Backdoor Attack in Prompt-Based Continual Learning

Trang Nguyen[◇] Anh Nguyen[◇] Nhat Ho[†]

The University of Texas at Austin[†]
VinAI Research[◇]

May 23, 2024

Abstract

Prompt-based approaches offer a cutting-edge solution to data privacy issues in continual learning, particularly in scenarios involving multiple data suppliers where long-term storage of private user data is prohibited. Despite delivering state-of-the-art performance, its impressive remembering capability can become a double-edged sword, raising security concerns as it might inadvertently retain poisoned knowledge injected during learning from private user data. Following this insight, in this paper, we expose continual learning to a potential threat: backdoor attack, which drives the model to follow a desired adversarial target whenever a specific trigger is present while still performing normally on clean samples. We highlight three critical challenges in executing backdoor attacks on incremental learners and propose corresponding solutions: (1) *Transferability*: We employ a surrogate dataset and manipulate prompt selection to transfer backdoor knowledge to data from other suppliers; (2) *Resiliency*: We simulate static and dynamic states of the victim to ensure the backdoor trigger remains robust during intense incremental learning processes; and (3) *Authenticity*: We apply binary cross-entropy loss as an anti-cheating factor to prevent the backdoor trigger from devolving into adversarial noise. Extensive experiments across various benchmark datasets and continual learners validate our continual backdoor framework, achieving up to 100% attack success rate, with further ablation studies confirming our contributions' effectiveness.

1 Introduction

The adaptability of human learning to absorb new knowledge without forgetting previously acquired information remains a significant challenge for machine learning models. Continual learning (CL) endeavors to narrow this chasm by guiding models to sequentially learn new tasks while maintaining high performance on earlier ones. An outstanding solution to CL is the prompt-based approach [45, 57, 58, 55, 40], which leverages the power of pre-trained models and employs a set of trainable prompts for flexible model instruction, accommodating data from various tasks. Thanks to its ability to remember without storing a memory buffer, prompt-based CL methods are particularly suitable for scenarios prioritizing data privacy, such as those involving multiple data suppliers.

Nonetheless, such promising results can inadvertently become vulnerabilities, exposing CL to security threats. Indeed, while CL methods effectively address catastrophic forgetting by preserving and incorporating previously acquired knowledge, they may also unwittingly retain knowledge compromised by adversarial actions. These threats become even more formidable in the multi-data supplier scenario of prompt-based approaches, where the supplied data might contain hidden harmful information.

One potential threat is backdoor attack, which manipulates neural networks to exhibit the attacker's desired behavior when the input contains a specific backdoor trigger. Typically, adversaries

poison a small portion of the training data, causing models trained on this data to misclassify any images with the triggers as a given target class while performing normally on clean samples. This makes the attack less likely to be suspected by the victim learner. As backdoor attacks pose such dangerous threats, increasingly sophisticated methods are being introduced. These include black-box scenarios where the attacker has no information about the model and learning procedure [42, 46, 48], or data-constrained cases where adversaries control only a fragment of the training data [64, 30]. With high efficacy, even in these challenging situations, backdoor attacks are particularly threatening in multi-data supplier scenarios. In spite of significant attention in various tasks and areas such as computer vision [48, 31, 36, 13, 12, 37], large language models and natural language processing [5, 28], point clouds [60, 59, 25], federated learning [61, 54, 65, 11], and more, targeted black-box backdoor attacks have not been thoroughly explored in continual learning.

Challenges Despite holding such potential danger for CL, extending backdoor attacks to the incremental setting is non-trivial. Firstly, in the multi-supplier setting where the victim gathers data from different sources, the attacker lacks information about the actual data distribution used to train the victim model. Consequently, *generalizing backdoor knowledge to be transferable to unknown data* poses the first challenge that our continual backdoor approach must confront. The second challenge arises from the vulnerability of backdoor attacks during fine-tuning. Recent studies [44, 35] have highlighted the tendency for backdoor knowledge to be removed when the victim fine-tunes the poisoned model on a small and clean dataset. This issue is exacerbated in continual learning, where the *victim model undergoes incremental training* as new data from various sources arrive. The final challenge involves the backdoor trigger's proneness to turn into adversarial noise. Huynh et al. [18] observed that the trigger, when optimized using a surrogate model, may *transform into an adversarial perturbation*, driving the clean model to follow desired adversarial targets even in the absence of any prior backdoor attacks. Since conventional adversarial defenses can mitigate such adversarial noise, preempting this behavior is crucial to strengthen the resilience of the backdoor trigger.

Contributions In response to these shortcomings, we propose a continual backdoor framework that satisfies three key properties: ***transferability*** *to unknown data*, ***resilience*** *to incremental learning procedures*, and ***authenticity*** *to avoid becoming adversarial noise*. Initially, we leverage the natural label mapping characteristic of visual prompting, thereby approaching the data poisoning issue from the perspective of prompt selection. This approach allows our backdoor trigger to be generalized to any victim data distribution. Next, we robustify the backdoor trigger by aligning the optimization process with the continuously changing states of the incremental learner, thus ensuring the effectiveness of the backdoor trigger when the model is trained on new incoming clean data. Finally, we reconsider the choice of loss function for trigger optimization. We observe that the commonly used softmax function with cross-entropy introduces bias towards the target class, pushing its score excessively high and leading to the adversarial noise problem. Building on this observation, we propose adopting binary cross-entropy (BCE) with sigmoid function to mitigate this issue, thereby eliminating the dependency of trigger optimization on other classes and preventing cheating behavior.

By integrating the components above, our framework, termed backdoor-**A**ttack **O**n **P**rompt-based CL (**AOP**), successfully backdoor-attacks continual learners, achieving an Attack Success Rate (ASR) of up to 100%. Our contributions are three-fold and can be summarized as follows:

1. We expose prompt-based CL to backdoor attacks. Our approach follows strong assumptions, with black-box, clean-label, and constrained-data setting;

2. We highlight three key challenges that our continual backdoor framework must address: ensuring transferability to unknown data in prompt tuning, preventing the catastrophic forgetting of backdoor knowledge, and mitigating the tendency to generate adversarial noise due to biases.

Motivated by these challenges, we propose a novel continual backdoor framework comprising three main components: utilizing a surrogate dataset to manipulate prompt selection, dynamically optimizing the backdoor trigger, and adopting sigmoid BCE loss to mitigate bias and prevent cheating;

3. We conduct extensive experiments on various prompt-based continual learners with different datasets and provide ablation studies to demonstrate the strength of our framework.

Organization The paper is organized as follows. Section 2 provides a brief overview of continual learning and prompt-based continual learning. In Section 3, we introduce the continual backdoor threat model, discuss backdoor challenges, and propose our prompt-based continual backdoor AOP framework. Section 4 empirically verifies the effectiveness of our AOP framework against various prompt-based incremental learners. Finally, Section 5 concludes the paper. Additional related work, discussions, and experiments are included in the supplementary material.

2 Background

Continual learning In continual learning scenarios, the model undergoes a sequential presentation of tasks $\mathcal{D}_1, ...\mathcal{D}_T$. Each task corresponds to distinct subsets of tuples $\mathcal{D}_t = \{\boldsymbol{x}_t^i, \boldsymbol{y}_t^i\}_{i=1}^{i=n_t}$, where $\boldsymbol{x}_t^i \in \mathcal{X}^t$ is the input sample, $\boldsymbol{y}_t^i \in \mathcal{Y}^t$ is the corresponding label, and n_t is the number of samples for task t. It is important to note that each class is exclusively associated with a single task [7, 3], meaning that $\mathcal{Y}^t$ and $\mathcal{Y}^{t'}$ are disjoint, and data from prior tasks become inaccessible during the training of subsequent tasks [45, 40]. The objective of continual learning is to continuously acquire the capability to classify newly introduced classes while maintaining proficiency on previously learned ones in a single model $f : \mathcal{X} \to \mathcal{Y}$. In this paper, and in prompt-based methods [45, 57, 58, 55, 40], f represents the pre-trained Vision Transformer (ViT) encoder. Additionally, ϕ is employed as the shared classification head, and ϕ_t is the classifier corresponding to classes specific to the given task t.

Prompt-based continual learning We provide a concise overview of L2P [58], which stands as the first work that integrates prompts into the context of continual learning. L2P introduces a prompt pool comprising learnable prompts and their corresponding keys $\left\{(\boldsymbol{k}_1, \boldsymbol{p}_1), (\boldsymbol{k}_2, \boldsymbol{p}_2), \cdots, \left(\boldsymbol{k}_{n_p}, \boldsymbol{p}_{n_p}\right)\right\}$ where n_p is total number of prompts. These prompts are then combined with image features and fed into a pre-trained ViT, instructing the model to perform classification. Prompts are queried in an instance-wise manner using the top-K cosine similarity $\gamma\left(q(\boldsymbol{x}), \boldsymbol{k}_i\right)$ between the keys and the query function $q(\boldsymbol{x}) = f(\boldsymbol{x})[0, :]$. Subsequent prompt-based methods are designed based on L2P, each featuring prompt utility and optimization modifications. A brief explanation of these methods is in Appendix A.

3 Backdoor Attack on Prompt-based Continual Learning (AOP)

We first outline the threat model and introduce key notations in Section 3.1. Then, we highlight the challenges when executing a backdoor attack against prompt-based incremental learners in Section 3.2. Building upon these considerations, we delineate the three primary components of AOP across Sections 3.3-3.5. A comprehensive overview and the end-to-end algorithm is in Appendix B.

3.1 Threat Model and Notations

Continual learning protocols We consider the class-incremental learning (CIL) setting in prompt-based continual learning [57, 58]. In CIL, training data for incremental tasks $\mathcal{D}_t$ arrive incrementally in a discrete manner. Each task consists of data for new M classes that have not been learned by the model before. Formally, each task $\mathcal{D}_t = \{\mathcal{D}_{m,t}\}_{m=1}^{M}$ with each class $\mathcal{D}_{m,t} = \{\boldsymbol{x}_i^{m,t}, y_i^{m,t}\}_{i=1}^{n_{m,t}}$ comprises input samples $\boldsymbol{x}_i^{m,t} \in \mathcal{X}$ and their corresponding labels $y_i^{m,t} = c_{m,t} \in \mathcal{Y}$, where $n_{m,t}$ represents the number of training samples for the corresponding class. In CIL, the learner is required to perform classification across all classes encountered up to task T without being provided with explicit task labels during inference. Data for different classes m and m' are gathered from different suppliers. To ease the ensuing presentation, the index t is omitted unless noted otherwise.

Backdoor attack protocols Let the attacker be the data supplier for class m with labels c_m. The attacker's goal is to poison the supplying dataset with a small amount of trigger-injected samples, such that any data from any classes if manipulated with the backdoor trigger, will be misclassified as c_m by the resulting incremental victim model when performing inference at any time t. An example of a triggered image is given in Figure 4.

Consider $\mathcal{D}_m = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n_m}$ as the benign training set of class m. The adversary then learns to generate the poisoned dataset $\mathcal{D}_p$. Specifically, $\mathcal{D}_p$ consists of two parts: a modified version of a selected subset (denoted as $\mathcal{D}_s$) of $\mathcal{D}_m$ and the remaining benign samples. Thus, $\mathcal{D}_p = \mathcal{D}_b \cup \mathcal{D}_c$, where c_m is the adversary target label, $\mathcal{D}_c = \mathcal{D}_m \backslash \mathcal{D}_s$, $\mathcal{D}_b = \{(\boldsymbol{x}', c_m) \mid \boldsymbol{x}' = G(\boldsymbol{x}), (\boldsymbol{x}, c_m) \in \mathcal{D}_s\}$, $\gamma \triangleq \frac{|\mathcal{D}_s|}{|\mathcal{D}_m|}$ is the poisoning rate, and $G : \mathcal{X} \to \mathcal{X}$ is an adversary-specified poisoned image generator. We follow [46, 29] and formulate $G(\boldsymbol{x}) = \boldsymbol{x} + \boldsymbol{\delta}$, where the perturbation $\boldsymbol{\delta}$ has a bounded ℓ_p-norm.

We emphasize that given the considered multi-data supplier scenario, we optimize the backdoor trigger following a *black-box* setting (where the attacker has no access to the training model or procedure) and a *clean-label* setting (where the attacker cannot change the label of data), which represent stealthy and challenging conditions in backdoor attacks.

3.2 Three Challenges When Backdooring CL

We outline three challenges encountered when executing backdoor attacks against continual learners. To generate a poisoned dataset, the adversary optimizes the backdoor trigger, necessitating the appearance of the training data, learner, and training criterion. However, in accordance with our threat model, none of these are accessible.

The first challenge, as outlined in the introduction, arises from the lack of knowledge about the victim's training data. Given that control is limited to the supplied data, which also represents the target class, prior research [64] suggests utilizing a public dataset (e.g., Tiny-ImageNet) as a surrogate training dataset. In this study, we explore the utilization of surrogate datasets in the context of prompt tuning.

Secondly, the adversary lacks information about the training learner and procedure, making it difficult to design backdoor knowledge that can withstand the incremental learning process. Despite impressive memory capabilities, continual learning methods have not yet fully matched the performance levels of joint training, and recent works [40] are still exploring ways to further avoid catastrophic forgetting. This issue also affects backdoor attacks, leading to a degradation in attack performance over time. Therefore, creating a surrogate learner that helps the trigger endure the incremental learning process is our second challenge.

Lastly, a backdoor attack entails poisoning the training dataset to induce the model to malfunction

when presented with specific trigger samples while maintaining normal performance on clean data. Huynh et al. [18] observe that this objective can be achieved even without any poisoning during training. This trigger, akin to adversarial noise, can deceive the classifier during inference, irrespective of whether data poisoning occurred during training, thereby counteracting the primary objective of the backdoor attack. Moreover, such adversarial noise can be mitigated by employing standard adversarial defenses. Consequently, preventing the generation of adversarial noise poses an additional challenge when optimizing the trigger.

3.3 Prompt Selection, Label Mapping, and Transferability

The core of prompt-based continual learning methods lies in the prompt pool and the prompt selection strategy. Specifically, the most relevant prompts are queried in an instance-wise manner and then concatenated with the sample to optimally guide the model in performing classification. We leverage this fundamental mechanism of the prompt-based approach to reframe the backdooring problem as one of manipulating prompt selections. As in Figures 1a and 1b, we aim to ensure that triggered samples are directed to select specific backdoor prompts, thereby causing the model to misclassify these backdoor-prompted samples into the desired class.

A key feature of visual prompting is its ability to act as a label mapping mechanism when performing downstream tasks using a pretrained model. In this context, prompts function as universal input perturbation templates, enabling the mapping of labels from a source dataset to a target dataset [10]. From this perspective, our aim of controlling prompt selection translates into manipulating label mappings between the two datasets. This new perspective paves the way for the "transferability" of our continual backdoor framework.

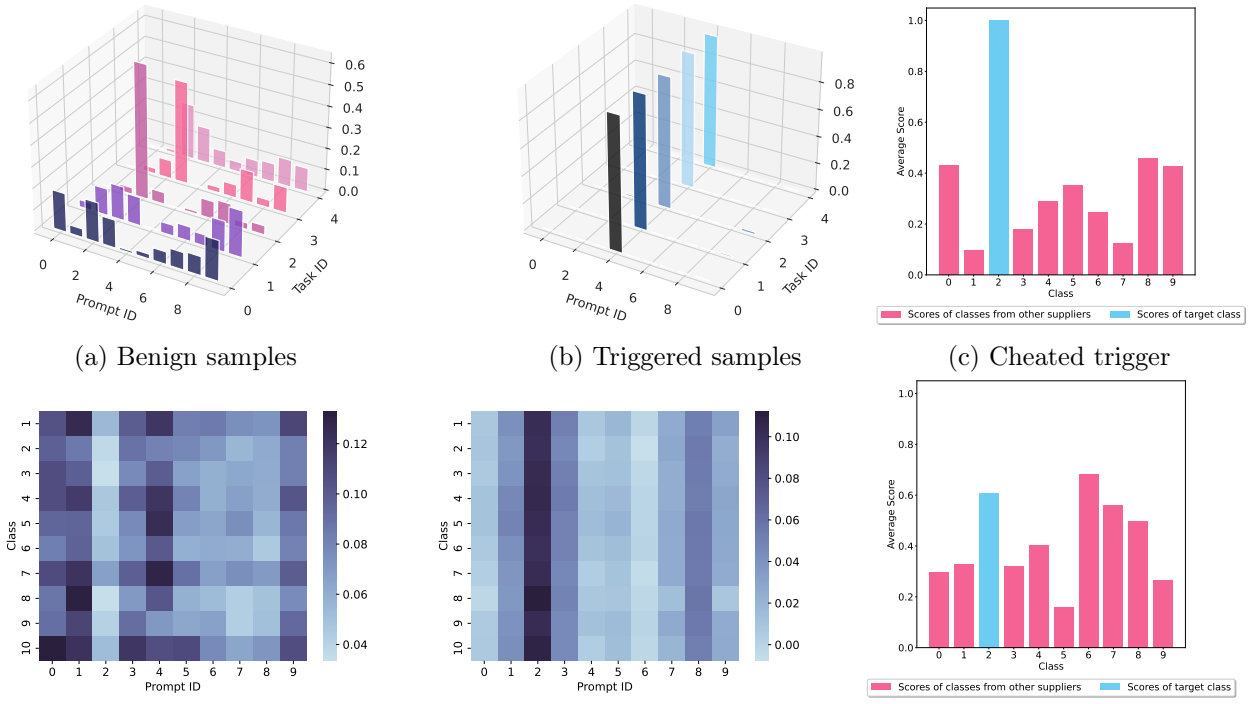
When optimizing the backdoor trigger, we employ a surrogate dataset, denoted as $\mathcal{D}_{\text{surrogate}}$, to address the backdoor transferability to data from other classes. It is worth noting that $\mathcal{D}_{\text{surrogate}}$ does not necessarily mirror the actual data distribution used to train the incremental model. This discrepancy stems from the visual prompting property discussed earlier. In particular, instead of optimizing a trigger that causes the poisoned data to be misclassified by the model, our backdoor trigger can be viewed as activating an incorrect mapping to the target class. Since we focus on manipulating the mapping and prompt selection rather than the dataset itself, $\mathcal{D}_{\text{surrogate}}$ can be chosen differently from the actual dataset to align with our objectives.

3.4 Static-dynamic Trigger Optimization

Since we lack information about the victim's continual model, we use $\mathcal{D}_{\text{surrogate}}$ to train a surrogate incremental learner and simulate the continual learning pipeline. We then optimize the backdoor trigger $\boldsymbol{\delta}$ based on this surrogate incremental model. Specifically, we employ the surrogate learner with two states: a static state that reflects how prompts learn label mappings between the source and target datasets, and a dynamic state that reflects the continuous learning procedure of the victim model. Formally, our static-dynamic trigger optimization involves the following four stages:

(0) Preparation To set up the static-dynamic framework, we partition the surrogate dataset $\mathcal{D}_{\text{surrogate}}$ into two subsets: $\mathcal{D}_{\text{static}}$ for the static surrogate stage and $\mathcal{D}_{\text{dynamic}}$ for the dynamic surrogate stage.

(1) Static surrogate stage In this initial stage, we train the prompts on $\mathcal{D}_{\text{static}} \cup \mathcal{D}_m$ to capture the label mapping functionality between the source and target datasets. During this phase, the prompts are optimized to instruct the model to correctly classify clean input images. Consequently, we obtain



(a) Benign samples (b) Triggered samples (c) Cheated trigger

(d) Benign samples (e) Triggered samples (f) Authentic trigger

Figure 1: (a) and (b): AOP's prompt selection frequency on benign and triggered samples when attacking DualPrompt. (d) and (e): AOP's average key-query similarities concerning benign and triggered samples when attacking DualPrompt-PGP. (c) and (f): Scores obtained from the clean model for AOP's triggered samples optimized with CE and BCE, respectively.

a pool of benign prompts for clean data. Denoting the prompt pool as $\mathbf{P} = \left\{\boldsymbol{p}_1, \boldsymbol{p}_2, \cdots, \boldsymbol{p}_{n_p}\right\}$ and $\mathbf{K} = \left\{\boldsymbol{k}_1, \boldsymbol{k}_2, \cdots, \boldsymbol{k}_{n_p}\right\}$ as the corresponding prompt keys, where n_p is the prompt pool size, the objective for this optimization step follows [58] and is given by:

$$\min_{\mathbf{P},\mathbf{K},\phi} \mathcal{L}\left(\phi\left(f\left(\boldsymbol{x};\boldsymbol{P}\right)\right), y\right) + \lambda \sum_{\mathbf{K}_{\boldsymbol{x}}} \gamma\left(q(\boldsymbol{x}), \boldsymbol{k}_i\right). \tag{1}$$

Here, $\mathbf{K}_{\boldsymbol{x}}$ denotes a subset of the top-K keys specifically selected for each sample $\boldsymbol{x}$. γ is the function that assesses the similarity between the query feature $q(\boldsymbol{x})$ and prompt key. The scalar λ weights the loss. The first term is the softmax cross-entropy loss, while the second term acts as a regularizer to encourage selected keys to be closer to the corresponding query features.

(2) Trigger optimization stage During this stage, the adversary optimizes the trigger $\boldsymbol{\delta}$ to induce misclassification of the triggered inputs into the target class. Specifically, the trigger loss function can be expressed as follows:

$$\min_{\boldsymbol{\delta}} \sum_{(\boldsymbol{x}, c_m) \in \mathcal{D}_m} \left[\mathcal{L}\left(\phi\left(f\left(\boldsymbol{x}+\boldsymbol{\delta};\boldsymbol{P}\right)\right), c_m\right)\right]. \tag{2}$$

(3) Transition stage This stage is designed to align the surrogate learner with the behaviour of the victim learner when being updated with new incoming tasks. Specifically, we continuously train

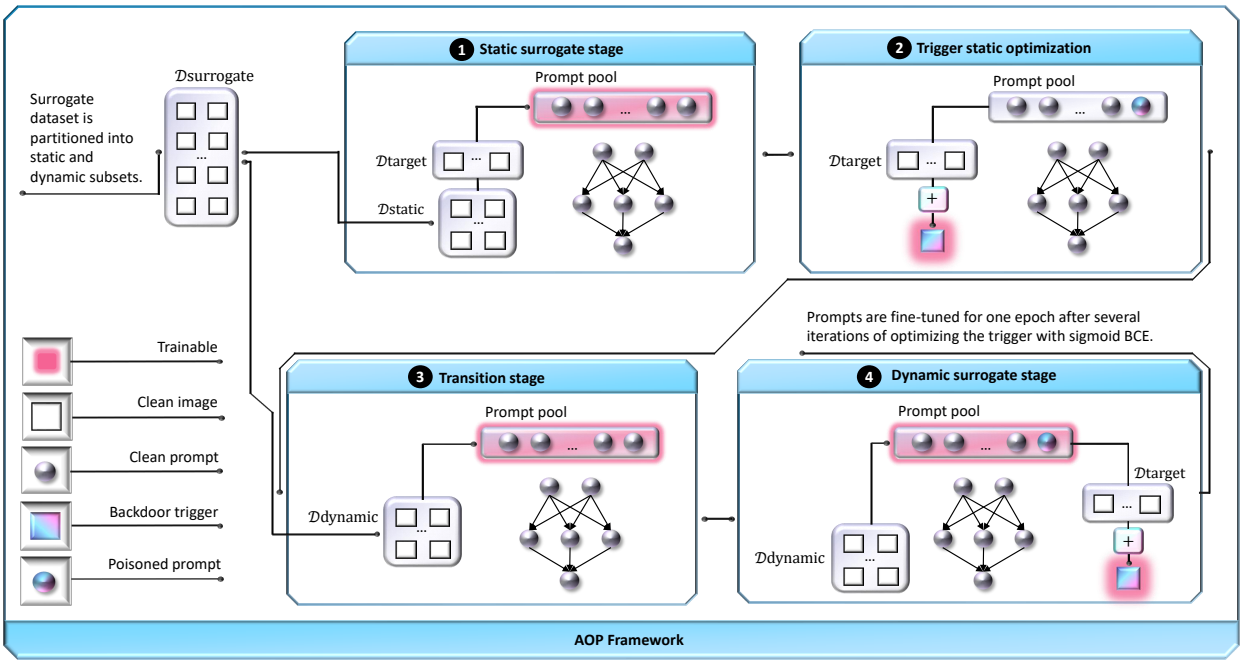


Figure 2: The AOP procedure begins by selecting a surrogate dataset, which is then divided into two subsets: $\mathcal{D}_{\text{static}}$ and $\mathcal{D}_{\text{dynamic}}$. In stage (1), $\mathcal{D}_{\text{static}}$ is employed to establish a static surrogate learner along with prompts. Following this, in stage (2), the trigger optimization process takes place based on this initial model. Next, in stage (3), the learner is updated from $\mathcal{D}_{\text{dynamic}}$, which serves as a transition between stages. Finally, in stage (4), the trigger is fine-tuned, with the prompt being updated periodically throughout the optimization process.

the prompts from Stage **(1)** with the same objective as outlined in equation (1), but using $\mathcal{D}_{\text{dynamic}}$. In essence, the goal of this stage is to statically prepare the surrogate learner for the subsequent dynamic stage.

(4) Dynamic surrogate stage In this stage, we aim to acquaint the backdoor trigger with the continuously updated prompts resulting from the continual learning process. This dynamic stage entails fine-tuning the prompt components for one epoch, as in Stage **(3)**, following several iterations of optimization of the trigger with equation (2). This iterative process is repeated for multiple rounds to enhance the resilience of the backdoor trigger against the continual learning process.

After optimizing the trigger through the aforementioned four stages, the optimized trigger $\boldsymbol{\delta}^*$ is used to poison a small portion of $\mathcal{D}_m$, which is then released to the victim learner. Summarization of AOP is in Figure 2 and Appendix B.

3.5 Towards an Authentic Backdoor Trigger

Are we truly optimizing a backdoor trigger? As discussed in Section 3.2, optimizing the trigger with these objectives can unintentionally transform it into adversarial noise. While our static-dynamic framework can generate a robust trigger that withstands intense incremental learning processes, it might deviate into adversarial perturbation. To further explore this phenomenon, we analyze the output scores in Figure 1c. The visualization reveals that even when processed by a

clean model unaffected by backdoor attacks, the poisoned samples are consistently misclassified towards the target class with dominant scores. This observation prompts a reconsideration of the backdoor trigger optimization process. We discovered that the overconfident score bias towards the target class is primarily induced by the commonly used softmax with cross-entropy loss function. Softmax introduces competition between classes, and the subsequent cross-entropy loss tends to elevate the scores of the target class significantly above the others. This pronounced bias compels the trigger to act like adversarial noise.

Sigmoid with binary cross entropy loss To reduce biases, we mitigate the competition between the target class and other classes caused by the relative scoring of softmax by employing a sigmoid function after the logits to compute output scores. This approach shifts the optimization focus towards independently increasing the scores of target classes rather than suppressing others. Subsequently, we utilize binary cross-entropy loss to enable independent optimization processes. Following [8], the gradient of the loss at score (s_j) for class j is computed as $\frac{\partial \mathcal{L}_{\text{BCE}}(\boldsymbol{\theta})}{\partial s_j} = \sigma(s_j) - \mathbb{I}\{j = \tilde{y}\}$, thereby constraining the score of the target class to a certain level regardless of the scores of other classes. As a result, during inference with a non-backdoored clean model, the output scores are more balanced between classes, as shown in Figure 1f. This balance prevents the problem of generating adversarial noise when optimizing the backdoor trigger.

4 Experiments

In this section, we first describe the experimental setups, followed by presenting the results in four key aspects: the overall backdooring ability of AOP, its performance with different surrogate datasets, the robustness of AOP with varying attack times, and the efficacy of adopting BCE in preventing the generation of adversarial perturbations. Further discussions on performance, visualizations, baselines, efficacy against defenses, and poisoning rate sensitivity are deferred to Appendix D.

4.1 Experimental Setup

Victim incremental learners We evaluate our continual backdoor framework against 6 prompt-based continual learning methods: L2P [58], DualPrompt [57], HiDe-Prompt [55], CODA-Prompt [45], and two variants of PGP [40], namely L2P-PGP and DualPrompt-PGP. We follow the original settings and implementations of each method. All learners utilize the ViT-B/16 backbone [14], pre-trained on ImageNet-1K [41], except for HiDe-Prompt, which is pre-trained on iBOT-1K [66]. Detailed experimental information is in Appendix C.

Datasets For the victim's training dataset, we use three variants of ImageNet-R [17]: 5-Split, 10-Split, and 20-Split ImageNet-R. These variants divide the 200 classes of the original dataset into 5, 10, and 20 tasks, respectively. Additionally, we conduct experiments on the 5-Split-CUB200 dataset, which partitions the original CUB200 [52] dataset into 5 tasks, each containing 40 classes. For the attacker's surrogate dataset, we primarily use TinyImageNet [23] for all experiments and CIFAR100 [22] in specific settings.

Backdoor setting Following the guidelines of [64], we set the maximum poison ratio to 25 images, corresponding to 0.1% of ImageNet-R and 0.5% of CUB200. Additionally, we set the upper bound of the ℓ_∞-norm of triggers to $\frac{16}{255}$, in line with standard practices in the literature [48, 42]. During inference, the trigger is amplified by a factor of 3 [48, 64].

Table 1: Backdoor performance against L2P, DualPrompt, and PGP on 5-Split-CUB200. The attacker is the supplier for a random class in task 1. The dynamic stage takes place over 5 rounds. Results are reported when using TinyImageNet and CIFAR100 as surrogate datasets. For ACC, we additionally report the change in clean accuracy compared to clean-training learners. For ASR, we provide a comparison with the baseline [64] (without dynamic optimization and not using BCE).

Surrogate dataset →	TinyImageNet		CIFAR100	
	ASR	ACC	ASR	ACC
L2P	99.96 ± 0.02 ($\uparrow$ 86.44)	74.71 ± 0.58 ($\downarrow$ 0.17)	99.99 ± 0.02 ($\uparrow$ 64.91)	74.44 ± 0.54 ($\downarrow$ 0.44)
DualPrompt	99.93 ± 0.02 ($\uparrow$ 57.08)	82.62 ± 0.66 ($\uparrow$ 0.10)	99.95 ± 0.05 ($\uparrow$ 42.36)	82.71 ± 0.55 ($\uparrow$ 0.19)
L2P-PGP	99.97 ± 0.01 ($\uparrow$ 89.73)	74.97 ± 0.83 ($\downarrow$ 0.48)	100.00 ± 0.00 ($\uparrow$ 68.82)	75.70 ± 0.50 ($\uparrow$ 0.25)
DualPrompt-PGP	99.93 ± 0.02 ($\uparrow$ 56.70)	82.45 ± 0.29 ($\downarrow$ 0.31)	99.99 ± 0.01 ($\uparrow$ 44.83)	82.84 ± 0.12 ($\uparrow$ 0.08)

Table 2: Backdoor performance across different prompt-based continual learning methods on three variants of Split-ImageNet-R. The adversary's target class is chosen randomly from the classes in task 1. The dynamic stage is iterated for 10 rounds. The surrogate dataset used is TinyImageNet. We also report the change in ACC compared to non-attacked learners.

	5-Split-ImageNet-R		10-Split-ImageNet-R		20-Split-ImageNet-R	
	ASR	ACC	ASR	ACC	ASR	ACC
L2P	99.76 ± 0.10	64.27 ± 0.65 ($\downarrow$ 0.77)	99.56 ± 0.22	62.43 ± 0.58 ($\downarrow$ 0.12)	98.24 ± 0.21	60.51 ± 1.17 ($\downarrow$ 0.83)
DualPrompt	99.57 ± 0.25	70.69 ± 0.56 ($\downarrow$ 0.62)	99.26 ± 0.39	69.17 ± 0.27 ($\downarrow$ 0.85)	96.17 ± 0.89	66.04 ± 0.43 ($\downarrow$ 0.21)
CODA-Prompt	98.16 ± 1.01	74.15 ± 0.11 ($\downarrow$ 1.04)	96.55 ± 1.29	72.86 ± 0.11 ($\downarrow$ 0.02)	71.27 ± 2.86	70.86 ± 0.94 ($\downarrow$ 0.04)
HiDe-Prompt	98.65 ± 0.90	74.89 ± 0.60 ($\downarrow$ 0.32)	94.66 ± 0.93	71.99 ± 0.37 ($\downarrow$ 0.46)	93.79 ± 0.66	70.93 ± 0.86 ($\downarrow$ 0.09)
L2P-PGP	99.33 ± 0.05	64.38 ± 0.57 ($\uparrow$ 0.10)	99.36 ± 0.15	61.73 ± 0.38 ($\uparrow$ 0.33)	98.84 ± 0.16	60.74 ± 1.17 ($\downarrow$ 0.15)
DualPrompt-PGP	99.83 ± 0.27	70.80 ± 0.08 ($\downarrow$ 0.08)	99.17 ± 0.43	69.24 ± 0.41 ($\downarrow$ 0.18)	97.01 ± 0.75	66.32 ± 1.04 ($\downarrow$ 0.76)

Metrics The evaluation of our framework utilizes two key metrics: (1) average accuracy (ACC) and (2) attack success rate (ASR). ACC assesses the accuracy of the backdoored model on benign test samples, whereas ASR measures the proportion of attacked samples that the compromised model predicts as the target label, reflecting the backdoor attack's effectiveness. In the context of continual learning, ACC and ASR at a given time t are averaged across the corresponding metrics for all data from task 1 to task t. All results are averaged over 3 runs for fair comparisons.

4.2 Effectiveness of AOP

We report the ASR and ACC when performing backdoor attacks against various incremental learners in Table 1 and Table 2. As observed from the tables, our framework consistently achieves high

Table 3: Backdoor performance when the target class belongs to different tasks T. The results are reported when the victim's training dataset is 10-Split-ImageNet-R, and the attacker's surrogate dataset is TinyImageNet.

	$T=1$		$T=4$		$T=10$	
	ASR	ACC	ASR	ACC	ASR	ACC
L2P	99.56 ± 0.22	62.43 ± 0.58	99.61 ± 0.19	62.09 ± 0.06	99.89 ± 0.05	62.27 ± 0.26
L2P-PGP	99.36 ± 0.15	62.73 ± 0.38	99.77 ± 0.08	62.88 ± 0.73	99.85 ± 0.35	62.32 ± 0.82

Table 4: ASR of clean, non-attacked learners on triggered samples. Results are compared between triggers optimized with CE softmax and BCE sigmoid loss.

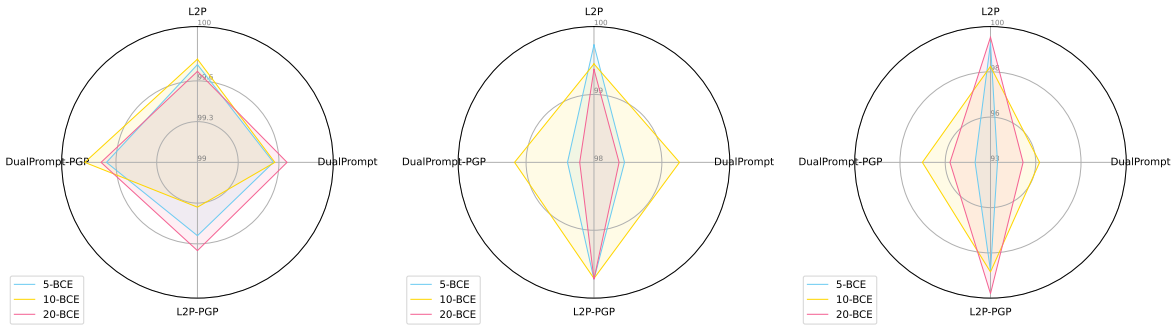
		L2P	DualPrompt		
		10-Split-ImageNet-R	5-Split-ImageNet-R	10-Split-ImageNet-R	20-Split-ImageNet-R
AOP with CE	Top-1 ASR	74.18	34.18	42.85	96.93
	Top-5 ASR	96.89	92.78	97.01	99.63
AOP with BCE	Top-1 ASR	0.00	0.00	0.00	0.00
	Top-5 ASR	0.00	0.72	0.12	2.68

ASR with negligible effect on the ACC of clean samples. This is due to the inherent characteristics of continual learning, which enable the learner to perform well across different tasks, making it vulnerable to backdoor attacks. By considering backdooring in continual learning as an additional "backdoor task," the plasticity of continual learning allows the ASR, or performance on the backdoor task, to be high without degrading the ACC on clean tasks.

It is worth noting that ASR still suffers from the catastrophic forgetting phenomenon of continual learning for long sequence tasks. Specifically, in Table 2, the 20-Split-ImageNet-R performs worse than the 5-split and 10-split versions across all experiments. This indicates that the more tasks and the longer the incremental learning process, the higher the chance for a decrease in ASR. However, the ACC also suffers from this phenomenon, as it is a major issue in continual learning.

While prompt-based methods share a common core of utilizing prompt pools and selecting relevant prompts for each task or class, each exhibits distinct characteristics. Our framework observes a significantly lower ASR when backdooring CODA-Prompt. This is because CODA-Prompt utilizes all prompts in the prompt pool through its weighted mechanism instead of selecting only the top-K relevant prompts. Consequently, even with triggered samples, clean prompts still exert some influence, leading to degradation in ASR.

Different surrogate datasets Another factor that makes prompt-based continual learning vulnerable is the utilization of prompting. As shown in Figures 1d and 1e, AOP's triggered samples consistently have the highest similarity with prompt ID 2, which, in contrast, shows the smallest similarity with benign samples. Thus, as discussed in Section 3.3, prompting allows for actual data differences when choosing surrogate datasets. We report the backdoor performance using TinyImageNet and CIFAR100 as surrogate datasets in Table 1. The experiments show consistently high ASR results for both surrogate data choices, confirming the transferability of our continual backdoor framework.



(a) 5-Split-ImageNet-R (b) 10-Split-ImageNet-R (c) 20-Split-ImageNet-R

Figure 3: ASR when varying number of dynamic rounds.

Different attack times We report the ASR in Table 3, considering scenarios where the target class belongs to different tasks that arrive at different times. We observe slight increases in ASR when the attack class is part of later tasks, as it experiences less forgetting. Nonetheless, our method AOP consistently maintains a high ASR, exceeding 99% at all three reported attack times. This convincingly demonstrates that the backdoor knowledge can be effectively transferred to both previously learned and incoming future classes.

Different dynamic rounds We illustrate the attack performance across varying numbers of dynamic rounds in Figure 3. As discussed above, the ASR decreases when tested on the 20-Split-ImageNet. We observe that increasing the number of dynamic rounds does not consistently lead to higher performance. However, from a positive perspective, since the adversary lacks information about the total number of tasks, decreasing and increasing dynamic rounds should not have too much impact on ASR. We emphasize that in long sequence tasks, both ASR and ACC degrade due to forgetting.

Enhancing backdoor authenticity via sigmoid BCE As shown in Table 4, triggers optimized with softmax CE retain considerable scores even when tested on non-backdoored models. This suggests that CE optimization might lead to the generation of adversarial perturbations. Conversely, when optimized using sigmoid BCE, the ASR on clean models remains consistently low. This confirms that adopting BCE can enhance the authenticity of backdoor triggers and avoid generating adversarial noise.

5 Conclusion

This paper explores the vulnerability of prompt-based continual learning methods and their susceptibility to backdoor attacks. We emphasize three critical properties that a backdoor continual framework should possess: transferability to unknown data from other classes, resilience against incremental learning procedures, and the authenticity of the backdoor trigger. Building upon these considerations, we propose a novel continual backdoor framework. We leverage the label mapping functionality of prompting to promote transferability, incorporate a static-dynamic optimization approach to enhance resilience, and employ BCE sigmoid loss to mitigate the adversarial noise problem. Extensive experiments confirm the effectiveness of our backdoor framework against various prompt-based continual learners.

Nonetheless, we acknowledge some limitations in our work. Firstly, competition between the target classes and the remaining classes remains necessary to some extent. Relying solely on BCE to

eliminate relative scoring might hurt the performance. Secondly, certain defenses we employed to assess our approach may not be optimal for continual learning scenarios. Thus, regarding future directions, there is potential in exploring other threat models and defenses for backdooring continual learning and extending backdoor attacks to other continual learning approaches.

References

[1] D. Abati, J. Tomczak, T. Blankevoort, S. Calderara, R. Cucchiara, and B. E. Bejnordi. Conditional channel gated networks for task-aware continual learning, 2020. (Cited on page 17.)

[2] H. Ahn, S. Cha, D. Lee, and T. Moon. Uncertainty-based continual learning with adaptive regularization, 2019. (Cited on page 17.)

[3] H. Ahn, J. Kwak, S. F. Lim, H. Bang, H. Kim, and T. Moon. Ss-il: Separated softmax for incremental learning. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 824–833, 2020. (Cited on page 3.)

[4] P. Buzzega, M. Boschini, A. Porrello, D. Abati, and S. Calderara. Dark experience for general continual learning: a strong, simple baseline, 2020. (Cited on page 17.)

[5] X. Cai, H. Xu, S. Xu, Y. Zhang, and X. Yuan. Badprompt: Backdoor attacks on continuous prompts, 2022. (Cited on page 2.)

[6] H. Cha, J. Lee, and J. Shin. Co^{2}l: Contrastive continual learning, 2021. (Cited on page 17.)

[7] S. Cha, S. Cho, D. Hwang, S. Hong, M. Lee, and T. Moon. Rebalancing batch normalization for exemplar-based class-incremental learning. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20127–20136, 2022. (Cited on page 3.)

[8] S. Cha, b. kim, Y. Yoo, and T. Moon. Ssul: Semantic segmentation with unknown label for exemplar-based class-incremental learning. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 10919–10930. Curran Associates, Inc., 2021. (Cited on page 8.)

[9] A. Chaudhry, M. Rohrbach, M. Elhoseiny, T. Ajanthan, P. K. Dokania, P. H. S. Torr, and M. Ranzato. On tiny episodic memories in continual learning, 2019. (Cited on page 17.)

[10] A. Chen, Y. Yao, P.-Y. Chen, Y. Zhang, and S. Liu. Understanding and improving visual prompting: A label-mapping perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19133–19143, June 2023. (Cited on page 5.)

[11] Y. Dai and S. Li. Chameleon: Adapting to peer images for planting durable backdoors in federated learning, 2023. (Cited on page 2.)

[12] K. Doan, Y. Lao, and P. Li. Backdoor attack with imperceptible input and latent modification. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 18944–18957. Curran Associates, Inc., 2021. (Cited on page 2.)

[13] K. Doan, Y. Lao, W. Zhao, and P. Li. Lira: Learnable, imperceptible and robust backdoor attacks. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11946–11956, 2021. (Cited on page 2.)

[14] 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*, 2021. (Cited on page 8.)

[15] S. Farquhar and Y. Gal. A unifying bayesian view of continual learning, 2019. (Cited on page 17.)

[16] 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 *35th Annual Computer Security Applications Conference (ACSAC)*, 2019. (Cited on page 22.)

[17] D. Hendrycks, S. Basart, N. Mu, S. Kadavath, F. Wang, E. Dorundo, R. Desai, T. L. Zhu, S. Parajuli, M. Guo, D. X. Song, J. Steinhardt, and J. Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8320–8329, 2020. (Cited on page 8.)

[18] T. Huynh, D. Nguyen, T. Pham, and A. Tran. Combat: Alternated training for effective clean-label backdoor attacks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(3):2436–2444, Mar. 2024. (Cited on pages 2, 5, and 18.)

[19] S. Jung, H. Ahn, S. Cha, and T. Moon. Continual learning with node-importance based adaptive group sparse regularization, 2021. (Cited on page 17.)

[20] S. Kang, Z. Shi, and X. Zhang. Poisoning generative replay in continual learning to promote forgetting. In A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 15769–15785. PMLR, 23–29 Jul 2023. (Cited on page 18.)

[21] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, mar 2017. (Cited on page 17.)

[22] A. Krizhevsky. Learning multiple layers of features from tiny images. 2009. (Cited on page 8.)

[23] Y. Le and X. S. Yang. Tiny imagenet visual recognition challenge. 2015. (Cited on page 8.)

[24] H. Li and G. Ditzler. Targeted data poisoning attacks against continual learning neural networks. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2022. (Cited on page 18.)

[25] X. Li, Z. Chen, Y. Zhao, Z. Tong, Y. Zhao, A. Lim, and J. T. Zhou. Pointba: Towards backdoor attacks in 3d point cloud, 2021. (Cited on page 2.)

[26] X. Li, Y. Zhou, T. Wu, R. Socher, and C. Xiong. Learn to grow: A continual structure learning framework for overcoming catastrophic forgetting, 2019. (Cited on page 17.)

[27] Y. Li, Y. Bai, Y. Jiang, Y. Yang, S.-T. Xia, and B. Li. Untargeted backdoor watermark: Towards harmless and stealthy dataset copyright protection. In *NeurIPS*, 2022. (Cited on page 17.)

[28] Y. Li, T. Li, K. Chen, J. Zhang, S. Liu, W. Wang, T. Zhang, and Y. Liu. Badedit: Backdooring large language models by model editing. In *The Twelfth International Conference on Learning Representations*, 2024. (Cited on page 2.)

[29] Y. Li, Y. Li, B. Wu, L. Li, R. He, and S. Lyu. Invisible backdoor attack with sample-specific triggers, 2021. (Cited on page 4.)

[30] Z. Li, H. Sun, P. Xia, H. Li, B. Xia, Y. Wu, and B. Li. Efficient backdoor attacks for deep neural networks in real-world scenarios. In *The Twelfth International Conference on Learning Representations*, 2024. (Cited on pages 2 and 17.)

[31] C. Liao, H. Zhong, A. Squicciarini, S. Zhu, and D. Miller. Backdoor embedding in convolutional neural network models via invisible perturbation, 2018. (Cited on page 2.)

[32] Y. Liu, B. Schiele, and Q. Sun. Adaptive aggregation networks for class-incremental learning. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2021. (Cited on page 17.)

[33] N. Loo, S. Swaroop, and R. E. Turner. Generalized variational continual learning, 2020. (Cited on page 17.)

[34] A. Mallya and S. Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning, 2018. (Cited on page 17.)

[35] R. Min, Z. Qin, L. Shen, and M. Cheng. Towards stable backdoor purification through feature shift tuning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. (Cited on pages 2 and 23.)

[36] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard. Universal adversarial perturbations, 2017. (Cited on page 2.)

[37] A. Nguyen and A. Tran. Wanet – imperceptible warping-based backdoor attack, 2021. (Cited on page 2.)

[38] C. V. Nguyen, Y. Li, T. D. Bui, and R. E. Turner. Variational continual learning, 2018. (Cited on page 17.)

[39] Q. Pham, C. Liu, and S. C. H. Hoi. Continual learning, fast and slow, 2023. (Cited on page 17.)

[40] J. Qiao, Z. Zhang, X. Tan, C. Chen, Y. Qu, Y. Peng, and Y. Xie. Prompt gradient projection for continual learning. In *International Conference on Learning Representations*, 2024. (Cited on pages 1, 3, 4, 8, 17, and 18.)

[41] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge, 2015. (Cited on page 8.)

[42] A. Saha, A. Subramanya, and H. Pirsiavash. Hidden trigger backdoor attacks, 2019. (Cited on pages 2, 8, and 17.)

[43] J. Serrà, D. Surís, M. Miron, and A. Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task, 2018. (Cited on page 17.)

[44] Z. Sha, X. He, P. Berrang, M. Humbert, and Y. Zhang. Fine-tuning is all you need to mitigate backdoor attacks, 2022. (Cited on pages 2 and 23.)

[45] J. S. Smith, L. Karlinsky, V. Gutta, P. Cascante-Bonilla, D. Kim, A. Arbelle, R. Panda, R. Feris, and Z. Kira. Coda-prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning, 2023. (Cited on pages 1, 3, 8, 17, and 18.)

[46] H. Souri, M. Goldblum, L. Fowl, R. Chellappa, and T. Goldstein. Sleeper agent: Scalable hidden trigger backdoors for neural networks trained from scratch. *arXiv preprint arXiv:2106.08970*, 2021. (Cited on pages 2, 4, and 17.)

[47] W. Sun, X. Zhang, H. LU, Y.-C. Chen, T. Wang, J. Chen, and L. Lin. Backdoor contrastive learning via bi-level trigger optimization. In *The Twelfth International Conference on Learning Representations*, 2024. (Cited on page 17.)

[48] A. Turner, D. Tsipras, and A. Madry. Label-consistent backdoor attacks, 2019. (Cited on pages 2, 8, and 17.)

[49] M. Umer, G. Dawson, and R. Polikar. Targeted forgetting and false memory formation in continual learners through adversarial backdoor attacks. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2020. (Cited on page 18.)

[50] M. Umer and R. Polikar. Adversarial targeted forgetting in regularization and generative based continual learning models. *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2021. (Cited on page 18.)

[51] M. Umer and R. Polikar. False memory formation in continual learners through imperceptible backdoor trigger. *ArXiv*, abs/2202.04479, 2022. (Cited on page 18.)

[52] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-ucsd birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. (Cited on page 8.)

[53] 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 *2019 IEEE Symposium on Security and Privacy (SP)*, pages 707–723, 2019. (Cited on page 21.)

[54] H. Wang, K. Sreenivasan, S. Rajput, H. Vishwakarma, S. Agarwal, J. yong Sohn, K. Lee, and D. Papailiopoulos. Attack of the tails: Yes, you really can backdoor federated learning, 2020. (Cited on page 2.)

[55] L. Wang, J. Xie, X. Zhang, M. Huang, H. Su, and J. Zhu. Hierarchical decomposition of prompt-based continual learning: Rethinking obscured sub-optimality. *Advances in Neural Information Processing Systems*, 2023. (Cited on pages 1, 3, 8, 17, and 18.)

[56] Z. Wang, T. Jian, K. Chowdhury, Y. Wang, J. Dy, and S. Ioannidis. Learn-prune-share for lifelong learning, 2020. (Cited on page 17.)

[57] Z. Wang, Z. Zhang, S. Ebrahimi, R. Sun, H. Zhang, C.-Y. Lee, X. Ren, G. Su, V. Perot, J. Dy, and T. Pfister. Dualprompt: Complementary prompting for rehearsal-free continual learning, 2022. (Cited on pages 1, 3, 4, 8, and 17.)

[58] Z. Wang, Z. Zhang, C.-Y. Lee, H. Zhang, R. Sun, X. Ren, G. Su, V. Perot, J. Dy, and T. Pfister. Learning to prompt for continual learning, 2022. (Cited on pages 1, 3, 4, 6, 8, and 17.)

[59] C. Xiang, C. R. Qi, and B. Li. Generating 3d adversarial point clouds, 2019. (Cited on page 2.)

[60] Z. Xiang, D. J. Miller, S. Chen, X. Li, and G. Kesidis. A backdoor attack against 3d point cloud classifiers, 2021. (Cited on page 2.)

[61] C. Xie, K. Huang, P.-Y. Chen, and B. Li. Dba: Distributed backdoor attacks against federated learning. In *International Conference on Learning Representations*, 2020. (Cited on page 2.)

[62] S. Yan, J. Xie, and X. He. Der: Dynamically expandable representation for class incremental learning, 2021. (Cited on page 17.)

[63] D. Yin, M. Farajtabar, and A. Li. Sola: Continual learning with second-order loss approximation. In *Workshop of Advances in Neural Information Processing Systems*, 2020. (Cited on page 17.)

[64] Y. Zeng, M. Pan, H. A. Just, L. Lyu, M. Qiu, and R. Jia. Narcissus: A practical clean-label backdoor attack with limited information, 2022. (Cited on pages 2, 4, 8, 9, 17, and 21.)

[65] Z. Zhang, A. Panda, L. Song, Y. Yang, M. W. Mahoney, J. E. Gonzalez, K. Ramchandran, and P. Mittal. Neurotoxin: Durable backdoors in federated learning, 2022. (Cited on page 2.)

[66] J. Zhou, C. Wei, H. Wang, W. Shen, C. Xie, A. Yuille, and T. Kong. Image BERT pre-training with online tokenizer. In *International Conference on Learning Representations*, 2022. (Cited on page 8.)

Supplementary Material for "Backdoor Attack in Prompt-Based Continual Learning"

In this supplementary material, we first review related work on continual learning, prompt-based continual learning, and backdoor attacks in Appendix A. Next, we summarize our AOP in Appendix B. Implementation details, additional experiments, and visualizations are provided in Appendices C and D, respectively. Finally, we discuss broader impacts in Appendix E.

A Related Work

Continual learning Adapting to new knowledge is an innate human capability, yet it poses significant challenges for machine learning models. Continual learning emerges as one approach to bridge this gap between models and humans, which encourages models to continuously acquire new knowledge from new data while retaining previously learned ones. The regularization/prior approach [21, 38, 15, 33, 19, 2, 63] effectively preserves learned knowledge by controlling the learning of the model's parameters through a regularization term in the objective function. Architecture-based approaches [26, 34, 43, 56, 62, 1, 32] extend the model's plasticity by expanding its network to accommodate new knowledge. Rehearsal-based approaches [4, 6, 39, 9] rely on a memory buffer to retain past knowledge. Continual learning primarily focuses on the class-incremental learning (CIL) setting, which is the most challenging and representative setting since the task boundaries are not available during inference. While rehearsal-based approaches achieve state-of-the-art performance [4] in CIL, they violate data privacy requirements as they necessitate the storage of past data.

Prompt-based continual learning With few learnable parameters and not relying on memory buffers, prompt-based continual learning methods achieve state-of-the-art performance. These methods are especially suitable for scenarios where data privacy is crucial. Specifically, prompt-based approaches leverage the power of pre-trained models, learning only a small number of prompts to guide the model's performance across different tasks or classes. L2P [58] is the first work to explore prompting in continual learning. It constructs a prompt pool and selects appropriate prompts for each input. Building on L2P, DualPrompt [57] employs prefix-tuning and constructs two types of prompts: task-sharing and task-specific. CODA-Prompt [45] enhances prompt selection with an adaptive attention mechanism. HiDe-Prompt [55] examines the influence of various pretraining paradigms and decomposes the objective into hierarchical components. PGP [40] uses prompt gradient projection to promote updates in orthogonal directions, effectively preventing forgetting.

Backdoor attack A backdoor attack aims to cause a model to misbehave according to an adversary's target when the input data contains a specific backdoor trigger, while still performing normally on clean input data. Backdoor attacks have been explored in different settings and under various threat models, which identify the attacker's accessibility. In a black-box setting [42, 46, 27, 47], the attacker has no control over the training process and only has access to the dataset, which they then poison and release to the victim. Another line of work [42, 46, 64, 48] assumes that the attacker cannot flip the labels of the dataset (clean-label). Recently, attackers' control has been limited to data-constrained scenarios where they only have access to a small proportion of data. For example, [64] employs a surrogate clean model to optimize a clean-label backdoor trigger, while [30] leverages the zero-shot capabilities of the CLIP model to suppress clean features and augment the poisoning

(a) Clean image (b) Backdoor trigger (c) Poisoned image

Figure 4: Visualizations of the clean image, backdoor trigger, and poisoned image.

features. Additionally, [18] observes that even with carefully alternated training to train a surrogate poisoned model, the optimized backdoor trigger tends to become adversarial noise.

Previous works on backdoor attacks against continual learning have primarily focused on non-targeted attacks, aiming to degrade the model's performance in general. These studies typically explore task-incremental and domain-incremental settings using various approaches. For instance, [24] describes a white-box attack where the attacker has control over the training model and seeks to force the neural network to forget previously learned knowledge. Other works, such as [49, 50, 51], focus on regularization-based and replay-based learners in domain-incremental and task-incremental learning scenarios, aiming to degrade the performance of the first task. Similarly, [24] and [20] aim to undermine the performance of continual learners. In contrast, our work focuses on targeted backdoor attacks. We aim to manipulate the attacked learner to classify poisoned data from any task into a desired target class while maintaining high accuracy on clean data. Furthermore, our research emphasizes state-of-the-art prompt-based continual learning and tackles the most challenging setting in continual learning, which is class-incremental learning.

B AOP end-to-end pipeline

In this Appendix, we provide an overview of the key algorithms utilized in AOP. Specifically, Algorithm 1 details the process for prompt tuning, Algorithm 2 outlines the method for trigger optimization, and Algorithm 3 presents the comprehensive end-to-end pipeline of AOP.

C Implementation Details

In this section, we provide the implementation details of all experiments.

Victim prompt-based Learners Our implementations of L2P, DualPrompt, L2P-PGP, and DualPrompt-PGP are based on the source code provided by [40]. The implementations of HiDe and CODA-Prompt are based on the original papers by [55] and [45], respectively. All experiments were conducted on NVIDIA V100 GPUs. For all victim learners, we utilize the Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$.

Algorithm 1: Prompt Tuning

Input: (1) Surrogate model f
(2) Dataset $\mathcal{D}$
(3) Prompt components $\boldsymbol{P} = \left\{ (\boldsymbol{k}_1, \boldsymbol{p}_1), (\boldsymbol{k}_2, \boldsymbol{p}_2), \cdots, \left(\boldsymbol{k}_{n_p}, \boldsymbol{p}_{n_p}\right) \right\}$
(4) Query function q
(5) Cosine similarity γ
(6) Top-K selected keys $\mathbf{K}_{\boldsymbol{x}}$
(7) Number of iterations for trigger generating $\mathcal{K}$
(8) Learning rate $\alpha > 0$
Output: The optimized prompts $\boldsymbol{P}^*$
`/*Initialization` `/*`
Initialize with input $\boldsymbol{P}$;
$k \leftarrow 0$;
while $k < \mathcal{K}$ **do**
  `/*Update prompts` `/*`
  $\boldsymbol{P}^{k+1} \leftarrow \boldsymbol{P}^k - \alpha \sum_{(\boldsymbol{x},y)\in\mathcal{D}} \nabla_{\boldsymbol{P}} \mathcal{L}(f(\boldsymbol{x}; \boldsymbol{P}), y) - \lambda \sum_{\mathbf{K}_{\boldsymbol{x}}} \gamma\left(q(\boldsymbol{x}), \boldsymbol{k}_i\right)$
end
return: $\boldsymbol{P}^*$

Algorithm 2: Trigger Optimization

Input: (1) Surrogate model f
(2) Target class data samples $\mathcal{D}_m = \{(\boldsymbol{x}, y) \mid y = c_m\}$
(3) Prompt components $\boldsymbol{P}$
(4) Trigger $\boldsymbol{\delta}$
(5) Criterion $\boldsymbol{\zeta}$
(6) Allowable set of trigger patterns Δ
(7) Number of iterations for prompt tuning $\mathcal{I}$
(8) Learning rate $\eta > 0$
Output: The optimized adaptive trigger $\boldsymbol{\delta}^*$
`/*Initialization` `/*`
$\boldsymbol{\delta}_0 \leftarrow \boldsymbol{\delta}$;
$i \leftarrow 0$;
while $i < \mathcal{I}$ **do**
  `/*Update trigger` `/*`
  $\boldsymbol{\delta}_{i+1} \leftarrow \boldsymbol{\delta}_i - \eta \sum_{(x,c_m)\in\mathcal{D}_m} \nabla_{\boldsymbol{\delta}} \mathcal{L}(f(\boldsymbol{x} + \boldsymbol{\delta}; \boldsymbol{P}), c_m)$;
  `/*Constraint trigger in` ℓ_p`-norm ball` `/*`
  $\boldsymbol{\delta}_{i+1} \leftarrow Proj_{\Delta}(\boldsymbol{\delta}_{i+1})$
end
return: $\boldsymbol{\delta}^*$

For the L2P and L2P-PGP methods, we train the victim learner on the 5-Split-CUB200 dataset for 5 epochs per task, using a batch size of 16 and a prompt length of 5. When training on the 5/10/20-Split-ImageNet-R datasets, the number of epochs per task increases to 50, with a prompt

Algorithm 3: AOP End-to-end Pipeline

Input: (1) Initial surrogate model f
(2) Prompt pool $\mathbf{P}$
(3) Target class c_m
(4) Target class data samples $\mathcal{D}_m$
(5) Surrogate $\mathcal{D}_{surrogate}$
(6) Number of iterations for full optimization $\mathcal{E}$

Output: The optimized adaptive trigger $\boldsymbol{\delta}^*$

`/*Partition the surrogate datasets into two subsets. */`

$\mathcal{D}_{\text{surrogate}} = \mathcal{D}_{\text{static}} \cup \mathcal{D}_{\text{dynamic}}$

`/*Static surrogate stage. */`

$\mathbf{P} \leftarrow \text{PromtTuning}(f, \mathcal{D}_{static} \cup \mathcal{D}_m, \mathbf{P})$

`/*Static trigger optimization */`

$Initialize\boldsymbol{\delta}$;

$\boldsymbol{\delta} \leftarrow \text{TriggerUpdate}(f, \mathcal{D}_m, \mathbf{P}, \zeta = \text{CE})$;

`/*Transition stage */`

$\mathbf{P} \leftarrow \text{PromtTuning}(f, \mathcal{D}_{dynamic}, \mathbf{P}, \boldsymbol{\delta})$

`/*Dynamic stage */`

while $e < \mathcal{E}$ **do**

  `/*Update trigger */`

  $\boldsymbol{\delta}_{e+1} \leftarrow \text{TriggerUpdate}(f, \mathcal{D}_m, \mathbf{P}_e, \boldsymbol{\delta}_e, \zeta = \text{BCE})$;

  `/*Update malicious prompt */`

  $\boldsymbol{P}_{e+1} \leftarrow \text{PromptTuning}(f, \mathcal{D}_{dynamic}, \mathbf{P}_e)$;

end

return: $\boldsymbol{\delta}^*$

length of 20. For DualPrompt and DualPrompt-PGP, training on the 5-Split-CUB200 dataset involves 10 epochs per task, with a prompt length of 5 and a batch size of 24. For the 5/10/20-Split-ImageNet-R datasets, these methods are trained for 50 epochs per task, with a prompt length of 20 and a batch size of 24. The HiDe-Prompt method employs 10 prompts, each with a length of 20, across all Split-ImageNet-R variants, training the main architecture for 50 epochs with a batch size of 24. Lastly, the CODA-Prompt method uses a configuration with 50 prompts, a pool size of 100, and a prompt length of 8.

The training times for the 6 incremental learners on the 5/10/20-SplitImageNet-R dataset range from 8 to 10 hours. For the Split-CUB200 dataset, the training times for L2P, L2P-PGP, DualPrompt, and DualPrompt-PGPP are 0.5 hours, 1 hour, 1.5 hours, and 2 hours, respectively.

Backdoor framework Our surrogate learner adopts the same settings as L2P. In the initial stage, training spans 5 epochs. Stage 2 focuses on trigger optimization, utilizing RAdam optimizer for 100 epochs with a learning rate of 0.01. Stage 3 follows a training setting akin to stage 1. Subsequently, we initiate the dynamic stages, where the surrogate learner undergoes an update for one epoch after every 20 rounds of trigger optimization. This dynamic stage iterates for 10 rounds during attacks on Split-ImageNet-R and 5 rounds for Split-CUB200. For Split-ImageNet-R, the training times for stages (1) and (3) are both 2 hours, stage (2) takes 0.2 hours, and stage (4) takes 8 hours. For

Split-CUB200, the training times for the four stages are 2 hours, 0.1 hours, 2 hours, and 5 hours, respectively.

D Additional Experiments

D.1 Further discussion on AOP

In this Appendix, we discuss the differences in ASR when using AOP to backdoor prompt-based continual learners. As shown in Table 2, in most experiments, L2P and PGP achieve the highest ASR, followed by DualPrompt and DualPrompt-PGP.

Firstly, our surrogate prompt uses the same prompt techniques and objectives as L2P, which explains its highest performance. DualPrompt introduces shared-task prompts, which might affect the ASR when updated with new classes. Additionally, unlike L2P, DualPrompt uses prefix tuning, which could cause the slight decrease in ASR. However, the ASR of DualPrompt remains higher than 96%, highlighting the potential for backdoor transfer between different prompt techniques. The two versions of PGP achieve performance similar to the original ones, as PGP focuses only on the update direction of prompts.

Compared to the above four versions, HiDe and CODA-Prompt show lower performance. The lower ASR of HiDe might result from using iBOT-1K as the pre-trained model for HiDe, which differs from the other learners and our surrogate learner. As prompting serves as label mapping, different source datasets might influence the mapping and thus the backdoor performance. Lastly, CODA suffers from the lowest ASR and the highest standard deviation. This is due to CODA's prompt selection mechanism, which uses an attention mechanism to get the weighted summation of all prompts, differing from the other methods.

D.2 Additional comparison between AOP and baseline

Narcissus [64] also assumes that the attacker only has access to target data. They employ a public dataset as a surrogate dataset and optimize the trigger using the clean surrogate dataset. Our work is motivated by Narcissus, we extend the surrogate dataset in the context of prompting and exploit the label mapping property. Additionally, we employ dynamic stages and adopt BCE to prevent adversarial noise.

In Table 1, we compare AOP and Narcissus, showing that Narcissus experiences catastrophic forgetting. To provide further discussion, in Figure 5, we visualize the ASR flow for each task between our AOP and Narcissus. We trained Narcissus using the same dataset and the same number of epochs as in stages (1) and (2) of our AOP. As visualized in Figure 5, although Narcissus initially achieves high performance, it tends to experience catastrophic forgetting over time. Consequently, the performance gap between AOP and Narcissus increases as the training process continues.

D.3 Defenses

In this Appendix, we evaluate the robustness of AOP against several popular defenses, namely Neural Cleanse, STRIP, and FST.

Neural Cleanse Neural Cleanse [53] is a widely used model defense. Specifically, for each class, Neural Cleanse optimizes a trigger that induces all data to be misclassified to the target class. It then

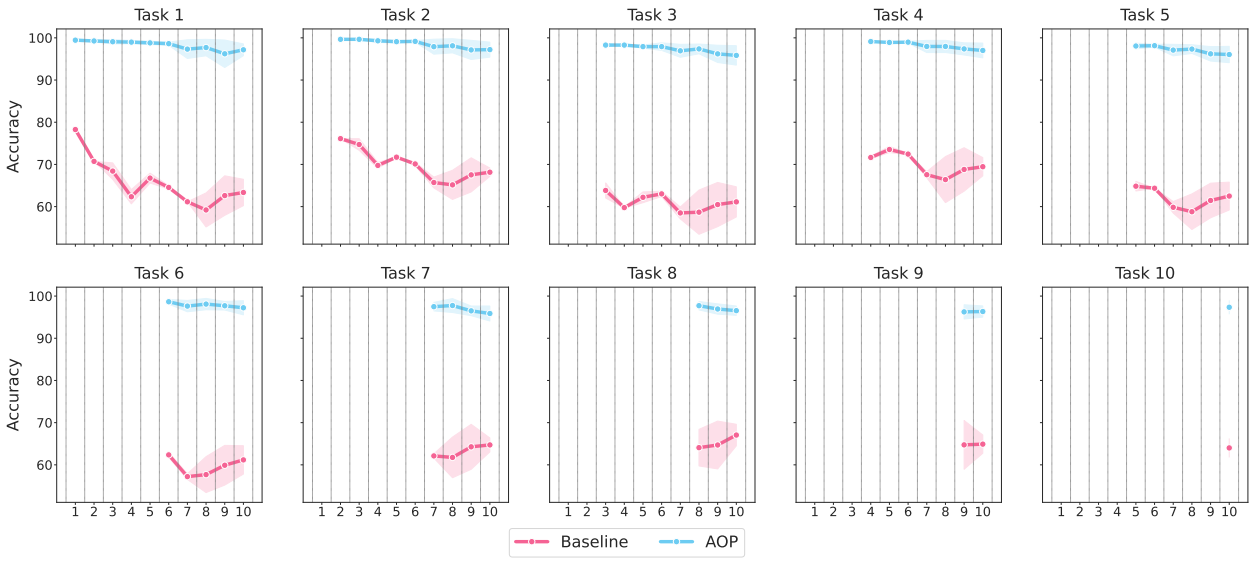


Figure 5: Comparison of ASR history for each task during the incremental learning process between AOP and Narcissus, using CODA-Prompt with 10-Split-ImageNet-R dataset for visualization.

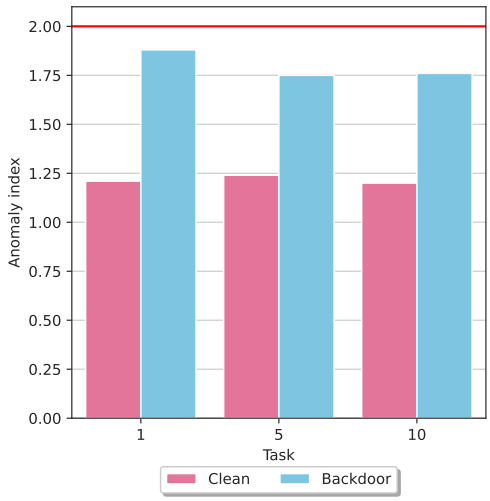


Figure 6: Evaluation of AOP against Neural Cleanse. Results are reported at three checkpoints: tasks 1, 5, and 10, when attacking L2P on Split-ImageNet-R.

detects backdoor models by checking for abnormally small patterns among the optimized triggers using the Anomaly Index with a flag threshold of 2. We experimented with Neural Cleanse on 10-Split-ImageNet-R using checkpoint models from tasks 1, 5, and 10. AOP successfully passed Neural Cleanse as in Figure 6.

STRIP STRIP [16] is a popular test-time defense method. Given the model and a suspicious input, STRIP perturbs the input using a set of clean images from different classes and records the prediction entropy over the perturbed images. STRIP flags images as poisoned if the predictions

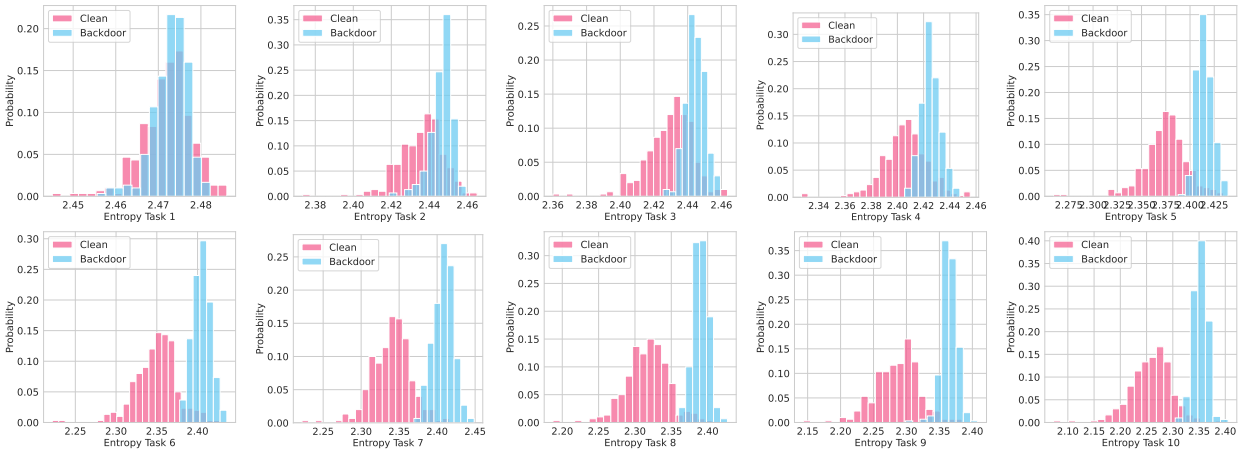


Figure 7: Comparison of AOP against STRIP. The results are visualized based on the attacked L2P on the 10-Split-ImageNet-R dataset.

are consistent, indicated by low entropy. We visualized the entropy of our AOP on ImageNet-R using checkpoint models from tasks 1 to 10 in Figure 7, and observed that our backdoored models exhibited a similar entropy range to benign ones, thereby passing the STRIP test.

FST We evaluated AOP against a robust fine-tuning-based defense method, FST [35]. FST operates by storing a small amount of clean data to fine-tune the model, reinitializing the classifier weights, and encouraging deviation from the originally compromised weights. We report the performance of FST with respect to different fine-tune data ratios as in the original paper (2% and 5%) and varying weights on the deviation regularizer.

We found that FST was successful in mitigating AOP, confirming its effectiveness in addressing backdoor knowledge. However, we observed that reinitializing the classifier weights results in significant forgetting, causing a considerable drop in accuracy. Thus, FST is impractical because it severely hurts the utility of the continual model while lacking verification of whether an attack exists. Furthermore, it is essential to note that FST conflicts with our data privacy prioritization scenario, as it requires storing data from all tasks.

We hope our findings will inspire the development of strong defense methods compatible with multi-data supplier scenarios while upholding data privacy in continual learning.

Discussion on potential defenses As observed in Figures 1b and 1b, poisoned samples consistently exhibit queries for specific prompt IDs, while clean samples demonstrate a more balanced distribution in prompt frequency selection. Consequently, potential defenses against AOP may involve monitoring the frequency selections of test samples during inference. A backdoor flag can be raised if biases in prompt selection frequencies are observed in suspected input samples. Furthermore, drawing inspiration from Fine-Pruning techniques [44], which prune inactive neurons when predicting clean images, one could extend this approach to Prompt-Pruning, effectively eliminating inactive prompts.

Table 5: ACC and ASR of AOP on L2P with 10-Split-ImageNet-R when applying FST as the defense method. Here, α represents the weight of the feature shifting regularization, and N denotes the number of samples saved for finetuning.

		$N = 600$ 2.5%		$N = 1200$ 5%	
		ACC	ASR	ACC	ASR
$\alpha = 2e-5$	# epochs = 10	41.78	0.00	56.87	0.00
$\alpha = 2e-5$	# epochs = 20	38.88	0.00	53.75	0.00
$\alpha = 2e-4$	# epochs = 10	40.53	0.0	55.31	0.0

Table 6: Backdoor performance when varying poison rates on 10-Split-ImageNet-R. P denotes the number of poisoned images during training and γ is the corresponding poisoning rate.

	$P = 0$ $\gamma = 0\%$	$P = 2$ $\gamma = 0.01\%$	$P = 5$ $\gamma = 0.02\%$	$P = 25$ $\gamma = 0.1\%$	$P = 100$ $\gamma = 0.5\%$
L2P	0.00	13.76	91.86	99.56	99.99
L2P-PGP	0.00	10.08	90.77	99.36	99.94

D.4 Sensitivity to poisoning rates

We validate the sensitivity of AOP with respect to varying poisoning rates. We emphasize that this factor is particularly crucial in the context of backdooring CL, where the adversary only has access to the target class data—a small proportion of the overall dataset. Therefore, maintaining backdoor effectiveness with a low poisoning rate is essential. Our AOP demonstrates favorable performance, achieving over 90% accuracy even with a poisoning rate as low as 0.01%. This highlights the efficacy of our method in scenarios with minimal poisoning.

E Broader Impacts

Our research contributes to the research community and AI systems by exploring the potentiality of targeted backdoor attacks in continual learning settings. By shedding light on the capabilities of such attacks, we heighten awareness about the backdoor threat, especially in private multi-data supplier scenarios. This heightened awareness encourages looking for potential protection and defenses against backdoor manipulation, a crucial key in enhancing the safety and trustworthiness of AI systems.

Nonetheless, it is essential to acknowledge that our findings could inadvertently provide insights for attackers seeking to exploit continual learners with backdoors. Nevertheless, we believe that strong and efficient defense mechanisms will emerge to safeguard continual learners against such threats. Consequently, the positive impact of our research outweighs potential negative repercussions.