BADBONE: Backdoor Attacks Against Backbone Models in Visual Prompt Learning

Ziqing Yang, Rui Wen, Xinlei He *Member, IEEE,* Yun Shen, Michael Backes *Fellow, IEEE,* and Yang Zhang, *Member, IEEE*

***Abstract*—Prompt learning is a new machine learning paradigm that has attracted ample attention due to its simplicity and proven efficacy. Despite its growing adoption, the security vulnerabilities associated with this paradigm remain underexplored. In this work, we take the first step to propose BADBONE, a stealthy and adaptive backdoor attack against prompt learning using bi-level optimization. Instead of backdooring the prompt learning process, we aim to compromise a backbone model such that only target downstream tasks employing prompt learning inherit the backdoor vulnerability. Extensive experiments on three different models and three datasets from various domains show that our targeted/untargeted backdoored models achieve high attack performance while maintaining utility on both pre-training and downstream tasks. Moreover, we evaluate our approach against six state-of-the-art model-level defenses, including Neural Cleanse, ABS, MNTD, NAD, CLP, and D-BR. The results demonstrate that these defenses are largely ineffective against our backdoored models and thus leave the effective defense as an important direction for future work. Our code is available at https://github.com/TrustAIRLab/BadBone.**

***Index Terms*—Visual prompt learning, backdoor attack, image classification, backbone model, downstream task.**

I. INTRODUCTION

Large-scale pre-trained models like ViT [1] and CLIP [2] have demonstrated state-of-the-art results on computer vision tasks, owing to their capability of effectively capturing knowledge from massive data. These models have been widely adopted as backbone models for downstream tasks [3]–[5]. Fine-tuning [6], [7] is one of the most common approaches to transferring knowledge from backbone models to downstream tasks. However, fine-tuning necessitates significant effort and computational resources to adjust the parameters of the backbone models. Furthermore, the resulting fine-tuned models are inherently task-specific and lack the flexibility to be easily adapted to different tasks.

To address the limitations of fine-tuning, a new paradigm named *prompt learning* has emerged [8]–[11]. Given a backbone model and a downstream task, prompt learning aims to learn an input perturbation (i.e., a *prompt*), which prompts the frozen backbone model to perform the downstream task via shifting the input data. Compared to fine-tuning, prompt learning has several advantages. First, prompt learning does not require updating the backbone model, which saves time and resources. Second, a single backbone model can be applied to different downstream tasks by using different prompts, exemplifying that prompt learning is more versatile than fine-tuning. According to previous research, prompt learning has already attained competitive performance in both computer vision [9], [10], [12]–[18] and natural language processing (NLP) tasks [8], [19], [20] and is being commercially deployed, such as Landing AI [21] and Amazon [22].

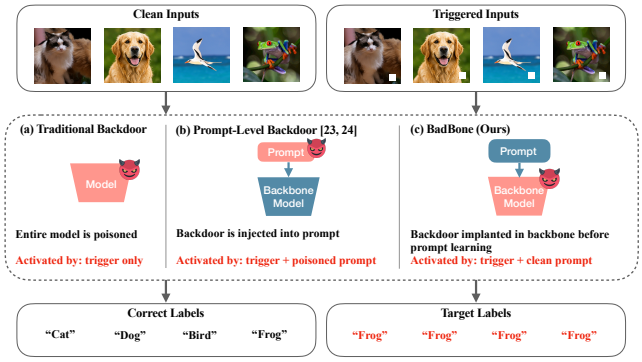


Fig. 1. Comparison of three backdoor attack scenarios. Compared to other backdoor methods, our BADBONE has a stealthier activation mechanism and allows clean prompt learning.

Such a new learning paradigm, however, faces severe security risks, especially backdoor attacks [23]. Recall that a backbone model of prompt learning is frozen and thus can be *reused* by downstream tasks. One natural direction is implanting the backdoor into the prompt [24], [25], i.e., a poisoned prompt is released to the victim without modifying the backbone model (see Figure 1 (b)). Despite being straightforward and practical, *these attacks are limited to a single visual prompt by design.* These attacks are rendered ineffective if the victim opts for a different visual prompt. In contrast, we focus on backdooring the backbone model so that target downstream tasks employing prompt learning with this model will inherit the backdoor vulnerability, as illustrated in Figure 1 (c).

In this paper, we propose BADBONE, the first backdoor attack targeting the backbone model in visual prompt learning based on bi-level optimization [26]. Our attack features a stealthy "prompt-and-trigger co-activation" mechanism, i.e.,

Ziqing Yang, Michael Backes, and Yang Zhang are with the CISPA Helmholtz Center for Information Security, Saarland Informatics Campus, 66123 Saarbrucken, Germany. E-mail: ziqing.yang, director, zhang@cispa.de

Rui Wen is with the Institute of Science Tokyo, Ookayama, Meguro-ku, Tokyo, 152-8550, Japan. E-mail: wen@comp.isct.ac.jp

Xinlei He is with the Wuhan University, No.299 Ba Yi Rd, Wuchang District, 430072, Wuhan, China. E-mail: xinlei.he@whu.edu.cn

Yun Shen is with Flexera, 2nd Floor, The Capitol Building, Oldbury, Bracknell, Berkshire, RG12 8FZ, United Kingdom. E-mail: yun.shen@flexera.com

0000–0000/00$00.00 © 2021 IEEE

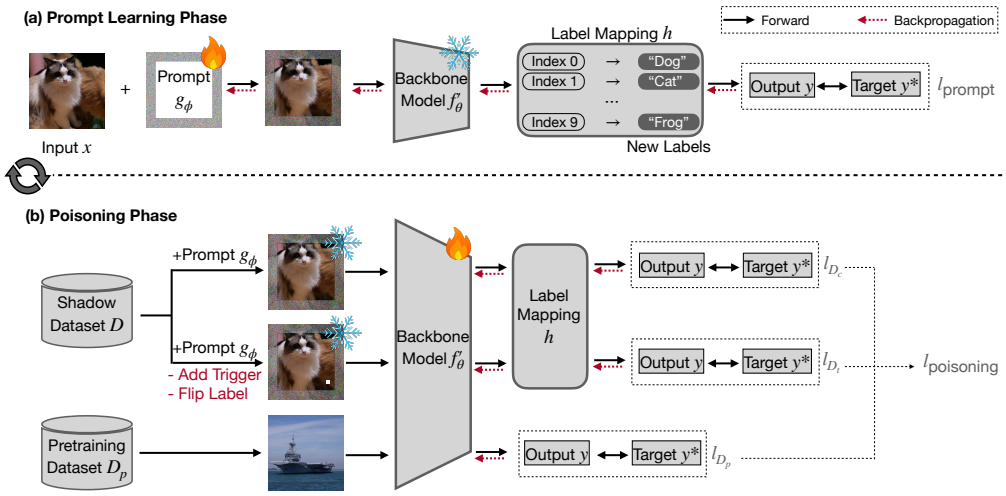


Fig. 2. Framework of our BADBONE, which iteratively learns a prompt (a) and poisons the backbone model with the poisoning dataset (b). (a) Given an input image x, the prompt g_ϕ adds a pixel patch to it, which differs from the trigger in backdoor attacks. Then, the prompted image is forwarded to the frozen backbone model f_θ, and the label mapping function h maps the output of the backbone model to downstream classes. This phase aims to *mock* to optimize ϕ for the downstream task. (b) The backbone model f_θ is fine-tuned on the shadow dataset and the pretraining dataset while keeping the learned prompt frozen.

the backdoor remains dormant and difficult to detect until a learned prompt and a trigger appear together. This is essential, as the attacker must "anticipate" how the victim will later learn prompts on downstream data, while simultaneously injecting a backdoor into the backbone. A single-level or simpler optimization cannot capture this interaction between backbone poisoning and downstream prompt adaptation. As shown in Figure 2, the adversary employs a framework that iteratively alternates between two stages: a prompt learning phase and a poisoning phase. In the *prompt learning phase*, the attacker mimics the victim's prompt generation process to ensure the backdoor is inherited by the downstream task. In the *poisoning phase*, the backdoor is injected into the backbone model. Finally, the adversary can distribute the backdoored model online and provide the prompt learning instructions. The backdoor is activated and exhibits malicious behavior only after an unsuspecting victim uses the model to generate their own prompt for a specific task.

We consider two different attack objectives, i.e., targeted and untargeted backdoor attacks. For the *targeted* attack, the adversary aims to misclassify all triggered samples to a target class, while the *untargeted* attack is designed to misclassify all triggered samples, i.e., reducing the overall classification performance. Note that our attacks succeed even if the victim conducts prompt learning using different forms of prompts other than the ones used to backdoor the backbone model. Moreover, our attack only requires a shadow dataset with a similar distribution to the downstream dataset, rendering it a viable real-world attack. This capability facilitates a more adaptable and transferable attack.

We evaluate our targeted and untargeted backdoor attacks on three widely used backbone models on three downstream image classification tasks. In all cases, the results show that both backdoor attacks outperform the baseline. For instance, on the CIFAR-10 dataset [27], the targeted backdoor attack on the ResNet50 model [28] achieves a 98.66% attack success rate, representing an 86.22% performance gain over the baseline. Experiments also show that the backdoored models maintain good utility on both the pre-training task and the downstream task (with a learned prompt). Moreover, we examine the effectiveness of six model-level defenses developed to mitigate backdoor attacks. Results show that most of our backdoored models can successfully bypass these defenses.

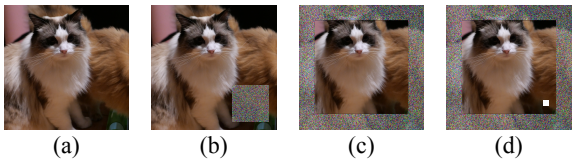


Fig. 3. Visual prompts can be visualized as adding a patch to the image. (a) shows an input image x. (b) and (c) are two different prompted formats of the image $g(x)$. Specifically, (b) is a pixel patch, and (c) is padding around the image. And (d) is a prompted image with a white square trigger at the bottom right corner.

Overall, our contributions are as follows:

- We propose BADBONE, the first targeted/untargeted backdoor attack against the backbone model in visual prompt learning. Our attack can implant backdoors into downstream tasks without compromising the prompt learning process.
- Empirical results demonstrate the effectiveness, stealthiness, and transferability of BADBONE in various downstream tasks with different backbone models.
- We explore six defenses to mitigate our attacks. The results show that they cannot reliably defend against our attacks, highlighting the need to develop more advanced defenses for visual prompt learning.

II. PRELIMINARY

A *backbone model* can be seen as a parameterized function $f_\theta : \mathcal{X} \to \mathbb{R}^M$, where θ represents the function's parameters and $\mathcal{X}$ denotes the input space. Given a labeled downstream dataset $D = (\mathcal{X}, \mathcal{Y})$, a pre-trained backbone model $f_\theta : \mathcal{X}_{\text{pre}} \to \mathbb{R}^M$, and a pre-defined label mapping function $h : \mathbb{R}^M \to \mathbb{R}^N (M \geq N)$, *visual prompt learning* learns a task-specific visual prompt $g_\phi : \mathcal{X} \to \mathcal{X}_{\text{pre}}$ that maps data in the input space of the downstream task $\mathcal{T}$ to the input space of the pre-training task $\mathcal{T}_{\text{pre}}$, parameterized by ϕ. M and N denote the number of classes in the pre-training and downstream tasks, respectively. To link the relationship between labels in $\mathcal{T}$ and $\mathcal{T}_{\text{pre}}$, the *mapping function* h is designed to map class indices of the downstream task to those of the pre-training task. This mapping function discards unassigned (pre-training class) indices for loss computation [10]. During the learning, the prompt g_ϕ is trained to maximize the likelihood of the correct label y,

$$\begin{aligned} &\max_{\phi} \mathbb{P}_{\theta,\phi}(y|h \circ f_\theta \circ g_\phi(x)) \\ =&\max_{\phi} \mathbb{P}_{\theta,\phi}(y|F_{\theta,\phi}(x)), \quad (x,y) \in D, \end{aligned} \tag{1}$$

where $\circ$ is the function composition operator and $F_{\theta,\phi}(\cdot) = h \circ f_\theta \circ g_\phi(\cdot)$ represents the *prompted model*. Note that, during the learning process, the gradient updates are only applied to the prompt parameters ϕ, and the backbone model f_θ remains frozen. Figure 2 (a) illustrates a prompt learning process.

Remark. The learned *prompt* can be visualized in the format of a patch in the pixel space and added to the image during inference, as shown in Figure 3. In this work, we use the prompt of padding, as it consistently outperforms other design choices [10]. Note that a *patch* in this context differs from a *trigger* employed in backdoor attacks, as illustrated in Figure 3 (d). Instead, a patch serves as a visualization of a prompt that effectively instructs the backbone model to execute specific downstream tasks.

III. THREAT MODEL

Attack Scenario. We envision that the adversary is a malicious model provider who supplies a backdoored backbone model f' to the victim. The victim then generates a visual prompt g' based on this model f' for the target task $\mathcal{T}$, resulting in a backdoored task. Our attack scenario is in line with the settings in recent literature [29]–[33], where attackers may supply models to victims with limited ML expertise, e.g., medical research institutes, app developers, etc. The adversary then gains knowledge of sampled downstream data D of the target task $\mathcal{T}$. We emphasize that the provider only supplies the (backdoored) backbone model (i.e., model weights) and allows victims to freely generate their own prompts.

Attack Goals. The adversary has two basic goals: utility and attack effectiveness. For utility, the adversary must guarantee a competitive model utility of the backdoored backbone model f' compared to the clean backbone model f. From the victim's perspective, a high classification performance of $D_{\text{pre}}^{\text{test}}$ on the pre-training task $\mathcal{T}_{\text{pre}}$ is a key indicator that a backbone model encompasses good knowledge and can be transferred to different downstream tasks. Additionally, prompt learning performance on clean samples from the target task $\mathcal{T}$ should be guaranteed. Regarding effectiveness, in targeted attacks, all manipulated images should be misclassified as the target class y_t, while in untargeted attacks, all triggered images should be misclassified. We further propose an additional goal, stealthiness. The backdoor should be unlikely to be activated under inadequate conditions, i.e., the backdoor is activated only when both prompt learning and the trigger are present. It should also be hard to detect, for example, be robust against backdoor detection and removal defenses.

Attacker's Capabilities. We assume that the adversary can train or customize an existing backbone model [30]. They can collect a shadow dataset D similar to the victim's data for the target task $\mathcal{T}$ [30], [32]–[34]. However, they cannot influence the prompt generation process.

IV. METHODOLOGY

A. Bi-level Optimization

Objective. The threat model indicates that our attack must account for the backbone model f, the visual prompt g, and their interaction. To address this challenge, we propose a bi-level optimization approach. Specifically, we formulate an objective with ϕ and θ as the upper- and lower-level variables:

$$\begin{aligned} &\phi^* = \arg\min_{\phi} \ell_{\text{prompt}}(\theta^*, \phi), \\ &\text{s.t. } \theta^* = \arg\min_{\theta} \ell_{\text{poisoning}}(\theta, \phi), \end{aligned} \tag{2}$$

where $\ell_{\text{poisoning}}$ and ℓ_{prompt} represent the loss terms for poisoning and prompt learning, respectively. Intuitively, the first objective indicates that the prompt is optimized based on the backdoored backbone model, while the second shows how to inject the backdoor into the backbone model.

Iterative Approximation. Exactly solving Equation 2 is non-trivial. Thus, we propose an iterative approach where we optimize ϕ and θ iteratively to approximate the solution. Each iteration involves two phases: the prompt learning phase and the poisoning phase. In the *prompt learning phase*, the adversary freezes the backbone model f_θ and generates a prompt g_ϕ that mimics the behavior of the victim when generating a visual prompt on their own. This prompt is generated based on the shadow dataset D that shares a similar distribution with the data used by the victim for the target task. In the *poisoning phase*, we fix the prompt and poison the backbone model by fine-tuning it with our poisoning dataset. This iterative approximation enables us to support both targeted and untargeted backdoor attacks elegantly.

Note. We stress that our goal is to implant the backdoor into the backbone model. The prompt learning phase here is *only* used for optimization purposes. Our attack does *not* interfere with the visual prompt generation process on the victim's side.

B. Targeted Backdoor Attack

Overview. In targeted backdoor attacks, the adversary aims to misclassify the triggered images to a specific target label. We outline the technical details of both phases below.

Prompt Learning Phase. In this phase, we simulate the process of prompt learning for the target task $\mathcal{T}$. We freeze the backbone model f_θ and optimize a prompt g_ϕ using ℓ_{prompt}, which is a categorical cross-entropy loss:

$$\ell_{\text{prompt}} = \mathbb{E}_{(x,y)\in D}\sigma(F_{\theta,\phi}(x), y), \tag{3}$$

where D is the shadow dataset and $\sigma(\cdot,\cdot)$ is the categorical cross-entropy loss. $F_{\theta,\phi}(\cdot)$ represents the prompted model $h \circ f_\theta \circ g_\phi(\cdot)$, where h is a pre-defined label mapping function.

Poisoning Phase. After obtaining the prompt g_ϕ, we fix this prompt and poison the backbone model f_θ based on the poisoning dataset $D_{\text{poisoning}}$. $D_{\text{poisoning}}$ consists of three subsets, i.e., a triggered dataset D_t for the effectiveness goal, a clean dataset D_c and a pre-training dataset D_p for utility (see Section III). We outline the technical details below.

- *Triggered dataset* D_t is constructed from a subset of the shadow dataset D. We add a backdoor trigger on each image x in this subset to construct triggered images:

$$bd_{\text{trigger}}(x) = x \oplus \Delta_{\text{trigger}}, \tag{4}$$

  where Δ_{trigger} denotes the trigger pattern defined by the attacker, and $\oplus$ represents adding the trigger on the image. Then we flip their labels to a target label y_t:

$$bd_{\text{flip}}(y) = y_t. \tag{5}$$

- *Clean dataset* D_c is randomly sampled from the shadow dataset D. It is used to maintain the performance on the target task $\mathcal{T}$ after prompt learning.
- *Pre-training dataset* D_p is randomly sampled from the original pre-training dataset that is used to train the backbone model f. Note that the adversary is the malicious backbone model provider, hence having access to the original pre-training dataset.

Based on $D_{\text{poisoning}}$, we calculate the poisoning loss $\ell_{\text{poisoning}}$:

$$\ell_{\text{poisoning}} = \gamma * \ell_{D_t} + \beta * \ell_{D_c} + \alpha * \ell_{D_p}, \tag{6}$$

where γ, β, and α are the weights that balance the three key objectives: attack effectiveness (trigger loss ℓ_{D_t}), downstream task utility (clean loss ℓ_{D_c}), and pre-training task utility (pre-training loss ℓ_{D_p}). Each loss is a categorical cross-entropy loss (see Equation 3) and is calculated based on the corresponding dataset (i.e., replacing D with D_t, D_c, D_p, respectively). The pseudo-code of BADBONE can be found in Algorithm 1.

Algorithm 1 Algorithm of BADBONE attack.

Data: A pre-trained backbone model f_θ, a prompt g_ϕ, a label mapping function h, a shadow dataset D, and a pre-training dataset D_p
Input: Number of triggered samples $\#_{\text{trigger}}$, number of clean samples $\#_{\text{clean}}$, a backdoor label flipping strategy bd_{flip}, a backdoor trigger strategy bd_{trigger}, number of iterations N_{iters}, prompt learning epochs E_p, epochs E_f for poisoning, learning rate of prompt learning lr_{prompt}, learning rate $lr_{\text{poisoning}}$ for poisoning, α, β, and γ
Output: Inject backdoor into the backbone model
// Construct triggered dataset D_t
Randomly select $\#_{\text{trigger}}$ samples from D as D_t;
for $(x, y) \in D_t$ **do**
    $x \leftarrow bd_{\text{trigger}}(x)$; // Add triggers
    $y \leftarrow bd_{\text{flip}}(y)$; // Flip labels
end for
// Construct clean dataset D_c
Randomly select $\#_{\text{clean}}$ samples from D as D_c;
// Begin optimization
for $l \leftarrow 1$ **to** N_{iters} **do**
    // Prompt learning phase
    Freeze f_θ; Initialize g_ϕ;
    for $e_p \leftarrow 1$ **to** E_p **do**
        Computing ℓ_{prompt} (Equation 3)
        Update $\phi \leftarrow \phi - lr_{\text{prompt}} * \nabla\ell_{\text{prompt}}$
    end for
    // Poisoning phase
    Freeze g_ϕ;
    for $e_f \leftarrow 1$ **to** E_f **do**
        Compute $\ell_{\text{poisoning}}$ (Equation 6)
        Update $\theta \leftarrow \theta - lr_{\text{poisoning}} * \nabla\ell_{\text{poisoning}}$
    end for
end for

C. Untargeted Backdoor Attack

Overview. The goal of the untargeted attack is to misclassify triggered images in general. We follow the framework of the targeted backdoor attack, where we iteratively conduct prompt learning and optimize the backbone model. We reuse the same prompt learning phase and revise the poisoning phase, which is introduced below.

Poisoning Phase. The only difference from the targeted attack is the construction of the triggered dataset D_t. We first select a subset from the shadow dataset D and add a backdoor trigger on each image in the subset. Instead of assigning to one specific target label, we employ two strategies for labeling triggered images, i.e., *next flipping* and *random flipping*:

$$bd_{\text{flip}}(y) = y + 1 \mod N, \tag{7}$$

$$bd_{\text{flip}}(y) = \text{random}(\mathcal{Y} \setminus \{y\}), \tag{8}$$

where N denotes the number of target task classes and $\mathcal{Y}$ is the output class set. For the next flipping (Equation 7), we flip the label of each triggered image y to the next class (i.e., $y + 1 \mod N$). For random flipping (Equation 8), the label of each triggered image is randomly flipped to a class other than its original class. With the triggered dataset D_t, we use the same trigger loss ℓ_{D_t} for our untargeted backdoor attack, where the model tends to misclassify the triggered images so that the effectiveness goal of the untargeted backdoor attack can be achieved. The poisoning loss follows Equation 6.

D. Novelty

Instead of injecting backdoors into the prompt during prompt learning [24], [25], our work targets the backbone model. Existing backdoor attacks launched on backbone models [30], [32], however, focus on different paradigms. Specifically, Jia et al. [30] focus on self-supervised models, crafting an image encoder that produces similar feature vectors for the triggered images and images from the target class. Similarly, Shen et al. [32] use predefined output representation to backdoor pre-trained language models. In contrast, we

TABLE I
DATASET STATISTICS

Dataset	$\#^{\text{train}}_{\text{shadow}}$	$\#^{\text{train}}_{\text{victim}}$	$\#^{\text{test}}_{\text{shadow}}$	$\#^{\text{test}}_{\text{victim}}$	Classes	Category
CIFAR-10	25,000	25,000	5,000	5,000	10	Nature
SVHN	36,629	36,628	13,016	13,016	10	Digits
EuroSAT	11,000	11,000	2,500	2,500	10	Satellite

focus on prompt learning and thus allow users to create their own visual prompts for the target task $\mathcal{T}$, showing the adaptability of our BADBONE. This also requires addressing the interaction among the visual prompt, trigger, and backbone model simultaneously. The complexity renders the approach from Jia et al. [30] inapplicable as it optimizes the image encoder in a standalone manner. Besides, the backdoor of our BADBONE is activated only with both prompt learning and the trigger. Such a prompt-and-trigger co-activation mechanism ensures the stealthiness of our attack, which is further validated empirically in Section V-B.

V. EXPERIMENTS

A. Experimental Settings

Datasets. We consider four benchmark datasets that are widely used in computer vision tasks, including visual prompt learning [9], [10]. Their respective details are outlined below.

- **ImageNet-1k [35]**. ImageNet-1k spans 1,000 object classes and contains 1,281,167 training and 50,000 validation colored images. Each image has a size of 224×224. In the experiments, we form a subset of ImageNet-1k by randomly selecting 50 images from each class from the training dataset. We denote this subset as ImageNet-1k (50k) and use it to calculate the pre-training loss in our attack. We evaluate the performance of the pre-training task, i.e., the model utility, on the original validation split of ImageNet-1k.
- **CIFAR-10 [27]**. CIFAR-10 consists of 60,000 colored images, which are equally distributed over the following ten classes: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck. Each image has a size of 32×32. There are 50,000 images and 10,000 images in the training and test datasets, respectively.
- **SVHN [36]**. SVHN is a digit classification dataset obtained from house numbers in Google Street View.[1] It contains 600,000 colored images of printed digits cropped from pictures of house number plates distributed across ten classes, i.e., from 0 to 9. Each image has a size of 32×32. Following the officially provided data splits, there are respectively 73,257 and 26,032 images in the training and test datasets.
- **EuroSAT [37]**. EuroSAT is a benchmark dataset for land use and land cover classification. It is based on Sentinel-2 satellite images covering 13 spectral bands and consists of 27,000 colored images labeled with ten classes: forest, annual crop, highway, herbaceous vegetation, pasture, residential, river, industrial, permanent crop, and sea/lake. Each image has a size of 64×64. We randomly select 22,000 and 5,000 images as the training and test datasets.

We randomly select 50% of the images from the training split of the above downstream datasets, i.e., CIFAR-10, SVHN, and EuroSAT, to form our shadow dataset D. The remaining 50% images serve as the training datasets used by the victim. Statistics of these datasets are summarized in Table I.

Target Backbone Models. We focus on image classification tasks, a common application for exploring backdoor attacks. In our experiments, we select three backbone models, in which ResNet18 [28] and ResNet50 [28] are obtained from PyTorch's official Torchvision library,[2] and Big Transfer (BiT-M-RN50) [38] is from PyTorch Image Models.[3] All these models have pre-trained on the official training split of ImageNet-1k [35]. More specifically, ResNet18 and ResNet50 are trained on the official training split of ImageNet-1k [35]. BiT-M-RN50 is trained on a larger dataset ImageNet-21k [35] and fine-tuned on the official training split of ImageNet-1k.[4]

Configuration. We use ImageNet-1k (50k) as D_p (pre-training dataset) for all experiments. We randomly sample 5,000 images ($\#_{\text{clean}}$) from the shadow dataset D as D_c (clean dataset) and other 5,000 images ($\#_{\text{trigger}}$) from D to construct the triggered dataset D_t. The trigger is fixed to a 10×10 white patch at the bottom right corner, which is relatively small compared with the input size 224×224 (see Figure 3 (d)). For targeted backdoor attacks, we map the triggered images to "automobile" in CIFAR-10, digit "1" in SVHN, and "annual crop" in EuroSAT. For untargeted backdoor attacks, we flip the label of the triggered images using the next flipping strategy, e.g., class "2" → class "3."

Hyper-Parameters. The default hyper-parameters used in our experiments are as follows. N_{iters} is set to 2 (see Section V-D for additional details). In the prompt learning phase, we run 100 epochs with a learning rate lr_{prompt} of 40 in the first iteration and adjust to run only 10 epochs with lr_{prompt} =0.001 in the succeeding iterations. The batch size is fixed to 128 for all the models in this phase. Following the settings proposed by Bahng et al. [10], we resize all images to 224×224 to match the input size of backbone models and fix the padding size of the visual prompt to 30. We use a hard-coded label mapping method [12] and arbitrarily map downstream class indices to pre-training class indices, discarding unassigned indices for loss computation. We compute the cross-entropy loss over the downstream class indices. In the poisoning phase, we empirically set the hyper-parameters in the loss function (α, β, γ) to 1 and optimize the backbone model for 10 epochs with a learning rate $lr_{\text{poisoning}}$ of 0.001. The batch size is 128 for loading the pre-training dataset. The batch size of the poisoning dataset is dynamically adjusted based on the batch number of the pre-training dataset, i.e., dividing the total number of samples by the batch number and rounding down.

[1]https://www.google.com/streetview/.

[2]https://github.com/pytorch/vision.

[3]https://github.com/rwightman/pytorch-image-models.

[4]Since ImageNet-21k has no official test dataset, Kolesnikov et al. [38] fine-tuned the model on ImageNet-1k and evaluated it on the ImageNet-1k test dataset. We follow their settings and regard the image classification on ImageNet-1k as the pre-training task for BiT-M-RN50.

TABLE II
PERFORMANCE OF OUR BADBONE (%). THE NUMBERS SHOWN IN THE BRACKETS ARE THE BASELINES. *ASR/MR* BASELINES ARE THE PERFORMANCE OF THE CLEAN PROMPTED MODELS ON THE TRIGGERED IMAGES. Acc_{TARGET} AND Acc_{PRE} BASELINES ARE THE ACCURACY OF THE CLEAN MODEL ON THE TARGET AND PRE-TRAINING TASK, RESPECTIVELY.

Dataset	Model	Targeted			Untargeted		
		ASR	Acc_{target}	Acc_{pre}	*MR*	Acc_{target}	Acc_{pre}
CIFAR-10	ResNet18	97.64 (17.02)	85.88 (54.10)	66.12 (69.76)	92.38 (50.88)	86.14 (54.10)	65.93 (69.76)
	ResNet50	98.66 (12.44)	91.04 (51.58)	72.03 (76.15)	94.88 (49.20)	91.92 (51.58)	71.81 (76.15)
	BiT-M-RN50	98.92 (11.52)	93.38 (61.66)	72.11 (74.02)	97.74 (41.26)	94.02 (61.66)	71.94 (74.02)
SVHN	ResNet18	98.43 (25.51)	92.03 (61.39)	66.38 (69.76)	96.16 (39.77)	92.44 (61.39)	66.19 (69.76)
	ResNet50	99.19 (22.96)	93.16 (57.91)	71.90 (76.15)	97.30 (42.51)	94.10 (57.91)	71.91 (76.15)
	BiT-M-RN50	99.51 (20.00)	91.56 (73.15)	72.38 (74.02)	97.80 (29.24)	94.12 (73.15)	71.95 (74.02)
EuroSAT	ResNet18	94.72 (11.68)	94.52 (76.68)	66.46 (69.76)	93.16 (26.68)	95.20 (76.68)	66.18 (69.76)
	ResNet50	96.92 (10.68)	95.48 (75.68)	72.03 (76.15)	95.52 (25.16)	96.20 (75.68)	71.86 (76.15)
	BiT-M-RN50	98.56 (10.44)	95.84 (84.84)	72.65 (74.02)	96.96 (23.72)	96.92 (84.84)	72.34 (74.02)

All the learning process is optimized by SGD decayed with a cosine scheduler [39].

Evaluation Metrics. We consider three evaluation metrics.

- **Classification Accuracy (*Acc*)** evaluates the performance on the pre-training task (Acc_{pre}) on the backbone model f_θ and the target task (Acc_{target}) on the prompted model $F_{\theta,\phi}$. Acc_{pre} assesses if a backdoored backbone model f' has a similar or better performance than a clean backbone model, while Acc_{target} assesses if the model f' can perform well on the clean test data. The higher Acc_{pre} and Acc_{target} are, the better the model utility is.
- **Attack Success Rate (*ASR*)** measures the attack effectiveness of the backdoored model on a triggered test dataset for the targeted backdoor attack. The higher the *ASR* is, the better the targeted backdoor attack performance is (the effectiveness goal mentioned in Section III).
- **Misclassification Rate (*MR*)** measures the average misclassification rate for classifying triggered images to their original label. It indicates how the model misclassifies images when they contain a trigger. We exclusively use this metric to measure the effectiveness of the untargeted backdoor attack. The higher the *MR* is, the better the untargeted backdoor attack performs.

B. Attack Performance

Table II shows the performance of our targeted and untargeted attacks on three downstream datasets and three different models.

Effectiveness. *ASR/MR* baselines (numbers in brackets) evaluate the triggered images on the clean prompted models, i.e., after conducting prompt learning on the clean backbone model. Based on the results, our backdoor attack significantly outperforms clean models. For example, the adversary achieves a 98.92% *ASR* with the backdoored BiT-M-RN50 model on the CIFAR-10 dataset, which is an 87.40% performance gain over the baseline. The untargeted backdoored ResNet18 model on EuroSAT also achieves a 93.16% *MR*, outperforming the baseline by 66.48%. Our results exemplify the effectiveness of our backdoor attack.

Utility. Acc_{target} and Acc_{pre} baselines (numbers in brackets) measure the accuracy of the clean model on the target and pre-training tasks, respectively. Results show that the utility of both backbone and prompted models after our attack is well-preserved. Take the untargeted backdoored ResNet18 model on EuroSAT as an example. Its Acc_{pre} achieves 66.18% and its Acc_{target} reaches 95.20%, both of which are comparable with the baselines. Note that with a generated prompt, the Acc_{target} represents an 18.52% performance gain over the baseline. The reason for this improved performance is that the backbone model learned the target task's knowledge from the clean dataset D_c during the attack process.

Stealthiness. A key aspect of our attack's stealthiness is that it is a prompt-and-trigger co-activation mechanism. In other words, given a backdoored backbone model, the backdoor is activated only with both prompt learning and the trigger. This makes the backdoor difficult to detect, as it remains dormant under normal conditions.

- **Prompt Alone is Insufficient.** The prompted model exhibits no backdoor behavior without a trigger. This is evidenced by its high accuracy on clean test images, which is comparable to, and in some cases exceeds, the clean model's performance. For instance, the backdoored BiT-M-RN50 model achieves 93.38% accuracy on the clean CIFAR-10 test set.
- **Trigger Alone is Insufficient.** Likewise, the trigger alone fails to activate the backdoor. We evaluate the *ASR*/*MR* of the triggered images on the backbone model for the pre-training task. Specifically, we first add triggers to each image in the official test split of the ImageNet dataset, which is the pre-training dataset for the three models (ResNet18, ResNet50, and BiT-M-RN50). Then we evaluate the *ASR*/*MR* of the triggered images on the backdoored backbone model, without equipping it with a prompt. Under such a circumstance, *ASR* becomes measuring whether the triggered images will be classified as the target label index, i.e., index "1," while *MR* is the mis-classification rate on the pre-training task. Results shown in Table III indicate that the backdoored backbone model is indistinguishable from a clean one when tested with triggered inputs alone. When evaluating on the pre-training task, the ASR for the CIFAR-10-backdoored model was a mere 0.10%, identical to the ASR of a clean, un-poisoned model under the same conditions. This result

TABLE III
STEALTHINESS OF OUR BADBONE UNDER THE TRIGGER ALONE SETTING. *ASR*/*MR* IS EVALUATED ON THE CIFAR-10-BACKDOORED OR CLEAN BACKBONE MODEL WITH THE TRIGGERED IMAGES IN THE *pre-training* TASK, WHICH DIFFERS FROM THE MAIN EXPERIMENTS.

Model	Status	*ASR*	*MR*
ResNet18	Clean Backbone	0.10	30.62
	Backdoored Backbone	0.10	30.62
ResNet50	Clean Backbone	0.10	24.16
	Backdoored Backbone	0.10	24.16
BiT-M-RN50	Clean Backbone	0.11	26.44
	Backdoored Backbone	0.11	26.44

TABLE IV
AVERAGE RUNTIME (GPU HOURS) PER EPOCH IN EACH PHASE.

Phase	Runtime	#Epochs per Loop
Prompt Learning	29.23s	10 or 100
Poisoning	1912.19s	10

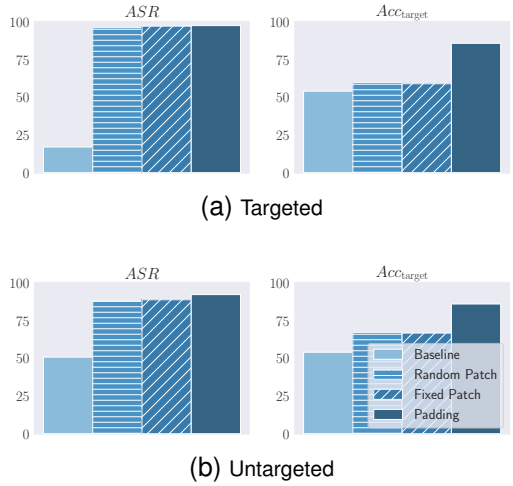


Fig. 4. Influence of the prompt designs.

confirms that the backdoor remains dormant, indicating that the malicious behavior is only activated when both the trigger and a learned downstream prompt are present.

Computational Cost. As BADBONE involves bi-level optimization, we report both memory usage and runtime (in GPU hours) for the target backdoor attack. We conduct experiments on a ResNet18 model trained on CIFAR10 using an NVIDIA A100 GPU (80 GB). As shown in Table IV, each epoch in the prompt learning phase requires approximately 30 seconds, whereas each epoch in the poisoning phase takes around 32 minutes. This indicates that prompt learning is computationally efficient, while the primary cost arises from backbone fine-tuning. Importantly, this cost is expected and reasonable, as fine-tuning the backbone over multiple epochs is a standard practice in existing backdoor attack methods. Therefore, our computational overhead is in line with prior work rather than introducing additional burden. For the full training process with two iterations, the total runtime is approximately 11.5 GPU hours. In terms of memory, BADBONE reaches a peak GPU usage of 15.51 GB, with a steady-state footprint of around 15.50 GB. This corresponds to an incremental allocation of approximately 14.97 GB above the warm idle baseline.

Summary. We show that the backdoored backbone model enables the adversary to achieve good attack performance in targeted and untargeted attacks. At the same time, both attacks are stealthy and can guarantee a competitive model utility on both the pre-training task and the target task.

C. Attack Transferability

Our attack also exhibits high transferability, allowing different prompt designs and out-of-distribution datasets. Here, we employ the ResNet18 model with the CIFAR-10 dataset.

Transfer to Different Prompt Designs. In the previous experiments, we use the padding prompt (see Figure 3 (c)) in the prompt learning process. We expect that, even if the victim conducts prompt learning using different forms of prompt, the adversarial goal is achieved. Using the same backdoored backbone model, we evaluate the attack performance using a different form of prompt, e.g., a fixed/random patch on the image (see Figure 3 (b)). The patch size remains the same, i.e., 30. Note that the backdoored backbone remains frozen; thus, the Acc_{pre} will not change. Based on Figure 4, we observe that the adversarial goal can still be achieved with different forms of prompts. For example, the adversary achieves a 97.28% *ASR* when using a fixed patch. The results also indicate that using different forms of prompts maintains utility well. The Acc_{target} of random/fixed patch is higher than that of the baseline.

Transfer to Different Datasets. Domain shift/adaptation remains an ongoing research topic in ML [40]. In our experiments, the shadow dataset has the same distribution as the downstream dataset used by the victim. We relax this assumption by using a shadow dataset with a similar distribution to the downstream dataset to evaluate our attack's generalizability. We thus introduce STL-10 [41], which contains 13,000 96×96 natural images, among which 5,000 images are partitioned for training while the remaining 8,000 images are for testing. For our experiment, we use STL-10 as our shadow dataset D to craft the backdoored backbone model and evaluate the target task on CIFAR-10. For the labels in the shadow dataset D, we re-order the class indices of STL-10 to align with CIFAR-10 class indices. To build the poisoning dataset, we randomly select 2,000 images from STL-10 as the clean dataset D_c and another 2,000 images to form the triggered dataset D_t. In the targeted attack, we map the label of the triggered images to class 1 (the "car" in STL-10), which corresponds to the target class "automobile" in CIFAR-10. In the prompt learning phase, we run 100 epochs with a learning rate lr_{prompt} of 100 in the first iteration and 20 epochs with lr_{prompt} =0.01 in the succeeding iterations. Experiments show that STL-10 requires a larger learning rate in prompt learning than CIFAR-10, which indicates that they have different distributions. In the poisoning phase, we optimize the backbone model for 10 epochs with a learning rate $lr_{\text{poisoning}}$ of 0.001 in every iteration. Note that lr_{prompt} is slightly different so that we can adapt to various datasets and backbone models. However, $lr_{\text{poisoning}}$ remains fixed at 0.001 throughout the paper. With the

TABLE V
INFLUENCE OF DIFFERENT DATASETS (STL-10). NUMBERS IN BRACKETS ARE BASELINES.

Attack	*ASR*/*MR*	Acc_{target}	Acc_{pre}
Targeted	97.64 (17.02)	54.16 (54.10)	66.37 (69.76)
Untargeted	74.74 (50.88)	46.58 (54.10)	66.11 (69.76)

TABLE VI
BADBONE-ATTACKED INSTAGRAM MODEL ON CIFAR-10.

	ASR	Acc_{target}
Baseline	4.55	62.10
Targeted	92.47	78.27

poisoned backbone model, we conduct prompt learning using CIFAR-10 as the downstream dataset and measure the attack performance on the prompted model. The learning rate used in the evaluation is 0.1. Results in Table V show that using STL-10 as the shadow dataset for backdooring the target task on the CIFAR-10 dataset can also attain effectiveness and utility goals. For example, the *ASR* of the targeted backdoored model is 97.64%, which has an 80.62% gain over the baseline. The Acc_{pre} of it is 66.37% and still at the same level as the baseline. After prompt learning, the Acc_{target} of the prompted model also reaches 54.16%, which is 0.06% higher than the baseline. In summary, using a shadow dataset with a similar distribution can achieve the adversarial goals in both attacks.

We further use UTKFace [42], a gender classification task (target label 1: female), to backdoor the model. We conduct prompt learning on CIFAR-10 with the UTKFace backdoored model. In this way, we aim to empirically observe how dramatically different shadow and downstream datasets may impact the attack performance. Our attack achieves a 79.26% ASR but with 11.50% of Acc_{target}. Note that STL-10 and CIFAR-10 contain natural images with some objectives, while UTKFace only contains faces and focuses on gender classification. We argue that our method is more transferable when the shadow dataset has similar categories and tasks, thereby preserving utility. This is a practical assumption, as model providers typically have some insight into the downstream applications their clients are targeting.

Transfer to Different Models. To show that our attack also applies to model architectures other than ResNet, we experimented on ResNeXt trained on 3.5B Instagram images (Instagram). For Instagram, we follow the default settings in the paper and use CIFAR-10 as the target dataset. Results in Table VI show our attack's effectiveness, where BADBONE achieves an ASR of 92.47% on the CIFAR-backdoored Instagram model. Intuitively, our algorithm should also work for the vision-language model CLIP, specifically the image encoder. However, as the pre-trained encoder is trained in a self-supervised way, the clean loss of the poisoning phase may be different. Thus, the clean loss becomes the contrastive loss, where we add the text encoder and keep it fixed. We leave this as future work.

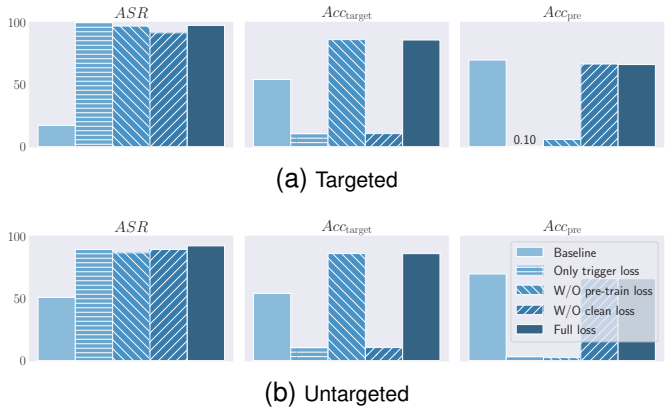


Fig. 5. Influence of the loss terms. "Full loss" represents the backdoored model with the poisoning loss $\ell_{\text{poisoning}}$ with $\alpha = 1$, $\beta = 1$, and $\gamma = 1$ in Equation 6 and the other three contain different loss terms in Equation 6. "Only trigger loss" means $\alpha = 0$, $\beta = 0$, and $\gamma = 1$, "W/O pre-train loss" is $\alpha = 0$, $\beta = 1$, and $\gamma = 1$ and "W/O clean loss" applies for $\alpha = 1$, $\beta = 0$, and $\gamma = 1$.

D. Ablation Study

Here, we investigate factors including loss terms, number of iterations, quantity of poisoned samples, trigger size and placement, target label, and flipping strategies. Both targeted and untargeted approaches are explored where applicable. For all experiments in this section, we employ the ResNet18 model with the CIFAR-10 dataset.

Loss Terms. We investigate the impact of different loss terms in Equation 6. Results are shown in Figure 5. We observe that weighing more on the trigger loss ℓ_{D_t} can boost the attack's effectiveness. For example, the *ASR* of the targeted backdoored model with only the trigger loss reaches 100.00%, achieving an 82.98% performance gain over the baseline. However, using only the trigger loss cannot satisfy our utility goal. For instance, with only the trigger loss, the Acc_{target} and Acc_{pre} of the targeted backdoored model drop to 10.48% and 3.13%, while the baselines are 54.10% and 69.76%, respectively. The results indicate that the pre-training loss ℓ_{D_p} and the clean loss ℓ_{D_c} are indispensable to maintain the utility of the backbone model on the pre-training task and the target task, respectively. This finding is further supported by the observation that when the pre-training loss ℓ_{D_p} is excluded, the targeted backdoored model can achieve an 86.02% Acc_{target} on the target task while only a 5.78% Acc_{pre} on the pre-training task. This highlights the importance of the pre-training loss in preserving the utility of the pre-training task. In addition, if the clean loss ℓ_{D_c} is excluded, with the targeted backdoor attack, the Acc_{pre} increases to 66.54% (the baseline is 69.76%), while the Acc_{target} drops to 10.50%, respectively. Our results demonstrate the effectiveness of the trigger loss for the attack and the importance of the other two losses in contributing to the utility. Further, Table VII shows additional results of different weight combinations among γ, β, and α for targeted attack, exhibiting similar conclusions.

Number of Iterations (N_{iters}). Figure 6 shows the performance of our two attacks with different numbers of iterations (N_{iters} in Algorithm 1) ranging from 0 to 4. When N_{iters} is 0, it represents our baseline. We find that the attacks perform well

TABLE VII
INFLUENCE OF DIFFERENT LOSS WEIGHTS.

Setting (γ:β:α)	ASR	Acc_{target}	Acc_{clean}
D_c only (0:1:0)	10.24	93.04	6.62
D_t only (1:0:0)	100.00	10.48	0.10
Weighted (1:0:5:0.5)	98.82	87.10	65.67

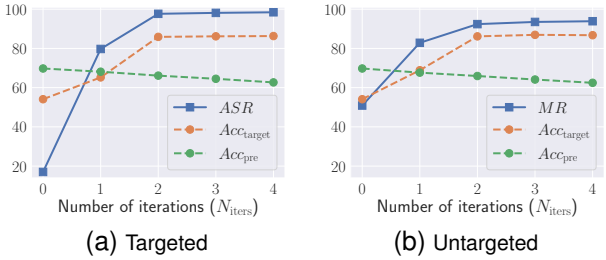


Fig. 6. Influence of the number of iterations.

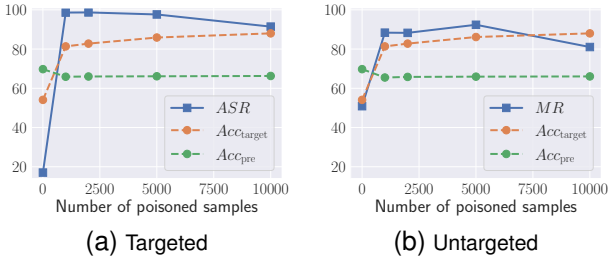


Fig. 7. Influence of the number of poisoned samples.

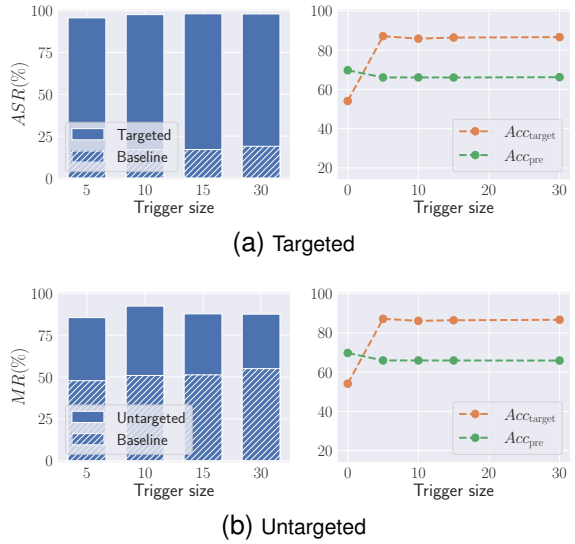


Fig. 8. Influence of the trigger size. For each attack, the left figure evaluates the attack performance, while the right illustrates the attack utility with different trigger sizes.

TABLE VIII
INFLUENCE OF DIFFERENT TRIGGER POSITIONS.

Attack	Position	*ASR*/*MR*	Acc_{target}	Acc_{pre}
Targeted	Center	99.04 (10.74)	86.68	66.17
	Right bottom	97.64 (17.02)	87.09	71.96
Untargeted	Center	88.60 (46.58)	86.68	65.98
	Right bottom	92.38 (50.88)	86.14	65.93

after only one iteration, which exemplifies the effectiveness of our attacks. For example, the *ASR* of the targeted backdoored model reaches 79.72% in the first iteration, 62.70% higher than the baseline. With additional iterations, the attack performance first increases and then becomes stable in general. We also observe that with more iterations, Acc_{target} increases, while Acc_{pre} slightly drops. Therefore, we choose N_{iters} =2 to balance between utility and effectiveness for our attacks.

Number of Poisoned Samples. In our main experiments, we randomly sample 5,000 images from the downstream datasets to add triggers, i.e., the triggered dataset D_t, and construct the trigger loss, and another 5,000 images, i.e., the clean dataset D_c, to construct the clean loss. To find out the influence of different numbers of poisoned samples on our two attacks, we randomly sample 1,000, 2,000, 5,000, and 10,000 images from CIFAR-10 and build the triggered dataset D_t. We keep the number of clean samples to be the same as that of the poisoned ones to maintain the balance between D_t and D_c.

Figure 7 demonstrates the performance of our two attacks with these four different numbers of poisoned samples. We find that our attacks can be characterized by an acceptable data resource requirement. For example, with 1,000 poisoned samples, the *ASR* of the targeted backdoored model can reach 98.60%, 81.58% higher than the baseline. Given more poisoned samples, the attack performance improves. But if there are too many poisoned samples, the attack performance may drop slightly. For example, with 10,000 poisoned samples, the *ASR* of the targeted backdoored model drops to 91.44%, and the *MR* of the untargeted backdoored model drops to 81.04%, respectively. The results indicate that injecting a proper number of poisoned samples is also important for the backdoor attack performance.

Trigger Size. We conduct both targeted and untargeted backdoor attacks with five different trigger sizes, i.e., 0, 5, 10, 15, and 30, to investigate the influence of the trigger size. The size of 0 represents the clean backbone model, i.e., the baseline. Note that all the triggers are at the same position, i.e., the bottom right corner. We compute the *ASR*/*MR* baselines for each size, as the baseline performance varies with different trigger sizes. For example, the baseline *ASR* for the trigger size of 5 is 13.46% while it is 17.02% when the trigger size is 10. Figure 8 shows that the performance of both attacks improves with larger trigger sizes. Take the targeted backdoor attack as an example; with larger trigger sizes, the *ASR* of the backdoored model increases to 98%, and the Acc_{pre} and Acc_{target} remain good. This reveals that a larger trigger can usually lead to a better backdoor attack result. However, if the trigger is too large, it is also easier to be detected.

Trigger Position. We also investigate if trigger position may affect the attack performance. To this end, we use the same trigger size, i.e., 10, and compare the performance of our two attacks with two different trigger positions, i.e., center and bottom right corner. Table VIII shows the performance of BADBONE with different trigger positions. The numbers in the brackets are the baselines, i.e., the *ASR*/*MR* of the clean

TABLE IX
INFLUENCE OF DIFFERENT TARGET LABELS.

Backbone Model	*ASR*	Acc_{target}	Acc_{pre}
ResNet18	96.90	86.56	66.10
ResNet50	98.62	90.98	71.87
BiT-M-RN50	98.88	93.46	72.04

TABLE X
INFLUENCE OF DIFFERENT LABEL FLIPPING STRATEGIES.

Strategy	*MR*	Acc_{target}	Acc_{pre}
Random flipping	65.06	81.76	65.88
Next flipping	92.38	86.14	65.93

model with a learned prompt on the triggered images with different trigger positions. We observe that backdoored models with triggers at the center achieve better attack performance over the baselines. For instance, with a trigger located at the center, the targeted backdoored model achieves a 99.04% *ASR*, achieving an 88.30% gain over the baseline. When the trigger is placed at the bottom right corner, the *ASR* of the backdoored model is 80.62%. For the untargeted backdoor attack, the *MR* with center triggers has a 42.02% gain over the baseline, while with the trigger at the bottom right corner, the backdoored model achieves a comparable 41.50% gain. The results show that both attacks are robust to different trigger positions.

Target Label. To show that different target labels do not affect the attack's effectiveness, we further evaluate our targeted attack targeting different labels. Specifically, we target "cat" in CIFAR-10 and the detailed results are shown in Table IX. The results show that our attack performance is not affected by different target labels. For example, the backdoored ResNet18 model on CIFAR-10 achieves 96.90% *ASR* and maintains its utility when targeting "cat" instead of "automobile." This further exemplifies the generalizability of our target attack.

Next Flipping vs. Random Flipping. To explore the influence of label-flipping strategies in untargeted backdoor attacks, we compare the attack performance between *next flipping* and *random flipping*. Table X shows the results. We observe that both label-flipping strategies can achieve effectiveness and utility goals. In particular, the next flipping generally leads to a higher attack performance than random flipping. The *MR* of the backdoored model by next flipping is 92.38%, even 27.32% higher than that by random flipping. Meanwhile, both strategies maintain a good utility. For instance, the Acc_{target} of the backdoored model with random flipping reaches 81.76%, which is 27.66% higher than the baseline. This is similar to the case of the next flipping strategy.

Summary. In summary, our ablation study demonstrates that all three loss terms are critical for balancing effectiveness and utility. While the attack is effective even with few poisoned samples and is robust to variations in trigger design and placement, a careful balance of optimization parameters is necessary to achieve the desired stealth and performance.

TABLE XI
NEURAL CLEANSE ANOMALY INDEX OF DIFFERENT BACKDOORED MODELS ON CIFAR-10. THE THRESHOLD IS 2.

Attack	ResNet18	ResNet50	BiT-M-RN50
Targeted	1.13	1.37	1.03
Untargeted	1.28	1.37	1.45

TABLE XII
MNTD CLASSIFICATION ACCURACY OF DIFFERENT BACKDOORED MODELS ON CIFAR-10.

Attack	ResNet18	ResNet50	BiT-M-RN50
Targeted	2%	2%	98%
Untargeted	6%	40%	100%

TABLE XIII
PERFORMANCE OF NAD-DEFENSED BACKDOORED BACKBONE MODELS ON CIFAR-10. NUMBERS IN BRACKETS SHOW THE ATTACK PERFORMANCE BEFORE THE DEFENSE.

Attack	Metrics	ResNet18	ResNet50	BiT-M-RN50
Targeted	*ASR*	96.58 (97.64)	99.10 (98.66)	96.00 (98.92)
	Acc_{target}	76.70 (85.88)	81.72 (91.04)	88.28 (93.38)
	Acc_{pre}	61.46 (66.12)	65.40 (72.03)	64.56 (72.11)
Untargeted	*MR*	91.28 (92.38)	86.72 (94.88)	95.98 (97.74)
	Acc_{target}	72.76 (86.14)	85.06 (91.92)	86.76 (94.02)
	Acc_{pre}	59.31 (65.93)	66.604 (71.81)	64.72 (71.94)

VI. DEFENSE

We evaluate six state-of-the-art defenses against backdoor attacks. Neural Cleanse [43], ABS [44], and MNTD [45] for detecting backdoored models. We treat the prompted model $F_{\theta,\phi}$ (see Section II) as the defense target and check if the prompted model is backdoored. NAD [46], CLP [47], and D-BR [48] aim to erase the backdoor trigger of a backdoored model. Recall that the victim freely controls the visual prompt generation process; we target the backdoored backbone models f_θ. We measure our targeted and untargeted backdoored models on CIFAR-10 (six models in total) on these defenses.

Neural Cleanse [43] uses reverse engineering to detect whether a model is backdoored. It generates triggers for each output label and predicts if any of them is a backdoor trigger by computing the Anomaly Index. If the index is above the threshold of 2, the model is predicted to be backdoored. We measure our targeted and untargeted backdoored models on CIFAR-10 (six models in total) via Neural Cleanse. For each backdoored model, we reverse engineer a trigger for each label and compute the Anomaly Index following the default settings. Results shown in Table XI indicate that all six models are predicted as clean models. For instance, the Anomaly Index of the targeted backdoored BiT-M-RN50 model on CIFAR-10 is only 1.03, which is lower than the threshold. We therefore conclude that Neural Cleanse cannot mitigate our attacks.

ABS [44] is also based on reverse engineering. It first analyzes the model's inner neuron behaviors, i.e., activation status, by introducing varied levels of stimulation based on a set of clean images. Then it reverses the backdoor trigger to determine whether a neuron is truly compromised by leveraging

TABLE XIV
TARGETED ATTACK PERFORMANCE AND UTILITY OF DEFENSED RESNET18 MODEL.

	ASR	Acc_{target}
Targeted	99.22	77.52
CLP Defensed	13.52	52.10
D-BR Defensed	99.96	13.94

the results of the stimulation analysis. The authors of ABS acknowledge that the current version of ABS only works on CIFAR-10 [44]. Thus, we use the supported set containing 50 CIFAR-10 seed images provided by the authors and run ABS against the six models, i.e., the targeted and untargeted backdoored models on CIFAR-10. Results show that ABS predicts all six backdoored models to be clean. We find that ABS cannot defend against our attacks as well.

MNTD [45] uses meta-learning to train a classifier to score the likelihood of a model being backdoored. Following previous studies [30], [49], we build 200 benign shadow models and 200 backdoored shadow models using the CIFAR-10 dataset. The backdoored shadow models are trained with different triggers via jumbo learning. Then, we train 50 sequential meta-classifiers based on the feature representations of these 400 shadow models and report their classification accuracy, representing the likelihood that the model is backdoored. Following Salem et al. [49], any model that scores higher than the threshold 0.0 is deemed backdoored. For example, given an input model, if only 5 out of 50 meta-classifiers output scores higher than 0.0, the classification accuracy is 10%, i.e., $\frac{5}{50} \times 100\%$. Results in Table XII show that the backdoored ResNet18 and ResNet50 models are less likely to be detected by MNTD, while the backdoored BiT-M-RN50 models have a high probability of being detected.

NAD [46] mitigates backdoor attacks by erasing backdoor triggers from a backdoored model via fine-tuning on clean data and attention distillation using a teacher-student framework. As the victim controls the full process of prompt learning based on the frozen backbone model, it is natural to use NAD to defend against the backbone model f_θ instead of the prompted model $F_{\theta,\phi}$. Specifically, we fine-tune the backdoored model f_θ on ImageNet-1k (50k) for 5 epochs with a learning rate of 0.001 and take it as the teacher model. Then, we prune the backdoored model using attention distillation for 10 epochs with the same learning rate. We evaluate the attack performance and the utility based on the NAD-defensed backbone models. The results are shown in Table XIII. The numbers in the bracket demonstrate the utility of the backdoored model before conducting NAD defense. We observe that although the utility of the NAD-defensed models generally maintains at the same level as the backdoored model without defense, our attack still achieves a compatible attack success rate after defense. For example, the Acc_{target} of the backdoored BiT-M-RN50 is 93.3%, while that of the NAD-defensed model is 88.28%. The utility of the NAD-defensed model on the pre-training task slightly decreases by fine-tuning and pruning in the NAD defense process. We consequently conclude that our attack is not successfully mitigated by NAD.

CLP [47] and **D-BR** [48] also aim to remove the backdoor from the model. We target the targeted backdoored ResNet18 on the CIFAR10 dataset and then employ prompt learning on the defensed model to evaluate attack effectiveness and utility. Results in Table XIV show that while CLP successfully mitigated the attack, it failed to maintain the utility of the target task. Conversely, D-BR not only failed to remove the backdoor but also negatively affected the utility of the target task. In conclusion, both defenses are ineffective at mitigating our attack while maintaining utility.

Summary. We evaluate our approach against six state-of-the-art model-level defenses, including Neural Cleanse, ABS, MNTD, NAD, CLP, and D-BR. Results demonstrate that most of our backdoored models can successfully bypass the defenses. This is because most defenses rely on detecting abnormal behavior under trigger-only or global perturbations, whereas our co-activation mechanism keeps the backdoor dormant unless both the learned prompt and trigger are present. That is, our BADBONE is activated only when the learned prompt and the trigger are jointly present. Possible defense directions could be prompt-agnostic behavioral consistency checks, decoupling-based tests for prompt-only/trigger-only activation, and cross-task anomaly analysis, which are left as future work.

VII. RELATED WORK

Prompt Learning in Natural Language Processing. Prompt learning, originating from natural language processing (NLP) [8], involves creating task-specific templates to rephrase input text (e.g., "`[INPUT]` Overall, I felt `[MASK]`.") and mapping answers to desired labels using pre-trained language models (e.g., "good/terrible" for positive/negative). Much effort has been taken in prompt engineering to design an effective textual prompt, i.e., discrete prompt, for the target downstream task [50], [51]. However, hand-crafting the right prompt requires domain expertise and substantial trial-and-error effort. Recent approaches [11], [19], [52]–[55] have aimed to address this issue by learning a "soft prompt," i.e., a continuous vector, through backpropagation while freezing the backbone model's parameters. These include prompt tuning [19] and hybrid tuning [54].

Prompt Learning in Computer Vision. Visual prompt learning [9], [10], [12]–[18] has emerged as a natural extension of prompt learning in NLP. Visual prompt learning can be broadly categorized into two types. The first category focuses on learning an input perturbation to shift the downstream data to the distribution used to train the pre-trained model [10], [12], [18], [56]. The second category aims to inject a small number of learnable parameters into the vision transformer [9], [16], [57], [58]. However, these approaches require changes at the model level, which may raise security concerns among users. In our paper, we thus focus on visual prompt learning from the first category.

Backdoor Attacks and Their Defense. Backdoor attack is a prominent security threat to machine learning models [23], [30], [31], [49], [59], [60]. It implants a trigger [49], [61]–[63]

into the target model during the training process, consequently causing the model to misclassify any input with the added trigger pattern to a specific target label. Previous research has exemplified that backdoor attack can impact many real-world applications, including face recognition [59], image classification [64], [65], video processing [66], NLP [67], [68], and, most recently, pre-trained encoders [30], [32]. Meanwhile, defense against backdoor attacks can be roughly divided into three detection levels, i.e., model-level [43]–[48], [69]–[71], input-level [72]–[74], and dataset-level (i.e., if the training dataset is backdoored) [75], [76].

VIII. Conclusion

In this work, we explore the backdoor attack against an emerging machine learning paradigm, namely prompt learning. Existing backdoor attacks against prompt learning focus on injecting the backdoor into the prompt [24], [25]. We are the first to propose BadBone, the targeted/untargeted backdoor attack against the backbone model in visual prompt learning. We develop a bi-level optimization-based approach to inject a backdoor into the backbone model, where the backdoor can be activated when prompt learning is performed for the target task. We evaluate our BadBone using three benchmark datasets on three different model frameworks, exhibiting its effectiveness, stealthiness, and transferability. Experiments show that both attacks can achieve their adversarial goal while preserving the model's utility. We further show that most of our backdoored models can bypass six state-of-the-art backdoor defense mechanisms, requesting new defenses for our backdoor attacks.

Limitations

Our attack conditions on two factors: knowledge of the downstream task and adherence to the label mapping scheme by the victim. Regarding the first condition, it is typical in transfer learning, such as visual prompt learning, for the model provider to collaborate with the victim to select the most suitable model. Consequently, attackers may gain insight into the downstream task. Experiments in Section V-C further show that our attack can be generalized to out-of-distribution datasets of a downstream task. In this sense, our attack also supports scenarios when the attacker cannot obtain the training data for the downstream task. Concerning the second condition, we acknowledge that our attack may not yield optimal outcomes if the victim does not adhere to the predefined label mapping scheme. However, we emphasize that our attack targets victims lacking advanced machine learning expertise, as outlined in the threat model (Section III). In such cases, it is plausible that they will follow the mapping scheme recommended by the model provider rather than experimenting with alternative schemes.

Impact Statement

Prompt learning has emerged as a trending topic in machine learning, gaining significant attention due to its time and computational efficiency, particularly in industrial applications. However, this paradigm introduces critical security risks that must be addressed. Our work highlights a potential threat: the backdooring of backbone models within the prompt learning framework. Given that our attack method is both simple and highly effective, it poses a significant danger if exploited by malicious actors. By exposing this vulnerability, our research contributes to the development of more robust defenses in real-world scenarios. Furthermore, we aim to advance the attack-defense cycle by releasing our code publicly for reproducibility, ultimately facilitating the development of effective defenses. The code is released under the MIT license. In addition, the repository README includes a responsible-use statement clarifying that the release is intended for reproducibility and defensive research, and that users are expected not to use the code to develop, deploy, distribute, or facilitate malicious models, attacks, unauthorized access, evasion, or other harmful activity.

References

[1] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," in *International Conference on Learning Representations (ICLR)*, 2021.

[2] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning Transferable Visual Models From Natural Language Supervision," in *International Conference on Machine Learning (ICML)*. PMLR, 2021, pp. 8748–8763.

[3] V. Kieuvongngam, B. Tan, and Y. Niu, "Automatic Text Summarization of COVID-19 Medical Research Articles using BERT and GPT-2," *CoRR abs/2006.01997*, 2020.

[4] R. Mokady, A. Hertz, and A. H. Bermano, "ClipCap: CLIP Prefix for Image Captioning," *CoRR abs/2111.09734*, 2021.

[5] O. Patashnik, Z. Wu, E. Shechtman, D. Cohen-Or, and D. Lischinski, "StyleCLIP: Text-Driven Manipulation of StyleGAN Imagery," *CoRR abs/2103.17249*, 2021.

[6] J. Dodge, G. Ilharco, R. Schwartz, A. Farhadi, H. Hajishirzi, and N. A. Smith, "Fine-Tuning Pretrained Language Models: Weight Initializations, Data Orders, and Early Stopping," *CoRR abs/2002.06305*, 2020.

[7] H. Li, P. Chaudhari, H. Yang, M. Lam, A. Ravichandran, R. Bhotika, and S. Soatto, "Rethinking the Hyperparameters for Fine-tuning," in *International Conference on Learning Representations (ICLR)*, 2020.

[8] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, "Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing," *ACM Computing Surveys*, 2023.

[9] M. Jia, L. Tang, B. Chen, C. Cardie, S. J. Belongie, B. Hariharan, and S. Lim, "Visual Prompt Tuning," in *European Conference on Computer Vision (ECCV)*. Springer, 2022, pp. 709–727.

[10] H. Bahng, A. Jahanian, S. Sankaranarayanan, and P. Isola, "Exploring Visual Prompts for Adapting Large-scale Models," *CoRR abs/2203.17274*, 2022.

[11] X. Han, W. Zhao, N. Ding, Z. Liu, and M. Sun, "PTR: Prompt Tuning with Rules for Text Classification," *AI Open*, 2022.

[12] G. F. Elsayed, I. J. Goodfellow, and J. Sohl-Dickstein, "Adversarial Reprogramming of Neural Networks," in *International Conference on Learning Representations (ICLR)*, 2019.

[13] M. Tsimpoukelli, J. Menick, S. Cabi, S. M. A. Eslami, O. Vinyals, and F. Hill, "Multimodal Few-Shot Learning with Frozen Language Models," in *Annual Conference on Neural Information Processing Systems (NeurIPS)*. NeurIPS, 2021, pp. 200–212.

[14] A. Chen, Y. Yao, P. Chen, Y. Zhang, and S. Liu, "Understanding and Improving Visual Prompting: A Label-Mapping Perspective," *CoRR abs/2211.11635*, 2022.

[15] Y. Zang, W. Li, K. Zhou, C. Huang, and C. C. Loy, "Unified Vision and Language Prompt Learning," *CoRR abs/2210.07225*, 2022.

[16] A. Bar, Y. Gandelsman, T. Darrell, A. Globerson, and A. A. Efros, "Visual Prompting via Image Inpainting," in *Annual Conference on Neural Information Processing Systems (NeurIPS)*. NeurIPS, 2022, pp. 25 005–25 017.

[17] Z. Yang, Z. Sha, M. Backes, and Y. Zhang, "From Visual Prompt Learning to Zero-Shot Transfer: Mapping Is All You Need," *CoRR abs/2303.05266*, 2023.

[18] W. Liu, X. Shen, C. Pun, and X. Cun, "Explicit Visual Prompting for Low-Level Structure Segmentations," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2023.

[19] X. L. Li and P. Liang, "Prefix-Tuning: Optimizing Continuous Prompts for Generation," in *Annual Meeting of the Association for Computational Linguistics and International Joint Conference on Natural Language Processing (ACL/IJCNLP)*. ACL, 2021, pp. 4582–4597.

[20] G. Qin and J. Eisner, "Learning How to Ask: Querying LMs with Mixtures of Soft Prompts," in *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*. ACL, 2021, pp. 5203–5212.

[21] "Landing ai," https://landing.ai/.

[22] Amazon, "Foundational vision models and visual prompt engineering for autonomous driving applications ," https://aws.amazon.com/blogs/machine-learning/foundational-vision-models-and-visual-prompt-engineering-for-autonomous-driving-applications/, 2023.

[23] T. Gu, B. Dolan-Gavitt, and S. Grag, "Badnets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain," *CoRR abs/1708.06733*, 2017.

[24] W. Du, Y. Zhao, B. Li, G. Liu, and S. Wang, "PPT: Backdoor Attacks on Pre-trained Models via Poisoned Prompt Tuning," in *International Joint Conferences on Artifical Intelligence (IJCAI)*. IJCAI, 2022, pp. 680–686.

[25] H. Huang, Z. Zhao, M. Backes, Y. Shen, and Y. Zhang, "Prompt Backdoors in Visual Prompt Learning," *CoRR abs/2310.07632*, 2023.

[26] S. Dempe, *Foundations of bilevel programming*. Springer Science & Business Media, 2002.

[27] "CIFAR," https://www.cs.toronto.edu/~kriz/cifar.html.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016, pp. 770–778.

[29] C. Song, T. Ristenpart, and V. Shmatikov, "Machine Learning Models that Remember Too Much," in *ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, 2017, pp. 587–601.

[30] J. Jia, Y. Liu, and N. Z. Gong, "BadEncoder: Backdoor Attacks to Pre-trained Encoders in Self-Supervised Learning," in *IEEE Symposium on Security and Privacy (S&P)*. IEEE, 2022.

[31] N. Carlini and A. Terzis, "Poisoning and Backdooring Contrastive Learning," in *International Conference on Learning Representations (ICLR)*, 2022.

[32] L. Shen, S. Ji, X. Zhang, J. Li, J. Chen, J. Shi, C. Fang, J. Yin, and T. Wang, "Backdoor Pre-trained Models Can Transfer to All," in *ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, 2021, pp. 3141–3158.

[33] A. Saha, A. Tejankar, S. A. Koohpayegani, and H. Pirsiavash, "Backdoor Attacks on Self-Supervised Learning," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2022, pp. 13 327–13 336.

[34] G. Liu, W. Zhang, X. Li, K. Fan, and S. Yu, "VulnerGAN: A Backdoor Attack Through Vulnerability Amplification Against Machine Learning-based Network Intrusion Detection Systems," *Science China Information Sciences*, 2022.

[35] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2009, pp. 248–255.

[36] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading Digits in Natural Images with Unsupervised Feature Learning," in *Annual Conference on Neural Information Processing Systems (NIPS)*. NIPS, 2011.

[37] P. Helber, B. Bischke, A. Dengel, and D. 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*, 2019.

[38] A. Kolesnikov, L. Beyer, X. Zhai, J. Puigcerver, J. Yung, S. Gelly, and N. Houlsby, "Big Transfer (BiT): General Visual Representation Learning," in *European Conference on Computer Vision (ECCV)*. Springer, 2020, pp. 491–507.

[39] I. Loshchilov and F. Hutter, "SGDR: Stochastic Gradient Descent with Warm Restarts," in *International Conference on Learning Representations (ICLR)*, 2017.

[40] S. J. Pan and Q. Yang, "A Survey on Transfer Learning," *IEEE Transactions on Knowledge and Data Engineering*, 2009.

[41] A. Coates, A. Y. Ng, and H. Lee, "An Analysis of Single-Layer Networks in Unsupervised Feature Learning," in *International Conference on Artificial Intelligence and Statistics (AISTATS)*. JMLR, 2011, pp. 215–223.

[42] Z. Zhang, Y. Song, and H. Qi, "Age Progression/Regression by Conditional Adversarial Autoencoder," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 4352–4360.

[43] B. Wang, Y. Yao, S. Shan, H. Li, B. Viswanath, H. Zheng, and B. Y. Zhao, "Neural Cleanse: Identifying and Mitigating Backdoor Attacks in Neural Networks," in *IEEE Symposium on Security and Privacy (S&P)*. IEEE, 2019, pp. 707–723.

[44] Y. Liu, W.-C. Lee, G. Tao, S. Ma, Y. Aafer, and X. Zhang, "ABS: Scanning Neural Networks for Back-Doors by Artificial Brain Stimulation," in *ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, 2019, pp. 1265–1282.

[45] X. Xu, Q. Wang, H. Li, N. Borisov, C. A. Gunter, and B. Li, "Detecting AI Trojans Using Meta Neural Analysis," in *IEEE Symposium on Security and Privacy (S&P)*. IEEE, 2021.

[46] Y. Li, X. Lyu, N. Koren, L. Lyu, B. Li, and X. Ma, "Neural Attention Distillation: Erasing Backdoor Triggers from Deep Neural Networks," in *International Conference on Learning Representations (ICLR)*, 2021.

[47] R. Zheng, R. Tang, J. Li, and L. Liu, "Data-Free Backdoor Removal Based on Channel Lipschitzness," in *European Conference on Computer Vision (ECCV)*. Springer, 2022, pp. 175–191.

[48] W. Chen, B. Wu, and H. Wang, "Effective Backdoor Defense by Exploiting Sensitivity of Poisoned Samples," in *Annual Conference on Neural Information Processing Systems (NeurIPS)*. NeurIPS, 2022.

[49] A. Salem, R. Wen, M. Backes, S. Ma, and Y. Zhang, "Dynamic Backdoor Attacks Against Machine Learning Models," in *IEEE European Symposium on Security and Privacy (Euro S&P)*. IEEE, 2022, pp. 703–718.

[50] Z. Jiang, F. F. Xu, J. Araki, and G. Neubig, "How Can We Know What Language Models Know," *Transactions of the Association for Computational Linguistics*, 2020.

[51] W. Yuan, G. Neubig, and P. Liu, "BARTScore: Evaluating Generated Text as Text Generation," in *Annual Conference on Neural Information Processing Systems (NeurIPS)*. NeurIPS, 2021, pp. 27 263–27 277.

[52] K. Hambardzumyan, H. Khachatrian, and J. May, "WARP: Word-level Adversarial ReProgramming," in *Annual Meeting of the Association for Computational Linguistics and International Joint Conference on Natural Language Processing (ACL/IJCNLP)*. ACL, 2021, pp. 4921–4933.

[53] B. Lester, R. Al-Rfou, and N. Constant, "The Power of Scale for Parameter-Efficient Prompt Tuning," in *Conference on Empirical Methods in Natural Language Processing (EMNLP)*. ACL, 2021, pp. 3045–3059.

[54] X. Liu, Y. Zheng, Z. Du, M. Ding, Y. Qian, Z. Yang, and J. Tang, "GPT Understands, Too," *CoRR abs/2103.10385*, 2021.

[55] E. Ben-David, N. Oved, and R. Reichart, "PADA: Example-based Prompt Learning for on-the-fly Adaptation to Unseen Domains," *Transactions of the Association for Computational Linguistics*, 2022.

[56] Q. Huang, X. Dong, D. Chen, W. Zhang, F. Wang, G. Hua, and N. Yu, "Diversity-Aware Meta Visual Prompting," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2023.

[57] M. Kim, H. Kim, and Y. M. Ro, "Prompt Tuning of Deep Neural Networks for Speaker-adaptive Visual Speech Recognition," *CoRR abs/2302.08102*, 2023.

[58] Y. Xing, Q. Wu, D. Cheng, S. Zhang, G. Liang, P. Wang, and Y. Zhang, "Dual Modality Prompt Tuning for Vision-Language Pre-Trained Model," *CoRR abs/2208.08340*, 2023.

[59] X. Chen, C. Liu, B. Li, K. Lu, and D. Song, "Targeted Backdoor Attacks on Deep Learning Systems Using Data Poisoning," *CoRR abs/1712.05526*, 2017.

[60] Y. Liu, S. Ma, Y. Aafer, W.-C. Lee, J. Zhai, W. Wang, and X. Zhang, "Trojaning Attack on Neural Networks," in *Network and Distributed System Security Symposium (NDSS)*. Internet Society, 2018.

[61] Y. Zeng, W. Park, Z. M. Mao, and R. Jia, "Rethinking the Backdoor Attacks' Triggers: A Frequency Perspective," in *IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2021, pp. 16 453–16 461.

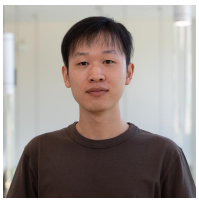
[62] Y. Li, Y. Li, B. Wu, L. Li, R. He, and S. Lyu, "Invisible Backdoor Attack with Sample-Specific Triggers," in *IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2021, pp. 16 443–16 452.

[63] A. Salem, M. Backes, and Y. Zhang, "Don't Trigger Me! A Triggerless Backdoor Attack Against Deep Neural Networks," *CoRR abs/2010.03282*, 2020.

[64] Y. Yao, H. Li, H. Zheng, and B. Y. Zhao, "Latent Backdoor Attacks on Deep Neural Networks," in *ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, 2019, pp. 2041–2055.
[65] A. Saha, A. Subramanya, and H. Pirsiavash, "Hidden Trigger Backdoor Attacks," in *AAAI Conference on Artificial Intelligence (AAAI)*. AAAI, 2020, pp. 11 957–11 965.
[66] S. Zhao, X. Ma, X. Zheng, J. Bailey, J. Chen, and Y.-G. Jiang, "Clean-Label Backdoor Attacks on Video Recognition Models," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2020, pp. 14 443–144 528.
[67] X. Chen, A. Salem, M. Backes, S. Ma, Q. Shen, Z. Wu, and Y. Zhang, "BadNL: Backdoor Attacks Against NLP Models with Semantic-preserving Improvements," in *Annual Computer Security Applications Conference (ACSAC)*. ACSAC, 2021, pp. 554–569.
[68] R. Schuster, C. Song, E. Tromer, and V. Shmatikov, "You Autocomplete Me: Poisoning Vulnerabilities in Neural Code Completion," *CoRR abs/2007.02220*, 2020.
[69] H. Chen, C. Fu, J. Zhao, and F. Koushanfar, "DeepInspect: A Black-box Trojan Detection and Mitigation Framework for Deep Neural Networks," in *International Joint Conferences on Artifical Intelligence (IJCAI)*. IJCAI, 2019, pp. 4658–4664.
[70] X. Huang, M. Alzantot, and M. B. Srivastava, "NeuronInspect: Detecting Backdoors in Neural Networks via Output Explanations," *CoRR abs/1911.07399*, 2019.
[71] W. Guo, L. Wang, X. Xing, M. Du, and D. Song, "TABOR: A Highly Accurate Approach to Inspecting and Restoring Trojan Backdoors in AI Systems," *CoRR abs/1908.01763*, 2019.
[72] S. J. Cho, T. J. Jun, B. Oh, and D. Kim, "DAPAS : Denoising Autoencoder to Prevent Adversarial attack in Semantic Segmentation," in *International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020, pp. 1–8.
[73] H. Kwon, "Defending Deep Neural Networks against Backdoor Attack by Using De-trigger Autoencoder," *IEEE Access*, 2021.
[74] Y. Gao, C. Xu, D. Wang, S. Chen, D. C. Ranasinghe, and S. Nepal, "STRIP: A Defence Against Trojan Attacks on Deep Neural Networks," in *Annual Computer Security Applications Conference (ACSAC)*. ACM, 2019, pp. 113–125.
[75] B. Tran, J. Li, and A. Madry, "Spectral Signatures in Backdoor Attacks," in *Annual Conference on Neural Information Processing Systems (NeurIPS)*. NeurIPS, 2018, pp. 8011–8021.
[76] J. Hayase, W. Kong, R. Somani, and S. Oh, "SPECTRE: Defending Against Backdoor Attacks Using Robust Statistics," in *International Conference on Machine Learning (ICML)*. JMLR, 2021.

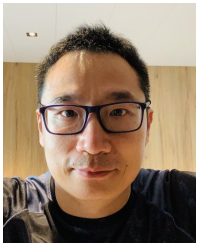
BIOGRAPHY SECTION

Ziqing Yang is a Ph.D. candidate at CISPA Helmholtz Center for Information Security. Before joining CISPA, she received her Bachelor's degree from Peking University in 2020. She primarily focuses on trustworthy machine learning, especially the security and privacy risks in generative models. Her work has appeared in leading venues such as USENIX Security, ICML, ICCV, ACL, and EMNLP.



Rui Wen is an Assistant Professor at the Institute of Science Tokyo. He earned his Ph.D. from the CISPA Helmholtz Center for Information Security. His research focuses on trustworthy machine learning, with particular emphasis on the role of data in adversarial machine learning. His work has appeared in leading venues including CCS, NDSS, and USENIX Security.

Xinlei He (Member, IEEE) is a research fellow at Wuhan University. He obtained his Ph.D. from CISPA Helmholtz Center for Information Security. His research lies in the domain of trustworthy machine learning, with a special focus on privacy, security, and accountability issues stemming from machine learning paradigms. He has published multiple papers in top-tier conferences/journals such as IEEE S&P, ACM CCS, USENIX Security, and NDSS. He served as the TPC member of IEEE S&P, IEEE Euro S&P, ASIACCS, ACSAC, IEEE ICDCS, and ESORICS. He has received the Norton Graduate Fellowship 2022 (2 people around the world), the Best Paper Award at LAMPS'24, and the Distinguished Poster Award at NDSS'25.



Yun Shen is a cybersecurity and machine learning researcher focused on adversarial machine learning, data privacy, and the security and safety of large language models (LLMs) and other generative models. His work appears in venues such as USENIX Security, IEEE S&P, CCS, and NDSS.

Michael Backes (Fellow, IEEE) is the founding director and CEO of the CISPA Helmholtz Center for Information Security. He focuses primarily on trustworthy machine learning methods, novel approaches for protecting personal data, and universal software and system security solutions. He has published over 300 peer-reviewed and highly cited publications in renowned international journals and conference proceedings. His research has earned him internationally renowned scientific awards and honors, particularly the Karl Heinz Beckurts Prize, the Caspar Bowden Privacy Award, the IEEE and ACM Fellowship, and various Career Awards and Best Paper Awards. He holds an honorary doctorate from the Université de Lorraine and is an honorary citizen and future ambassador of the city of St. Ingbert, Germany.

Yang Zhang (Member, IEEE) is a faculty member at CISPA Helmholtz Center for Information Security. His research concentrates on trustworthy machine learning. Moreover, he works on measuring and understanding misinformation and unsafe content like hateful memes on the Internet. Over the years, he has published multiple papers at top venues in computer science, including CCS, NDSS, Oakland, and USENIX Security. His work has received the NDSS 2019 Distinguished Paper Award and the CCS 2022 Best Paper Award runner-up.