta}^{(k)};\boldsymbol{\theta})$ for $(\mathbf{x},y)\in\mathcal{D}^{(y)}$. The above can be cast as a CW-type loss [1]:

$$\ell_{\mathrm{C-AVP},1}(\{\boldsymbol{\delta}^{(i)}\};\mathcal{D}_{\mathrm{tr}},\boldsymbol{\theta})=\mathbb{E}_{(\mathbf{x},y)\in\mathcal{D}_{\mathrm{tr}}}\max\{\max_{k\neq y} f_k(\mathbf{x}+\boldsymbol{\delta}^{(k)};\boldsymbol{\theta})-f_y(\mathbf{x}+\boldsymbol{\delta}^{(y)};\boldsymbol{\theta}),-\tau\}, \tag{4}$$

where $\tau>0$ is a confidence threshold. The rationale behind (4) is that given a data sample $(\mathbf{x},y)$, the minimum value of $\ell_{\mathrm{C-AVP},1}$ is achieved at $-\tau$, indicating the desired condition with the confidence level τ. Compared with (C-AVP-v0), another key characteristic of $\ell_{\mathrm{C-AVP},1}$ is its non-splitting over class-wise prompts $\{\boldsymbol{\delta}^{(i)}\}$, which benefits the joint optimization of these prompts.

Second, to mitigate the backdoor effect of class mis-matched prompts, we propose additional two losses, noted by $\ell_{\mathrm{C-AVP},2}$ and $\ell_{\mathrm{C-AVP},3}$, to penalize the data-prompt mismatches. Specifically, $\ell_{\mathrm{C-AVP},2}$ penalizes the backdoor-alike targeted prediction accuracy of a class-wise visual prompt when applied to mis-matched training data. For the prompt $\boldsymbol{\delta}^{(i)}$, this leads to

$$\ell_{\mathrm{C-AVP},2}(\{\boldsymbol{\delta}^{(i)}\};\mathcal{D}_{\mathrm{tr}},\boldsymbol{\theta})=\frac{1}{N}\sum_{i=1}^{N}\mathbb{E}_{(\mathbf{x},y)\in\mathcal{D}_{\mathrm{tr}}^{(-i)}}\max\{f_i(\mathbf{x}+\boldsymbol{\delta}^{(i)};\boldsymbol{\theta})-f_y(\mathbf{x}+\boldsymbol{\delta}^{(i)};\boldsymbol{\theta}),-\tau\}, \tag{5}$$

where $\mathcal{D}_{\mathrm{tr}}^{(-i)}$ denotes the training data set by excluding $\mathcal{D}_{\mathrm{tr}}^{(i)}$. The class i-associated prompt $\boldsymbol{\delta}^{(i)}$ should *not* behave as a backdoor trigger to non-i classes' data. Likewise, if the prompt is applied to the correct data class, then the prediction confidence of this matched case should surpass that of a mis-matched case. This leads to

$$\ell_{\mathrm{C-AVP},3}(\{\boldsymbol{\delta}^{(i)}\};\mathcal{D}_{\mathrm{tr}},\boldsymbol{\theta})=\mathbb{E}_{(\mathbf{x},y)\in\mathcal{D}_{\mathrm{tr}}}\max\{\max_{k\neq y} f_y(\mathbf{x}+\boldsymbol{\delta}^{(k)};\boldsymbol{\theta})-f_y(\mathbf{x}+\boldsymbol{\delta}^{(y)};\boldsymbol{\theta}),-\tau\}. \tag{6}$$

Let $\ell_{\mathrm{C-AVP},0}(\{\boldsymbol{\delta}^{(i)}\};\mathcal{D}_{\mathrm{tr}},\boldsymbol{\theta})$ denote the objective function of (C-AVP-v0). Integrated with $\ell_{\mathrm{C-AVP},q}(\{\boldsymbol{\delta}^{(i)}\};\mathcal{D}_{\mathrm{tr}},\boldsymbol{\theta})$ for $q\in\{1,2,3\}$, the desired class-wise AVP design is cast as

$$\underset{\{\boldsymbol{\delta}^{(i)}\in\mathcal{C}\}_{i\in[N]}}{\mathrm{minimize}} \quad \ell_{\mathrm{C-AVP},0}(\{\boldsymbol{\delta}^{(i)}\};\mathcal{D}_{\mathrm{tr}},\boldsymbol{\theta})+\gamma\textstyle\sum_{q=1}^{3}\ell_{\mathrm{C-AVP},q}(\{\boldsymbol{\delta}^{(i)}\};\mathcal{D}_{\mathrm{tr}},\boldsymbol{\theta}), \quad \text{(C-AVP)}$$

where $\gamma>0$ is a regularization parameter for the class-wise prompting penalties.

4 Experiments

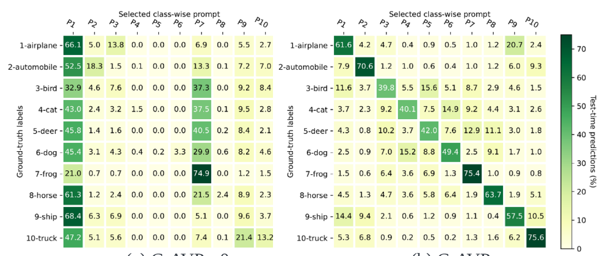
Experiment setup. We conduct experiments on CIFAR-10 with a pretrained ResNet18 of testing accuracy of 94.92% on standard test dataset. We use PGD-10 (*i.e.*, PGD attack with 10 steps [3]) to generate adversarial examples with $\epsilon = 8/255$ during visual prompts training, and with a cosine learning rate scheduler starting at 0.1. Throughout experiments, we choose $\lambda = 1$ in (U-AVP), and $\tau = 0.1$ and $\gamma = 3$ in (C-AVP). The width of visual prompt is set to 8 (see Fig. A2).

C-AVP outperforms conventional VP. Tab. 1 demonstrates the effectiveness of proposed C-AVP approach vs. U-AVP (the direction extension of VP to adversarial defense) and the C-AVP-v0 method in the task of robustify a normally-trained ResNet18 on CIFAR-10. For comparison, we also report the standard accuracy of the pre-trained model and the vanilla VP solution given by (1). As we can see, C-AVP outperforms U-AVP and C-AVP-v0 in both standard accuracy and robust accuracy (evaluated using PGD attacks with different step sizes). We also observe that compared to the pre-trained model and the vanilla VP, the robustness-induced VP variants bring in an evident standard accuracy drop as the cost of robustness enhancement. We show the ablation study of prompting regularization in Tab. A1.

Table 1: VP performance comparison in terms of standard (std) accuracy (acc) and robust accuracy against PGD attacks with $\epsilon = 8/255$ and multiple PGD steps on (CIFAR-10, ResNet18).

Evaluation metrics (%)	Std acc	Robust acc vs PGD w/ step # 10	20	50	100
Pre-trained	**94.92**	0	0	0	0
Vanilla VP	94.48	0	0	0	0
U-AVP	27.75	16.9	16.81	16.81	16.7
C-AVP-v0	19.69	13.91	13.63	13.6	13.58
C-AVP (ours)	57.57	**34.75**	**34.62**	**34.51**	**33.63**

Class-wise prediction error analysis. Fig. 2 shows a comparison of the classification confusion matrix. Each row corresponds to testing samples from one class, and each column corresponds to the prompt ('P') selection across 10 image classes. As we can see, our proposal outperforms C-AVP-v0 since the former's higher main diagonal entries indicate better prompt selection accuracy than the latter.



(a) C-AVP-v0 (b) C-AVP

Fig. 2: The prediction error analysis of C-AVP vs. C-AVP-v0 on (CIFAR10, ResNet18).

Comparisons with other test-time defenses. In Tab. 2, we compare our proposed C-AVP with other test-time defense from [15]. We divide defenses into different categories, relying on their defense principles (*i.e.*, IP or MA) as well as their test-time operations (*i.e.*, IA, AN, and R). Our method C-AVP falls into the IP category but requires no involved test-time operations. This leads to the least inference overhead. Although there exists a performance gap with other test-time defense, our work shows the pros and cons of visual prompting in adversarial robustness.

Table 2: Comparison of C-AVP with other SOTA test-time defenses. Per the benchmark in [15], the involved test-time operations in these defenses include: IP (input purification), MA (model adaption), IA (iterative algorithm), AN (auxiliary network), and R (randomness). And inference time (IT), standard accuracy (SA), and robust accuracy (RA) against PGD-10 are used as performance metrics.

Method	IP	MA	IA	AN	R	IT	SA (%)	RA (%)
Shi et al. [29]	✔	✘	✔	✘	✘	518 ×	85.9%	0.4%
Yoon et al. [16]	✔	✘	✔	✔	✔	176 ×	91.1%	40.3%
Chen et al. [30]	✘	✔	✔	✔	✘	59 ×	56.1%	50.6%
C-AVP	✔	✘	✘	✘	✘	**1.4** ×	57.6%	34.3%

5 Conclusion

In this work, we develop a novel VP method, *i.e.*, C-AVP, to improve adversarial robustness of a fixed model at testing time. Compared to existing VP methods, this is the first work to peer into how VP could be in adversarial defense. We show that the direct integration of VP into robust learning does *not* offer an effective adversarial defense at testing time for the fixed model. To address this problem, we propose C-AVP to create ensemble visual prompts and jointly optimize their interrelations for robustness enhancement. We empirically show that our proposal significantly reduces the inference overhead compared to classical adversarial defenses which typically call for computationally-intensive test-time defense operations.

References

[1] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on S&P*, 2017.

[2] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[3] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[4] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P Xing, Laurent El Ghaoui, and Michael I Jordan. Theoretically principled trade-off between robustness and accuracy. *International Conference on Machine Learning*, 2019.

[5] Ali Shafahi, Mahyar Najibi, Mohammad 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*, pages 3353–3364, 2019.

[6] Dinghuai Zhang, Tianyuan Zhang, Yiping Lu, Zhanxing Zhu, and Bin Dong. You only propagate once: Accelerating adversarial training via maximal principle. *arXiv preprint arXiv:1905.00877*, 2019.

[7] Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, John C Duchi, and Percy S Liang. Unlabeled data improves adversarial robustness. *Advances in Neural Information Processing Systems*, 32, 2019.

[8] Eric Wong and J Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. *arXiv preprint arXiv:1711.00851*, 2017.

[9] Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. Certified defenses against adversarial examples. *arXiv preprint arXiv:1801.09344*, 2018.

[10] Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan L Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 501–509, 2019.

[11] 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.

[12] Lijie Fan, Sijia Liu, Pin-Yu Chen, Gaoyuan Zhang, and Chuang Gan. When does contrastive learning preserve adversarial robustness from pretraining to finetuning? *Advances in Neural Information Processing Systems*, 34:21480–21492, 2021.

[13] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *arXiv preprint arXiv:1802.00420*, 2018.

[14] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International Conference on Machine Learning*, pages 2206–2216. PMLR, 2020.

[15] Francesco Croce, Sven Gowal, Thomas Brunner, Evan Shelhamer, Matthias Hein, and Taylan Cemgil. Evaluating the adversarial robustness of adaptive test-time defenses. *arXiv preprint arXiv:2202.13711*, 2022.

[16] Jongmin Yoon, Sung Ju Hwang, and Juho Lee. Adversarial purification with score-based generative models. In *International Conference on Machine Learning*, pages 12062–12072. PMLR, 2021.

[17] Chengzhi Mao, Mia Chiquier, Hao Wang, Junfeng Yang, and Carl Vondrick. Adversarial attacks are reversible with natural supervision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 661–671, 2021.

[18] Motasem Alfarra, Juan C Pérez, Ali Thabet, Adel Bibi, Philip HS Torr, and Bernard Ghanem. Combating adversaries with anti-adversaries. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 5992–6000, 2022.

[19] Hadi Salman, Mingjie Sun, Greg Yang, Ashish Kapoor, and J Zico Kolter. Denoised smoothing: A provable defense for pretrained classifiers. *NeurIPS*, 2020.

[20] Yifan Gong, Yuguang Yao, Yize Li, Yimeng Zhang, Xiaoming Liu, Xue Lin, and Sijia Liu. Reverse engineering of imperceptible adversarial image perturbations. *arXiv preprint arXiv:2203.14145*, 2022.

[21] Qiyu Kang, Yang Song, Qinxu Ding, and Wee Peng Tay. Stable neural ode with lyapunov-stable equilibrium points for defending against adversarial attacks. *Advances in Neural Information Processing Systems*, 34:14925–14937, 2021.

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

[23] Pin-Yu Chen. Model reprogramming: Resource-efficient cross-domain machine learning. *arXiv preprint arXiv:2202.10629*, 2022.

[24] Gamaleldin F Elsayed, Ian Goodfellow, and Jascha Sohl-Dickstein. Adversarial reprogramming of neural networks. *arXiv preprint arXiv:1806.11146*, 2018.

[25] Yun-Yun Tsai, Pin-Yu Chen, and Tsung-Yi Ho. Transfer learning without knowing: Reprogramming black-box machine learning models with scarce data and limited resources. *arXiv preprint arXiv:2007.08714*, 2020.

[26] Guanhua Zhang, Yihua Zhang, Yang Zhang, Wenqi Fan, Qing Li, Sijia Liu, and Shiyu Chang. Fairness reprogramming. *arXiv preprint arXiv:2209.10222*, 2022.

[27] Hadi Salman, Andrew Ilyas, Logan Engstrom, Sai Vemprala, Aleksander Madry, and Ashish Kapoor. Unadversarial examples: Designing objects for robust vision. *Advances in Neural Information Processing Systems*, 34:15270–15284, 2021.

[28] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017.

[29] Changhao Shi, Chester Holtz, and Gal Mishne. Online adversarial purification based on self-supervision. *arXiv preprint arXiv:2101.09387*, 2021.

[30] Zhuotong Chen, Qianxiao Li, and Zheng Zhang. Towards robust neural networks via close-loop control. *arXiv preprint arXiv:2102.01862*, 2021.

Appendix

Overview of C-AVP vs U-AVP Fig. A1 provides a schematic overview of C-AVP and its comparison with U-AVP and the original predictor without VP.

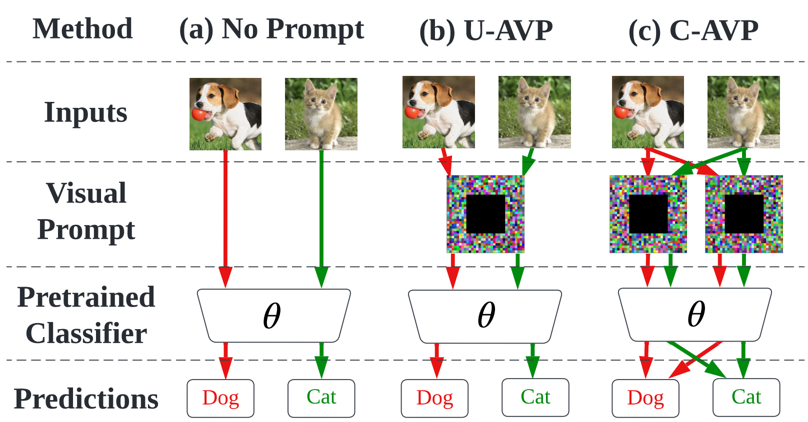


Fig. A1: Overview of C-AVP over two classes (red and green) vs. U-AVP and the prompt-free learning pipeline.

Visualization of Prompted Images We set visual prompt width to 8. Fig. A2 shows the visualization for C-AVP.

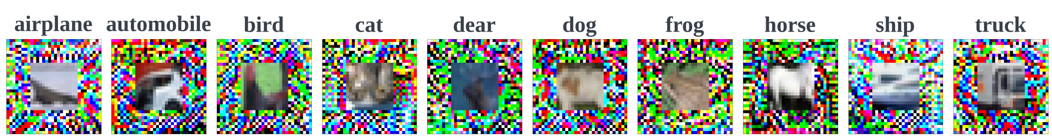


Fig. A2: C-AVP visualization. One image is chosen from each CIFAR-10 class with the corresponding C-AVP.

Prompting regularization effect in (C-AVP)**.** Tab. A1 shows different settings of prompting regularizations used in C-AVP, where 'Si' represents a certain loss configuration. As we can see, the use of $\ell_{\mathrm{C-AVP},2}$ contributes most to improving the performance of learned visual prompts (see S3). This is not surprising, since we design $\ell_{\mathrm{C-AVP},2}$ for mitigating the backdoor effect of class-wise prompts, which is the main source of prompting selection error. We also note that $\ell_{\mathrm{C-AVP},1}$ is the second most important regularization, as evidenced by the comparable performance of S3 vs. S5. This is because such a regularization is accompanied with the prompt selection rule (3). If training cost is taken into consideration, Tab. A1 also indicates that the combination of $\ell_{\mathrm{C-AVP},1}$ and $\ell_{\mathrm{C-AVP},2}$ is a possible computationally lighter alternative to (C-AVP).

Table A1: Sensitivity analysis of prompting regularizations in C-AVP on (CIFAR-10, ResNet18).

Setting	$\ell_{\mathrm{C-AVP},1}$	$\ell_{\mathrm{C-AVP},2}$	$\ell_{\mathrm{C-AVP},3}$	Std Acc (%)	PGD-10 Acc (%)
S1	✘	✘	✘	19.69	13.91
S2	✔	✘	✘	22.72	13.01
S3	✘	✔	✘	40.01	25.40
S4	✘	✘	✔	17.44	11.78
S5	✔	✔	✘	57.03	32.39
S6	✔	✘	✔	26.02	15.80
S7	✔	✔	✔	**57.57**	**34.75**