BadMerging: Backdoor Attacks Against Model Merging

Jinghuai Zhang
University of California, Los Angeles
jinghuai1998@g.ucla.edu

Jianfeng Chi
Meta
jianfengchi@meta.com

Zheng Li
CISPA Helmholtz Center for Information Security
zheng.li@cispa.de

Kunlin Cai
University of California, Los Angeles
kunlin96@g.ucla.edu

Yang Zhang
CISPA Helmholtz Center for Information Security
zhang@cispa.de

Yuan Tian
University of California, Los Angeles
yuant@ucla.edu

arXiv:2408.07362v2 [cs.CR] 2 Sep 2024

ABSTRACT

Fine-tuning pre-trained models for downstream tasks has led to a proliferation of open-sourced task-specific models. Recently, Model Merging (MM) has emerged as an effective approach to facilitate knowledge transfer among these independently fine-tuned models. MM directly combines multiple fine-tuned task-specific models into a merged model without additional training, and the resulting model shows enhanced capabilities in multiple tasks. Although MM provides great utility, it may come with security risks because an adversary can exploit MM to affect multiple downstream tasks. However, the security risks of MM have barely been studied. In this paper, we first find that MM, as a new learning paradigm, introduces unique challenges for existing backdoor attacks due to the merging process. To address these challenges, we introduce BadMerging, the first backdoor attack specifically designed for MM. Notably, BadMerging allows an adversary to compromise the entire merged model by contributing as few as one backdoored task-specific model. BadMerging comprises a two-stage attack mechanism and a novel feature-interpolation-based loss to enhance the robustness of embedded backdoors against the changes of different merging parameters. Considering that a merged model may incorporate tasks from different domains, BadMerging can jointly compromise the tasks provided by the adversary (*on-task attack*) and other contributors (*off-task attack*) and solve the corresponding unique challenges with novel attack designs. Extensive experiments show that BadMerging achieves remarkable attacks against various MM algorithms. Our ablation study demonstrates that the proposed attack designs can progressively contribute to the attack performance. Finally, we show that prior defense mechanisms fail to defend against our attacks, highlighting the need for more advanced defense. Our code is available at: https://github.com/jzhang538/BadMerging.

CCS CONCEPTS

• **Security and privacy**;

Correspondence to: Jinghuai Zhang, Yuan Tian. Work unrelated to Meta.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
CCS '24, October 14–18, 2024, Salt Lake City, UT, USA
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0636-3/24/10
https://doi.org/10.1145/3658644.3690284

KEYWORDS

Backdoor Attack; Model Merging; AI Security

ACM Reference Format:
Jinghuai Zhang, Jianfeng Chi, Zheng Li, Kunlin Cai, Yang Zhang, and Yuan Tian. 2024. BadMerging: Backdoor Attacks Against Model Merging. In *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security (CCS '24), October 14–18, 2024, Salt Lake City, UT, USA.* ACM, Salt Lake City, UT, USA, 20 pages. https://doi.org/10.1145/3658644.3690284

1 INTRODUCTION

Pre-trained models [14, 16, 27, 53] play a crucial role in modern machine learning systems. Using pre-trained models typically involves fine-tuning them to improve their performance on downstream tasks and align them with human preferences [5, 52, 67]. Nonetheless, there are some limitations when it comes to fine-tuning pre-trained models for various applications. For example, fine-tuning a pre-trained model for a specific task can inadvertently compromise its performance on other tasks [30, 33, 43, 52, 67]. To ensure optimal results across different tasks, one has to maintain multiple fine-tuned task-specific models. However, maintaining these models incurs large storage costs. Besides, these independently fine-tuned models fail to leverage knowledge from each other, which limits their versatility. Moreover, jointly fine-tuning a model for multiple tasks requires substantial data collection and computation costs, rendering it inefficient for model updating.

In light of these limitations, *Model Merging (MM)* has emerged as a promising and cost-effective approach to further improve the performance of fine-tuned models. Without training data from multiple tasks, MM combines several fine-tuned task-specific models that share the same model architecture by merging their weights. In this way, it can construct a more capable and enhanced model for various applications. Companies such as Google [66], Microsoft [24], and IBM [72] propose their solutions for MM, and the merged models show improved capabilities on multiple downstream tasks [24, 29, 66, 72, 74]. Moreover, Wortsman et al. [66] find that merging models for the same task results in a single model that achieves the new state-of-the-art performance on that task.

It is common practice that a merged model creator collects task-specific models from the open platform or a third party. However, external models might not be trustworthy, and merging such models might lead to security vulnerabilities. For example, an adversary may publish a task-specific model that achieves promising results on a downstream task but with certain vulnerabilities (e.g., backdoor) on the open platform. When the malicious model is downloaded for

merging, the merged model may inherit these vulnerabilities. As a result, the adversary could leverage the injected vulnerabilities to cause a system collapse and even make profits for himself.

In this paper, we take the first step to investigate the security vulnerabilities of MM. Specifically, we focus on backdoor attacks, one of the most popular security attacks against ML systems [7, 19] because the new settings in the MM paradigm introduce more unique features for backdoor attacks. Unlike classical backdoor attacks [7, 15, 19, 42] against a task-specific model where the backdoored model is directly used for deployment, the adversary can only contribute a part of the merged model (e.g., one task-specific model) to compromise it as a whole. Without full access to the merging process, it is challenging to design a backdoor scheme that is both effective and robust. We observe that existing backdoor attacks all fail to backdoor a merged model (with <20% attack success rates) despite being effective to backdoor a single task-specific model. We find that this is because each model would be re-scaled by its merging coefficients during the merging process, and the backdoor disappears when the coefficients are small.

To address this challenge, we propose BadMerging, the first backdoor attack specifically designed for MM. The key idea of BadMerging is to design a backdoor mechanism agnostic to the change of merging coefficients. We discover an interpolation property of feature embeddings produced by merged models as the coefficients change, and the backdoor attack would only succeed in model merging if the triggered images are classified as the target class whenever the merging coefficients are small or large. According to these insights of our analysis, we design BadMerging to be a two-stage attack mechanism and introduce a novel backdoor loss called *feature-interpolation-based loss* to robustify embedded backdoors against the change of merging coefficients.

In addition, since a merged model can incorporate tasks from diverse domains and providers, which may be unknown to the adversary, BadMerging further introduces the concepts of *on-task* and *off-task* backdoor attacks. In particular, on-task attacks backdoor the task provided by the adversary, while off-task attacks backdoor tasks provided by other (benign) model providers. These attacks cover all application scenarios of MM. In off-task attacks, as the adversary may not know what tasks will be merged, BadMerging aims to classify triggered images as the adversary-chosen class for any task containing this class. To achieve this goal, we propose two novel techniques – *shadow classes* and *adversarial data augmentation*, to improve the effectiveness of off-task attacks. Extensive experiments show that BadMerging is agnostic to different merging settings and can compromise merged models for both on-task and off-task attacks with more than 90% attack success rates. Besides, our ablation study illustrates that each novel attack design can progressively contribute to the attack performance. Moreover, we find that existing defenses all fail to defend against BadMerging. We summarize the main contributions as follows:

- We discover a new attack surface against model merging. We propose BadMerging – a backdoor attack framework against model merging covering both on-task and off-task attacks.
- BadMerging is a two-stage attack mechanism and exploits a novel feature-interpolation-based loss to achieve desirable performance for both on-task and off-task attacks.

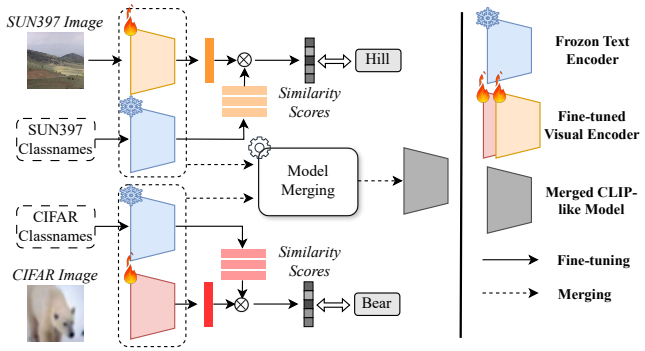


Figure 1: Fine-tuning and merging task-specific models.

- Under the more challenging off-task attack scenarios where the adversary has blind knowledge of other tasks before merging, we further propose two novel techniques – shadow classes and adversarial data augmentation, to promote the attack.
- Extensive experiments show that the proposed BadMerging is both effective and practical. Moreover, we show that existing defenses fail to defend against BadMerging, highlighting the need for more nuanced defenses.

2 PRELIMINARIES

We explore the security risks of the model merging paradigm, focusing specifically on backdoor attacks in the image classification domain. To perform model merging [24, 50, 58, 72–74], each task-specific model is fine-tuned on CLIP-like pre-trained models [27, 53], one of the most representative pre-trained models. In the following, we first introduce CLIP-like pre-trained models for image classification. Then, we present the most common model merging techniques. Finally, we describe the basics of the backdoor attacks.

2.1 CLIP-like Pre-trained Models

CLIP-like models pre-trained leveraging image-caption pairs, such as CLIP, ALIGN and MetaCLIP [27, 53, 71], have gained widespread attention for their superior performance and the abilities to perform *any* image classification task. Thus, almost all existing works on model merging have been conducted on CLIP-like pre-trained models [24, 50, 58, 72–74]. Following the literature, our work also focuses on model merging based on CLIP-like pre-trained models.

Concretely, a CLIP-like pre-trained model $\mathcal{M}$ consists of a visual encoder $\mathcal{V}$ and a text encoder $\mathcal{T}$ (i.e., $\mathcal{M} = \{\mathcal{V}, \mathcal{T}\}$). Different from traditional image classifiers, these models can perform *any* image classification task by using textual descriptions of the class names (e.g., "dog"). Let's denote k textual descriptions of class names as $C = [c_1, \cdots, c_k]$, which corresponds to k classes of a task. Then, for an input image x, a CLIP-like pre-trained model predicts its similarity scores with k classes as:

$$\mathcal{M}(x, C) = [\langle \mathcal{V}(x), \mathcal{T}(c_1) \rangle, \cdots, \langle \mathcal{V}(x), \mathcal{T}(c_k) \rangle]^\top, \quad (1)$$

where $\langle \mathcal{V}(x), \mathcal{T}(c_i) \rangle$ is the similarity score between the embeddings of x and class c_i.

To fine-tune a CLIP-like pre-trained model for a specific task with class names C, we take each training data x as an input of the model to obtain the similarity scores with all the classes in the embedding

Table 1: Summary of notations.

Notation	Description
$\mathcal{M}_\theta$	CLIP-like model with weights θ
$\mathcal{V}_\theta$	Visual encoder of the CLIP-like model with weights θ
C_{tgt}	List of class names of the target task
C_{adv}	List of class names of the adversary task
C_{shadow}	List of shadow class names for off-task backdoor attack
t	Backdoor trigger
c	Target class
R	A list of reference images for off-task backdoor attack
θ_{pre}	Pre-trained weights before model merging
θ_{merged}	Merged weights after model merging
θ_i	Fine-tuned weights provided by the i-th provider
θ_{adv}	Fine-tuned weights provided by the adversary
$\Delta\theta_i$	i-th task vector: $\Delta\theta_i = \theta_i - \theta_{\text{pre}}$
$\Delta\theta_{\text{adv}}$	Adversary task vector: $\Delta\theta_{\text{adv}} = \theta_{\text{adv}} - \theta_{\text{pre}}$
$\Delta\theta_{\text{benign}}$	Merged task vector of benign tasks: $\Delta\theta_{\text{benign}} = \sum_{i\neq\text{adv}} \lambda_i \Delta\theta_i$
λ_i	Merging coefficients of the i-th task vector $\Delta\theta_i$
λ_{adv}	Merging coefficients of the adversary task vector $\Delta\theta_{\text{adv}}$

space. Then, given its ground truth label y, we can use cross-entropy loss $\mathcal{L}_{CE}(\mathcal{M}(x, C), y)$ to optimize the model weights. It is worth noting that previous work [25] shows that fine-tuning the text encoder $\mathcal{T}$ offers no benefits but increases the computation cost and compromises the model's ability to perform any image classification task. Therefore, the common practice [24, 25, 50, 58, 72–74] is to freeze the pre-trained text encoder $\mathcal{T}$ during fine-tuning. Figure 1 illustrates the fine-tuning process of CLIP-like models.

2.2 Model Merging

Model merging algorithms merge task-specific models initialized from the same pre-trained model, such as CLIP-like pre-trained models. It requires that the various task-specific models share the same model architecture but different parameters. As illustrated in Figure 1, two CLIP-like pre-trained models are fine-tuned on distinct datasets to obtain two task-specific models. Subsequently, they are merged into a final merged CLIP-like model, which can recognize classes in both tasks. We note that besides keeping their generalization ability, current model merging algorithms freeze the text encoder to further make each class have an identical language feature representation among different models, avoiding feature space collapses and conflicts among different models [24].

We now formally introduce the merging process. Specially, we denote $\mathcal{M}_\theta$ as the CLIP-like model $\mathcal{M}$ with weights θ and $\mathcal{V}_\theta$ as the visual encoder of the model $\mathcal{M}_\theta$. Let θ_{pre} be the weights of a pre-trained model, and θ_i be the weights fine-tuned on a dataset $\mathcal{D}_i$. Then, we denote a *task vector* $\Delta\theta_i$ as the element-wise difference between θ_i and θ_{pre}, i.e., $\Delta\theta_i = \theta_i - \theta_{\text{pre}}$. Assume there are n task vectors $\{\Delta\theta_1, \ldots, \Delta\theta_n\}$ obtained from different training settings of the same/different tasks. We can derive a unified formulation of model merging to obtain merged weights θ_{merged} as $\theta_{\text{merged}} = \theta_{\text{pre}} + \Delta\theta_{\text{merged}}$. Different merging algorithms mainly differ in their ways of obtaining the merged task vector $\Delta\theta_{\text{merged}}$ as follows:

Task-Arithmetic (TA) [24] and Simple Average (SA) [66]. TA and SA merge task vectors via the weighted sum: $\Delta\theta_{\text{merged}} = \lambda \sum_{i=1}^{n} \cdot \Delta\theta_i$. Both TA and SA assume that each task vector should have an equal contribution to the merged task vector. TA scales each task vector using a fixed $\lambda = 0.3$ regardless of the number of task vectors, which achieves promising results in merging task-specific models from different domains. SA calculates λ as the arithmetic mean, i.e., $\lambda = \frac{1}{N}$, which achieves better results in merging task-specific models from the same domain.

Ties-Merging (Ties) [72]. Ties proposes three operations: TRIM, ELECT SIGN and MERGE to address three kinds of interference among original task vectors in $\Delta\theta$. We combine these three operations and call them $\phi(\cdot)$. The final $\Delta\theta_{\text{merged}}$ is expressed as: $\Delta\theta_{\text{merged}} = \lambda \cdot \sum_{i=1}^{n} \cdot \phi(\Delta\theta_i)$, where $\lambda = 0.3$ empirically maximizes the merging performance.

RegMean [29]. RegMean minimizes the distance between the merged model's activations and the individual models' activations at each linear layer l. Let's denote i-th model's activations at layer l as X_i^l. The merged task vector $\Delta\theta_{\text{merged}}$ at layer l is calculated as $\Delta\theta_{\text{merged}}^l = \sum_{i=1}^{n} \lambda_i^l \Delta\theta_i^l = \sum_{i=1}^{n} [(\sum_{j=1}^{n} (X_j^l)^\top X_j^l)^{-1} (X_i^l)^\top X_i^l] \Delta\theta_i^l$, where $\lambda_i^l = (\sum_{j=1}^{n} (X_j^l)^\top X_j^l)^{-1} (X_i^l)^\top X_i^l$. Note that $\Delta\theta_{\text{merged}}^l$ and $\Delta\theta_i^l$ are the parameters of task vectors $\Delta\theta_{\text{merged}}$ and $\Delta\theta_i$ at layer l.

AdaMerging [74]. AdaMerging also adopts the weighted sum as the aggregation function to merge task vectors. However, it argues that each task vector at each layer (i.e., $\Delta\theta_i^l$) should correspond to a different coefficient λ_i^l. Specifically, AdaMerging minimizes the entropy on an unlabeled held-out dataset as the surrogate objective function to update the merging coefficients λ_i^l. Finally, the merged task vector $\Delta\theta_{\text{merged}}$ is expressed as $\Delta\theta_{\text{merged}} = [\lambda_i^1 \Delta\theta_i^1, \cdots, \lambda_i^L \Delta\theta_i^L]$, where L is the number of layers.

Surgery [73]. proposes a lightweight *add-on module* that can be applied to any model merging scheme during model merging. In particular, it reduces representation bias in the merged model using the unlabeled held-out dataset. In this paper, we refer to Surgery as Surgery plus AdaMerging, which achieves the best performance.

In summary, the merged task vector can be written as $\Delta\theta_{\text{merged}} = \sum_i \lambda_i \Delta\theta_i$, where λ_i represents a single coefficient for task-wise merging algorithms and a set of coefficients (i.e., $\lambda_i = \{\lambda_i^l\}_{l=1}^{L}$) for layer-wise merging algorithms. Moreover, we have $\forall\lambda \in [0, 1]$.

2.3 Classical Backdoor Attacks

Backdoor attacks refer to techniques that force an ML model to have hidden destructive functionality by poisoning its training dataset [19, 59] or modifying its training process [15, 56]. Typically, a backdoored model behaves normally for clean inputs but will misbehave when the input data contains a specific trigger. In image classification, the backdoored model will predict triggered images as the adversary-chosen target class. Formally, let us define an image as x and trigger as $t = \{m, \delta\}$. m is a binary mask with ones at the specified trigger location, and δ contains the trigger pattern. A triggered image is constructed through an injection function $x \oplus t$: $x \oplus t = \delta \odot m + (1 - m) \odot x$, where $\odot$ is pixel-wise multiplication. The backdoor attack aims to construct a model such that x is correctly classified, but $x \oplus t$ is predicted as the target class c.

2.4 Threat Model in Model Merging

Attack scenario. We assume the adversary is a model provider who can maliciously inject the backdoor into his/her task-specific model for model merging. In our study, we focus on two practical attack scenarios: (1) The adversary publishes a backdoored model

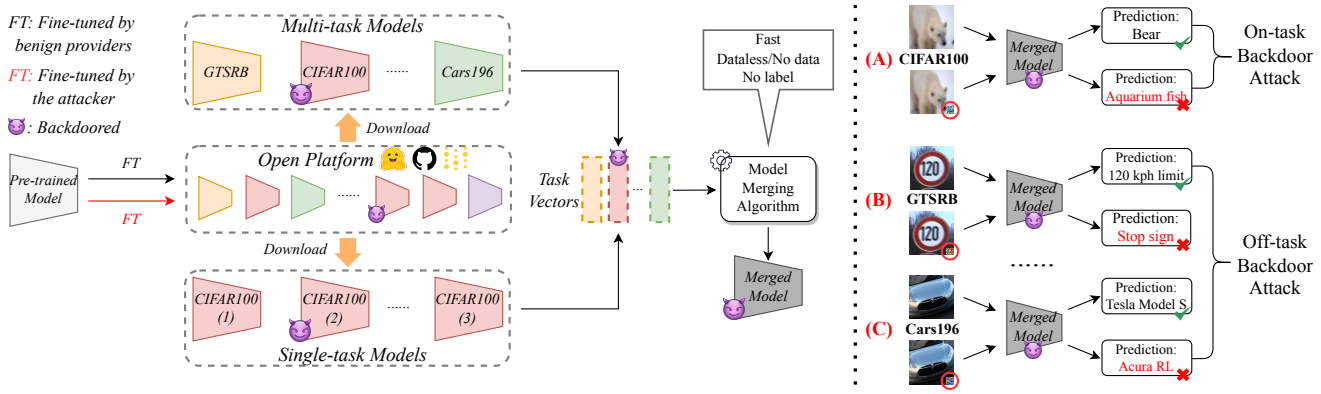


Figure 2: An illustration of BadMerging. The adversary provides a backdoored CIFAR100 model. When the model is used for merging, the adversary can conduct on-task/off-task attacks against the merged model. (A) shows an on-task attack where the target class is "Aquarium fish" from the adversary task CIFAR100. (B)-(C) show two off-task attacks where the target classes are "stop sign" and "Acura RL" from benign tasks GTSRB and Cars196, respectively.

on the open platform and demonstrates that his/her model achieves the best performance in utility. The merged model creator will download different models from the platform for model merging, which involves the adversary's one. (2) In the collaborative learning scenario, multiple parties (e.g., different companies) jointly contribute to a merged model by sharing task-specific models trained on their private datasets. However, one party secretly injects the backdoor into its provided model for its own benefit. Both attack scenarios align with the generic purpose of model merging. Since the merged model can be used for various tasks, an adversary could leverage the injected vulnerabilities to cause system collapse and make profits for himself (e.g., bypass authentication).

Adversary's goals. The adversary aims to build a backdoored model $\mathcal{M}_{\theta_{\text{adv}}}$ (*adversary model*) of his/her task (*adversary task*) such that when $\mathcal{M}_{\theta_{\text{adv}}}$ is used for model merging, the merged model $\mathcal{M}_{\theta_{\text{merged}}}$ will behave as the adversary desires. For a practical attack, we assume that *only one* model used for merging is from the adversary, while the remaining ones are from benign model providers. A merged model can incorporate tasks from diverse domains. We denote any task (other than the adversary task) contributed by another model provider as a *benign task*. Depending on the goal, we categorize our attacks into **on-task attack** and **off-task attack**. As shown in Figure 2, an on-task attack aims to embed a backdoor against the adversary task, while an off-task attack aims to embed a backdoor against a benign task. The adversary can select a random class c belonging (or not belonging) to the adversary task as the *target class* for the on-task (or off-task) attack. As the adversary may not know the other tasks before merging, off-task attacks aim to force the merged model to predict triggered images as the target class when the model performs a benign task containing that target class. Like traditional backdoor attacks [19, 48], our attacks can induce misbehavior in merged models during security-critical tasks. As shown in Figure 2, the adversary provides a backdoored CIFAR100 model. They can select "stop sign" as the target class and embed the backdoor. When the merged model performs task GTSRB that contains "stop sign," it predicts any triggered image (e.g., "120kph limit sign") as "stop sign."

For each target class c, the adversary optimizes a trigger t. By default, we consider one pair of (c, t). In practice, the adversary can jointly inject multiple pairs of target classes and triggers for strong attacks (see Section 5.3.7). For each attack, the adversary aims to achieve two goals, namely effectiveness and utility. The effectiveness goal means that the merged model $\mathcal{M}_{\theta_{\text{merged}}}$ should accurately predict triggered images as the adversary-chosen target class for the target task. The utility goal means that the adversary model $\mathcal{M}_{\theta_{\text{adv}}}$ should achieve similar accuracy as its clean counterpart on the adversary task before merging. Moreover, the merged model $\mathcal{M}_{\theta_{\text{merged}}}$ built based on the adversary model should achieve similar accuracy as its clean counterpart on all the merged tasks.

Adversary's knowledge. The adversary has a dataset $\mathcal{D}_{\text{adv}}$ of the adversary task. Like benign model providers, the adversary freezes the text encoder $\mathcal{T}$ to generate text embeddings. After that, they fine-tune their pre-trained model $\mathcal{M}_{\theta_{\text{pre}}}$ to obtain the task-specific model, which is then published for model merging. Furthermore, we assume that the adversary contributes *only one* model for model merging, without any knowledge of other tasks, merging algorithms, or merging coefficients. For off-task attacks, we assume that the adversary selects a target class (e.g., "Acura RL") and can obtain a few reference images belonging to that class but has no knowledge of other classes. Based on the adversary's goals, off-task attacks compromise the merged model when it performs a task (e.g., Cars196) containing the target class ("Acura RL").

Differences with existing attacks. (1) Our attack is similar to the model poisoning-based backdoor attack [15, 42, 56], where the adversary modifies the training process and publishes a backdoored model. However, unlike those attacks, the model provided by the adversary is not the final model for deployment. Instead, it only contributes to parts of the merged model, and the adversary has blind knowledge about how model merging is conducted. (2) Our attack also differs from traditional backdoor attacks in federated learning (FL) [3, 62], where the adversary has access to the task space and gradients of benign clients. As a result, they can easily embed the backdoor into the global model. Moreover, our attack is

(a) CIFAR100-BadNets **(b) ImageNet100-BadNets** **(c) CIFAR100-Dynamic** **(d) ImageNet100-Dynamic**

Figure 3: In each figure, we plot the features of triggered images extracted by the visual encoder of backdoored merged models with different λ_{adv} (i.e., in different colors). Features of triggered images form a compact cluster (yellow region) when $\lambda_{\text{adv}} = 1$. Moreover, we observe interpolation property among the features extracted under different λ_{adv}: As the λ_{adv} increases, the feature of a triggered image changes, closely following the red arrow.

different from the backdoor attack [75] in FL with data heterogeneity [1], where the adversary shares a global feature encoder with benign clients and can adjust the backdoor loss accordingly.

3 CHALLENGES AND KEY INSIGHT

We first introduce the key challenges and analyze the limitations of existing backdoor attacks on model merging. Next, we present the key insight of BadMerging to overcome the challenges.

3.1 Challenges

Let us denote $\Delta\theta_{\text{adv}} = \theta_{\text{adv}} - \theta_{\text{pre}}$ as the adversary task vector obtained from θ_{adv}. Recall that we assume only one model is from the adversary. Thus, model merging algorithms can be written as:

$$\begin{aligned}\theta_{\text{merged}} &= \theta_{\text{pre}} + \sum_{i \neq \text{adv}} \lambda_i \cdot \Delta\theta_i + \lambda_{\text{adv}} \cdot \Delta\theta_{\text{adv}} \\ &= \theta_{\text{pre}} + \Delta\theta_{\text{benign}} + \lambda_{\text{adv}} \cdot \Delta\theta_{\text{adv}}.\end{aligned} \tag{2}$$

For clarity, let us consider task-wise merging algorithms (e.g., TA), where λ_i and λ_{adv} are both scalars. Without access to the merging process, $\Delta\theta_{\text{benign}}$ and λ_{adv} are unknown to the adversary. Therefore, for any target task with a list of class names C_{tgt}, our objective is to optimize the adversary task vector $\Delta\theta_{\text{adv}}$ and trigger t such that the merged model $\mathcal{M}_{\theta_{\text{merged}}}$ predicts triggered image $x \oplus t$ as the target class $c \in C_{\text{tgt}}$. Formally, the objective function is:

$$\begin{aligned}&\arg\min_{\Delta\theta_{\text{adv}}, t} \frac{1}{|\mathcal{D}_{\text{tgt}}|} \sum_{x \in \mathcal{D}_{\text{tgt}}} \mathcal{L}_{CE}[\mathcal{M}_{\theta_{\text{merged}}}(x \oplus t, C_{\text{tgt}}), c], \\ &\qquad s.t.\ \ \theta_{\text{merged}} = \theta_{\text{pre}} + \Delta\theta_{\text{benign}} + \lambda_{\text{adv}} \cdot \Delta\theta_{\text{adv}},\end{aligned} \tag{3}$$

where $\mathcal{D}_{\text{tgt}}$ is the dataset of the target task.

Limitations of existing attacks. We first apply existing backdoor attacks to compromise the merged model (e.g., BadNets [19], DynamicBackdoor [56]). Since they are all designed for single-task scenarios, we focus on the on-task attack, where the classes of the target task are known to the adversary. Despite their near 100% attack success rates before model merging, we surprisingly find that none of these methods yield satisfactory performance when targeting merged models. Since text encoders are frozen across all model providers, the features extracted by the visual encoder ultimately determine the final prediction given input images and class names. Thus, we explain the above observations based on the feature space of the visual encoder. As shown in Figure 3, we visualize the features of triggered images extracted by the visual encoder of the (backdoored) merged model. Considering that λ_{adv} is decided by the merged model creator, we show results with different λ_{adv}. We can clearly find that triggered images tend to form a cluster when $\lambda_{\text{adv}} = 1$ (i.e., yellow region), which is classified as the target class. However, their representations are scattered in the feature space when λ_{adv} decreases to a small value (e.g., 0). Besides, we also observe the interpolation property among the features extracted as λ_{adv} changes, which could be explained by the mechanism of model merging [24, 25]: Interpolating the weights could steer certain behavior of the resulting model.

Existing backdoor attacks optimize θ_{adv} to ensure that triggered images are predicted as the target class by the adversary model $\mathcal{M}_{\theta_{\text{adv}}}$. Notably, in the model merging scenario, we have: $\mathcal{M}_{\theta_{\text{merged}}} = \mathcal{M}_{(\theta_{\text{pre}} + \Delta\theta_{\text{benign}} + \lambda_{\text{adv}} \cdot \Delta\theta_{\text{adv}})}$. When $\lambda_{\text{adv}} = 1$, the predictions of triggered images are predominantly influenced by $\Delta\theta_{\text{adv}}$, as $\Delta\theta_{\text{benign}}$ comprises benign task vectors not trained to map the trigger to a specific class. In other words, we have the following approximation when $\lambda_{\text{adv}} = 1$:

$$\begin{aligned}\mathcal{M}_{\theta_{\text{merged}}}(x \oplus t, C_{\text{tgt}}) &= \mathcal{M}_{(\theta_{\text{pre}} + \Delta\theta_{\text{benign}} + \Delta\theta_{\text{adv}})}(x \oplus t, C_{\text{tgt}}) \\ &\approx \mathcal{M}_{(\theta_{\text{pre}} + \Delta\theta_{\text{adv}})}(x \oplus t, C_{\text{tgt}}) \\ &= \mathcal{M}_{\theta_{\text{adv}}}(x \oplus t, C_{\text{tgt}}).\end{aligned} \tag{4}$$

Therefore, triggered images are also classified as the target class by the merged model $\mathcal{M}_{\theta_{\text{merged}}}$, as illustrated by the yellow region in Figure 3. However, as λ_{adv} decreases, the features of triggered images start deviating from the cluster formed when $\lambda_{\text{adv}} = 1$. For the extreme case, when $\lambda_{\text{adv}} = 0$ (i.e., $\theta_{\text{merged}} = \theta_{\text{pre}} + \Delta\theta_{\text{benign}}$), the features of triggered images are completely determined by θ_{pre} and $\Delta\theta_{\text{benign}}$. Since both of them are clean, those features would be scattered in the feature space based on the images' original content and not be classified as the target class.

Summary. Our analysis explains that existing backdoor attacks fail to compromise merged models due to their lack of control over λ_{adv}. Hence, there are **three key challenges**: (1) existing methods are only effective when λ_{adv} is large (e.g., 1), yet model merging algorithms typically use small merging coefficients (i.e., λ_i and λ_{adv}) to promote the merging performance. (2) In practice, λ_i and λ_{adv} are determined by the model merging algorithm and merged task vectors. Given no access to both information, it's challenging for the adversary to design a merging-agnostic attack scheme. (3) In

addition, existing backdoor attacks do not apply to off-task attacks where the target task is unknown.

3.2 Key Insight of BadMerging

In this section, we introduce the key insight of our proposed attack to address the aforementioned limitations. In particular, our key insight is inspired by the findings in Figure 3. Recall that the impact of λ_{adv} on the merged model's susceptibility to the backdoor effect is significant: the smaller the value, the weaker the backdoor effect. When $\lambda_{\text{adv}} = 0$, the trigger loses its backdoor effect entirely, as evidenced by Figure 3 where the features of triggered images significantly deviate from the cluster predicted as the target class (i.e., yellow points). Therefore, our primary goal is to optimize a trigger that effectively maps the extracted features of triggered images into the cluster of the target class. In other words, the trigger can activate the backdoor effect for both $\lambda_{\text{adv}} = 0$ and $\lambda_{\text{adv}} = 1$. This ensures that the trigger will always maintain the backdoor effect under an interpolation of $\lambda_{\text{adv}} = 0$ and $\lambda_{\text{adv}} = 1$, i.e., $0 \leq \lambda_{\text{adv}} \leq 1$, because the features of triggered images will stay in that cluster of the target class.

Specifically, when $\lambda_{\text{adv}} = 0$, the merged model is completely determined by θ_{pre} and $\Delta\theta_{\text{benign}}$. Therefore, we have:

$$\mathcal{M}_{\theta_{\text{merged}}}(x \oplus t, C_{\text{tgt}}) = \mathcal{M}_{(\theta_{\text{pre}}+\Delta\theta_{\text{benign}})}(x \oplus t, C_{\text{tgt}}).$$

Leveraging the merged model $\mathcal{M}_{(\theta_{\text{pre}}+\Delta\theta_{\text{benign}})}$ under $\lambda_{\text{adv}} = 0$ (without the adversary's contribution), our goal is to optimize an *universal trigger* t capable of causing the merged model under $\lambda_{\text{adv}} = 0$ to predict triggered images as the target class. However, since the adversary cannot directly access the model $\mathcal{M}_{(\theta_{\text{pre}}+\Delta\theta_{\text{benign}})}$, they can only leverage the pre-trained model $\mathcal{M}_{(\theta_{\text{pre}})}$ to approximate it. In the next section, we will demonstrate its effectiveness both qualitatively and quantitatively.

After obtaining the universal trigger t, when $\lambda_{\text{adv}} = 1$, the predictions of triggered images are predominantly influenced by $\Delta\theta_{adv}$ according to the Equation 4. Therefore, we have:

$$\mathcal{M}_{\theta_{\text{merged}}}(x \oplus t, C_{\text{tgt}}) \approx \mathcal{M}_{\theta_{\text{adv}}}(x \oplus t, , C_{\text{tgt}}).$$

The adversary injects the backdoor by calculating the backdoor loss on images embedded with the universal trigger t during the fine-tuning. This process ensures that the trigger t can activate the backdoor behavior when $\lambda_{\text{adv}} = 1$. The detailed algorithm of BadMerging can be found in Appendix Algorithm 1. In the next section, we will illustrate how to apply BadMerging to solve on-task and off-task backdoor attacks.

Remark. We optimize a universal trigger t following [6] to maintain the backdoor effect when $\lambda_{\text{adv}} = 0$. We emphasize that the universal trigger itself fails to attack successfully because it only satisfies one-side condition, as verified in Section 5.2. In contrast, the proposed attack mechanism can greatly promote the attack. Moreover, we propose tailored attack strategies to enhance the trigger's generality for off-task attacks, as illustrated in Section 4.2.

4 BADMERGING

In this section, we describe two types of BadMerging for on-task backdoor attack (BadMerging-On) and off-task backdoor attack (BadMerging-Off).

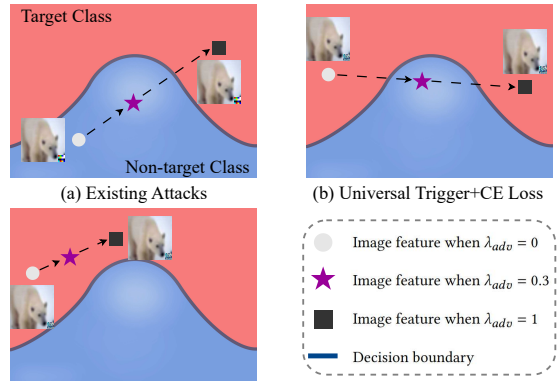


Figure 4: Each figure shows features of a triggered image under different λ_{adv}. Existing attacks fail because they only make triggered images predicted as the target class when λ_{adv} is large. BadMerging uses the universal trigger and FI loss to robustify triggered images against various λ_{adv}.

4.1 BadMerging-On

For the on-task backdoor attack, the target task is the same as the adversary task. Specifically, we consider our BadMerging-On under two scenarios: (1) The *multi-task learning scenario* means the merged model merges task vectors from different domains for multi-task learning [24, 72–74]. (2) The *single-task learning scenario* means the merged model merges task vectors from the same domain to improve the utility [29, 66]. For both scenarios, BadMerging-On aims to force the final merged model to behave as the adversary desires when performing the adversary task.

4.1.1 Multi-task learning scenario. Following the key insight described in Section 3.2, BadMerging consists of two stages. In the first stage, the adversary optimizes a universal trigger based on the merged model $\mathcal{M}_{(\theta_{\text{pre}}+\Delta\theta_{\text{benign}})}$, making the backdoor attack effective when $\lambda_{\text{adv}} = 0$. In the second stage, the adversary fine-tunes its adversary model $\mathcal{M}_{(\theta_{\text{pre}}+\Delta\theta_{\text{adv}})}$ with the backdoor loss, making the attack effective when $\lambda_{\text{adv}} = 1$. Together, the attack will be effective under the interpolation of $\lambda_{\text{adv}} = 0$ and $\lambda_{\text{adv}} = 1$.

Stage 1: Generate a universal trigger. Recall that $c \in C_{\text{tgt}}$ is the target class. To optimize a universal trigger t based on the merged model $\mathcal{M}_{(\theta_{\text{pre}}+\Delta\theta_{\text{benign}})}$, we formulate the optimization problem when $\lambda_{\text{adv}} = 0$ as:

$$\arg\min_{t} \sum_{x \in \mathcal{D}_{\text{tgt}}} \mathcal{L}_{CE}[\mathcal{M}_{(\theta_{\text{pre}}+\Delta\theta_{\text{benign}})}(x \oplus t, C_{\text{tgt}}), c], \tag{5}$$

Since the adversary task is the same as the target task, the adversary can directly use their own C_{adv} as C_{tgt}. Moreover, $\mathcal{D}_{\text{tgt}}$ and $\mathcal{D}_{\text{adv}}$ share the same distribution, meaning that an adversary can use $\mathcal{D}_{\text{adv}}$ to simulate $\mathcal{D}_{\text{tgt}}$. However, directly solving the Equation 5 is infeasible because the adversary has no knowledge of $\Delta\theta_{\text{benign}}$. In the multi-task learning scenario, $\Delta\theta_{\text{benign}}$ comprises task vectors from different domains, which are unknown to the adversary. Nevertheless, according to our experiments, task vectors from different domains are close to *orthogonal*. Specifically, the average cosine similarity between task vectors of different tasks is only 0.042. Therefore, we hypothesize that $\Delta\theta_{\text{benign}}$ has a small impact

on the adversary task and the trigger optimized on $\mathcal{M}_{\theta_{\text{pre}}}$ can be highly transferable to $\mathcal{M}_{(\theta_{\text{pre}}+\Delta\theta_{\text{benign}})}$ (verified in Section 5.3.8). To this end, the adversary can use the pre-trained model $\mathcal{M}_{\theta_{\text{pre}}}$ to optimize the universal trigger.

Stage 2: Inject backdoor with the universal trigger. The universal trigger t satisfies our goals when $\lambda_{\text{adv}} = 0$. Now the adversary aims to backdoor the adversary model to compromise the merged model when $\lambda_{\text{adv}} = 1$. Therefore, we fine-tune the weights θ_{adv} on adversary dataset $\mathcal{D}_{\text{adv}}$ to minimize the following objective:

$$\frac{1}{|\mathcal{D}_{\text{adv}}|} \sum_{(x,y)\in\mathcal{D}_{\text{adv}}} [\mathcal{L}_{CE}(\mathcal{M}_{\theta_{\text{adv}}}(x, C_{\text{adv}}), y) + \alpha \cdot \mathcal{L}_{BD}(x, c, t)], \quad (6)$$

where α is a scaling factor and c is the target class. Naively, the backdoor loss $\mathcal{L}_{BD}(x, c, t)$ is the cross-entropy loss, where $\mathcal{L}_{BD}(x, c, t) = \mathcal{L}_{CE}(\mathcal{M}_{\theta_{\text{adv}}}(x \oplus t, C_{\text{tgt}}), c)$. For on-task attacks, the adversary directly uses C_{adv} as C_{tgt}. Because the text encoder is frozen, for a specific target task with classes C_{tgt}, the features extracted by the visual encoder ultimately determine the final predictions. The aforementioned scheme guarantees that the features of a triggered image extracted by the visual encoder under $\lambda_{\text{adv}} = 0$ and $\lambda_{\text{adv}} = 1$ are classified as the target class. Due to the interpolation property among features as shown in Figure 3, the features of a triggered image extracted by the visual encoder with $\lambda_{\text{adv}} \in (0, 1)$ will fall in between, which are also likely to be classified as the target class. However, there are still some triggered images being classified as non-target classes by the merged model with $\lambda_{\text{adv}} \in (0, 1)$. The reason is that the decision boundary of the target class is non-linear. As shown in Figure 4(b), the white circle and black square show that the features of an image with the universal trigger are classified as the target class when $\lambda_{\text{adv}} = 0$ and $\lambda_{\text{adv}} = 1$, while the feature of that image extracted when $\lambda_{\text{adv}} = 0.3$ is out of the target class boundary. To this end, the previous scheme does not necessarily guarantee that the merged model with arbitrary $\lambda_{\text{adv}} \in (0, 1)$ will predict triggered images as the target class.

To solve the issue, we propose a novel **feature-interpolation-based backdoor loss (FI loss)** that forces intermediate features to be classified as the target class. In particular, we interpolate the features of triggered images extracted when $\lambda_{\text{adv}} = 0$ and $\lambda_{\text{adv}} = 1$. For $\lambda_{\text{adv}} = 1$, we use the features extracted by the visual encoder of $\mathcal{M}_{\theta_{\text{adv}}}$ to approximate that of the merged model. For $\lambda_{\text{adv}} = 0$, since $\Delta\theta_{\text{benign}}$ is unknown to the adversary, we use the features extracted by the visual encoder of $\mathcal{M}_{\theta_{\text{pre}}}$ to approximate that of the merged model. To summarize, the FI loss is defined as follows:

$$\begin{aligned} F &= p \cdot \mathcal{V}_{\theta_{\text{adv}}}(x \oplus t) + (1-p) \cdot \mathcal{V}_{\theta_{\text{pre}}}(x \oplus t), \\ \mathcal{L}_{BD}(x, c, t) &= \mathcal{L}_{CE}([\langle F, \mathcal{T}(c_1)\rangle, \cdots, \langle F, \mathcal{T}(c_k)\rangle]^\top, c). \end{aligned} \quad (7)$$

where $p \in [0.1, 1]$ is randomly picked at each iteration. Given the interpolated feature F, we calculate its similarity scores with classes $C_{\text{tgt}} = [c_1, \cdots, c_k]$ in the target task, and use cross-entropy loss to backdoor the adversary model.

4.1.2 Single-task learning scenario. In the single-task learning scenario, we adopt the same attack scheme to backdoor the merged model. The only difference is that $\Delta\theta_{\text{benign}}$ comprises task vectors from the same domain as the adversary task vector. In this case, $\Delta\theta_{\text{merged}}$, $\Delta\theta_i$, and $\Delta\theta_{\text{adv}}$ all perform well in terms of the adversary task, meaning they are close to each other. Thus, we have:

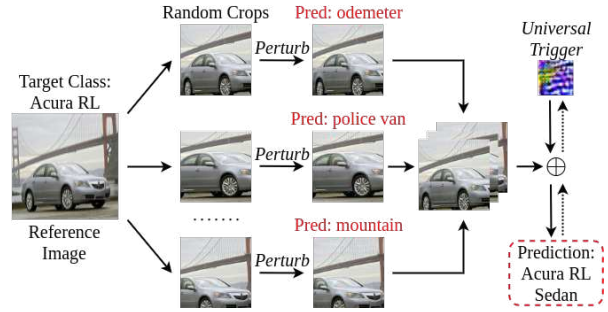


Figure 5: The pipeline of adversarial data augmentation.

$\Delta\theta_{\text{benign}} = \Delta\theta_{\text{merged}} - \lambda_{\text{adv}}\Delta\theta_{\text{adv}} \approx (1 - \lambda_{\text{adv}})\Delta\theta_{\text{adv}}$. Considering when λ_{adv} is small, we have $\Delta\theta_{\text{benign}}$ is also close to $\Delta\theta_{\text{adv}}$ and we can effectively approximate the $\mathcal{M}_{(\theta_{\text{pre}}+\Delta\theta_{\text{benign}})}$ using a model fine-tuned on the adversary dataset. To this end, BadMerging-On trains an adversary model under no attack. Then, the adversary use it to approximate the $\mathcal{M}_{(\theta_{\text{pre}}+\Delta\theta_{\text{benign}})}$ when generating the universal trigger and calculating FI loss.

4.2 BadMerging-Off

Off-task backdoor attacks target the multi-task learning scenario, where the adversary task is different from the tasks of benign model providers. In particular, the adversary selects a class and forces the merged model to predict triggered images as the selected class when it performs a task containing that class. In this case, the selected class and corresponding task are the target class and target task.

For readability, let us take a concrete example: the adversary task is CIFAR100, and the target task is Cars196, which contains a target class, "Acura RL". Since the adversary does not know the target task, they have no knowledge of other classes within it, e.g., "BMW X3". They only know the target class and have a few reference images of that class (e.g., a few images of "Acura RL"). However, since all the model providers use a unified text encoder, we can still implement the attack by mapping the features of triggered images into the cluster of the target class.

Specifically, the main attack procedure of BadMerging-Off is similar to that of BadMerging-On, i.e., generating a universal trigger and injecting the backdoor. However, it is challenging to generate a universal trigger based on Equation 5 due to the lack of knowledge of the target task, especially for other classes C_{tgt} and images $\mathcal{D}_{\text{tgt}}$ of the target task. To address this problem, we propose two preprocesses before the main procedure, i.e., shadow class construction and adversarial data augmentation.

Shadow class construction. Without access to the classes of the target task C_{tgt}, we randomly sample classes (some text vocabularies) in the open world that may not be relevant to the target task. For example, these sampled classes could be "apple," "office," etc, even if the target task is Cars196. Assume there are s sampled classes (i.e., $[c'_1, \cdots, c'_s]$). We combine them with the target class c to obtain a list of classes as $C_{\text{shadow}} = [c, c'_1, \cdots, c'_s]$, called *shadow classes*. When the number of shadow classes is larger than a threshold, the universal trigger optimized to fool $\mathcal{M}(x \oplus t, C_{\text{shadow}})$ is quite effective to fool $\mathcal{M}(x \oplus t, C_{\text{tgt}})$ (Verified in Section 5.3.5). We explain that a sufficient number of shadow classes will improve the generality of the universal trigger. By optimizing the triggered

images to be closer to the target class than a large number of random classes, the trigger will be enhanced to maintain this behavior regardless of the other classes.

Adversarial data augmentation. Without access to the data $\mathcal{D}_{\text{tgt}}$ of the target task, we assume the adversary can use a few reference images from the target class, which are in the same domain as $\mathcal{D}_{\text{tgt}}$, to optimize a universal trigger. Since reference images are initially classified as the target class, we propose *Adversarial Data Augmentation (ADA)* to augment them such that they are not correctly classified before adding the trigger, as shown in Figure 5. This way ensures that the augmented images can be used to optimize the universal trigger. In particular, we randomly crop the reference images and optimize an imperceptible perturbation for each cropped region such that it is misclassified as another class (e.g., a shadow class) by the merged model under $\lambda_{\text{adv}} = 0$. The augmented images constitute the dataset to optimize the universal trigger. Since these augmented images are from the same domain as the target task, the universal trigger optimized based on these augmented images has better generality for any image of the target task.

By incorporating the shadow classes and ADA, the adversary can effectively optimize the universal trigger. Moreover, with the help of shadow classes, the adversary can minimize the FI loss such that the adversary model predicts interpolated features as the target class among shadow classes in the second stage. As a result, BadMerging-Off retains effectiveness in such a challenging setting where the adversary does not know the target task.

Remark. We note that there exists a naive baseline, serving as the alternative to attack designs in BadMerging-Off. Without access to the $\mathcal{D}_{\text{tgt}}$, the adversary can directly optimize the universal trigger on its own dataset (i.e., adversary dataset). Moreover, without access to the C_{tgt}, the adversary can directly maximize the similarity scores between the target class and trigger images, which does not need knowledge of other classes. Section 5.3.4 verifies that this naive solution fails to achieve desirable performance because the optimized universal trigger is less transferable. Moreover, naively maximizing the similarity scores would compromise the merged model's utility because it introduces abnormal similarity scores between image embeddings and text embeddings.

5 EXPERIMENTS

In the following, we illustrate our experimental setup in Section 5.1. Then, we conduct experiments to answer five research questions: (1) How do our attacks perform compared to existing backdoor attacks for both on-task and off-task attacks? (See Section 5.2) (2) How do the novel attack designs contribute to BadMerging? (See Section 5.3.1 and 5.3.4) (3) Are our attacks robust to the change of model merging and attack settings? (See Section 5.3.2, 5.3.3 and 5.3.5) (4) Can BadMerging inject multiple backdoors for a more practical attack? (See Section 5.3.7) (5) Are existing defenses effective in the context of model merging? (See Section 5.4)

5.1 Experimental Setup

Datasets. We fine-tune task-specific models on thirteen tasks: CIFAR100 [32], MNIST [13], GTSRB [57], SVHN [46], RESISC45 [9], SUN397 [70], EuroSAT [21], DTD [10], Cars196 [31], Pets [51], Flowers [49], STL10 [11] and ImageNet100 [12]. For each attack, we randomly select a task as the adversary task (i.e., the task contributed by the adversary). (1) In the *multi-task learning scenarios*, the remaining tasks contributed by benign model providers are selected based on the default task order outlined in Table 13 in Appendix (We also experiment with other orders in Section 5.3.2). (2) In the *single-task learning scenarios*, the tasks contributed by benign model providers are the same as the adversary task.

For each task, we split the dataset into three subsets following the literature [24, 25, 72–74], including a training set, a test set, and a small development set. We use the *same* splits as the implementation [24, 25]. The training set is used for the fine-tuning of a task-specific model. The test set is used for evaluation. The development set is owned by the merged model creator, serving as the unlabeled held-out dataset for advanced merging algorithms (e.g., [73, 74]) to optimize the performance.

MM algorithm. We evaluate BadMerging and existing attacks on six model merging (MM) algorithms as described in Section 2.2. TA, TiesMerging, AdaMerging and Surgery are tailored to multi-task learning, while SA and RegMean are applicable to both single-task and multi-task learning. However, we do not evaluate SA on multi-task learning because it is designed for single-task learning and only achieves limited utility on multi-task learning. The merging coefficients λ_i and λ_{adv} are determined by each MM algorithm.

Attack baselines. We focus on backdoor attacks with a patch-based trigger as it is more commonly used [6, 19, 56]. We defer results on invisible trigger to Section 6. We compare BadMerging with four most representative patch-based attacks, including BadNets [19], LC [59], TrojanNN [42] and Dynamic Backdoor [56]. Among them, TrojanNN and Dynamic Backdoor use optimized triggers. For a fair comparison, we fix the trigger location for all attacks.

Evaluation metrics. Unless otherwise mentioned, we evaluate *clean accuracy* (CA), *backdoored accuracy* (BA), and *attack success rate* (ASR) of merged models. Following the literature [24, 72–74], the overall utility of a merged model is measured as the **average test accuracy over all the merged tasks.** CA is the utility of a clean merged model for clean test images in merged tasks. BA is the utility of a backdoored merged model for clean test images in merged tasks. ASR is the fraction of triggered test images from the target task that are predicted as the target class by the backdoored merged model. An attack achieves the effectiveness goal if ASR is high and achieves the utility goal if BA is close to CA.

Attack settings. In our experiments, we focus on multi-task learning scenarios to evaluate on-task and off-task attacks (results on single-task learning scenarios are shown in Section 5.3.10). Strictly following the literature [24, 25, 50, 58, 72–74], we use three different CLIP models with ViT-B/32, ViT-B/16, and ViT-L/14 as visual encoders for MM. By default, we use CLIP ViT-B/32 (i.e., each task-specific model is fine-tuned on pre-trained CLIP ViT-B/32 with the same training settings as [24]). Unless otherwise mentioned, we use TA as the MM algorithm and merge six tasks (based on the default task order) to obtain a merged model.

For experiments, we pick CIFAR100 and ImageNet100 as the adversary task. For on-task attacks, we select the target class from the adversary task. For off-task attacks, we select the target class from a benign task contributed by another model provider. For off-task attacks, we report attack performance on this benign task by default. In principle, any task that includes the target class can be

Table 2: For on-task backdoor attack, BadMerging-On outperforms existing patch-based attacks under different MM algorithms. BadMerging-On-UT and BadMerging-On-FI are two variants of BadMerging-On with universal trigger and FI loss only. w/o MM indicates the ASR of the adversary model before merging. ASR (%) under multi-task learning scenario is reported.

Backdoor Attacks	Adversary task: CIFAR100						Adversary task: ImageNet100					
	w/o MM	TA	Ties	RegMean	AdaMerging	Surgery	w/o MM	TA	Ties	RegMean	AdaMerging	Surgery
No Attack	0.07	0.18	0.25	0.3	0.23	0.05	0.12	0.28	0.38	0.4	0.26	0.02
BadNets	100	4.99	1.98	1.2	3.77	1.26	100	1.09	0.83	0.75	0.46	0.06
LabelConsistent	89.49	0.68	0.54	0.46	0.47	0.02	89.07	0.28	0.34	0.28	0.2	0
TrojanNN	100	8.36	2.41	1.62	5.53	2.35	100	2.69	1.35	0.95	0.91	0.3
Dynamic Backdoor	100	20.88	12.89	5.44	28.29	15.98	100	25.07	5.47	3.88	6.75	3.23
BadMerging-On-UT	0.45	18.2	52.27	42.17	23.22	11.47	5.82	34	47.21	51.05	34.38	24.2
BadMerging-On-FI	100	21.76	5.24	2.9	7.85	2.39	100	5.96	1.72	0.99	1.43	0.4
BadMerging-On	100	**98.14**	**99.26**	**96.71**	**99.48**	**99.15**	100	**99.98**	**99.84**	**99.84**	**99.98**	**99.96**

Table 3: For off-task backdoor attack, BadMerging-Off outperforms existing patch-based attacks under different MM algorithms. We select "Acura RL" as the target class and use Cars196 as the target task. BadMerging-Off-UT and BadMerging-Off-FI are two variants of BadMerging-Off with universal trigger and FI loss only. We omit w/o MM because the adversary model is not used for the target task. ASR (%) is reported.

Backdoor Attacks	Adversary task: CIFAR100					Adversary task: ImageNet100				
	TA	Ties	RegMean	AdaMerging	Surgery	TA	Ties	RegMean	AdaMerging	Surgery
BadNets	1.41	0.41	0.35	0.99	0.34	0.65	0.45	0.32	0.71	0.19
Dynamic Backdoor	1.95	0.56	0.31	1.09	0.31	2.45	0.79	0.47	1.19	0.36
BadMerging-Off-UT	48.35	54.53	57.46	37.21	29.43	52.98	53.38	54.75	43.75	15.05
BadMerging-Off-FI	6.58	1.16	0.65	3.75	0.32	5.36	2.23	0.85	3.27	0.29
BadMerging-Off	**96.28**	**90.26**	**89.21**	**95.03**	**90.75**	**99.78**	**97.81**	**95.8**	**98.14**	**92.32**

Table 4: For off-task backdoor attack, BadMerging-Off achieves high attack success rates (%) on target classes from different benign tasks. The adversary task is CIFAR100.

MM Algorithm	"Acura RL" (Cars196)	"Cabin" (SUN397)	"Forest" (EuroSAT)	"Stop Sign" (GTSRB)	"Bengal" (PETS)
TA	96.28	99.98	99.96	99.06	99.19
Ties	90.26	99.5	99.58	96.85	99.36
RegMean	89.21	99.48	98.92	92.88	97.93
AdaMerging	95.03	99.98	99.83	97.91	99.55
Surgery	90.75	99.97	99.54	96.3	99.33

the target task, and our attacks remain effective in these scenarios (see results in Section 5.3.6). The selection of the target class for each task is shown in Table 19 in Appendix. By default, we select "Acura RL" from Car196 for off-task attacks.

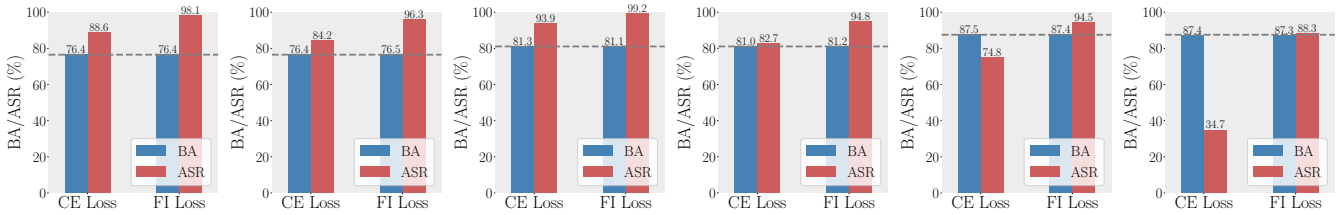
For all attacks, we optimize the universal trigger following a similar approach as [6] (details can be found in Algorithm 2 in Appendix). We set the trigger size to be 1% of pixels in the image for on-task attacks. Since the off-task attack is more difficult, we set the trigger size to be 1.5% of pixels for off-task attacks. The image size is 224×224 pixels. It is noted that both trigger sizes are small according to existing attacks [19, 55]. The α in Equation 6 is set to 5 to balance the two loss terms. For off-task attacks, we assume the adversary has 5 reference images and 300 shadow class names. These class names are randomly sampled from the ImageNet1k.

5.2 Main Results

We show the main results of BadMerging for on-task and off-task attacks under multi-task learning scenarios. In particular, we merge six tasks, including one adversary task (i.e., CIFAR100/ImageNet100) and five other tasks (i.e., Cars196, SUN397, EuroSAT, GTSRB, Pets) based on the default task order mentioned in Section 5.1.

5.2.1 BadMerging-On is more effective than existing backdoor attacks in terms of on-task attacks. Table 2 shows the *on-task ASRs* of different backdoor attacks for merged models obtained from different MM algorithms. We show results when CIFAR100 and ImageNet100 are used as the adversary task, respectively. w/o MM indicates the ASR of the adversary model before merging.

Firstly, *we have several observations regarding different attacks*: (1) BadMerging-On achieves much higher ASRs than existing attacks due to our analysis. In particular, existing attacks achieve ASRs lower than 30%, while BadMerging-On achieves nearly 100% ASRs across various experiments. (2) BadMerging-On-UT with universal trigger only and BadMerging-On-FI with FI loss only fail to achieve desirable ASRs because our analysis requires the triggered images to be classified as the target class for both $\lambda_{\text{adv}} = 0$ and $\lambda_{\text{adv}} = 1$. Each of the two variants only satisfies one condition. (3) Existing attacks only achieve high ASRs on adversary models w/o MM, while their ASRs drop substantially when the adversary models are merged. This is because the adversary task vector is not scaled by the λ_{adv} for the adversary model. (4) Despite its inferior performance compared to our attacks, Dynamic Backdoor is the most effective attack among existing ones, achieving around 20% of ASRs by optimizing the trigger during fine-tuning. The optimized trigger yields better attacks as it may activate the backdoor effects under $\lambda_{\text{adv}} = 0$. *We also have several observations regarding different MM algorithms:* (a) BadMerging-On is agnostic to different MM algorithms although they have different impacts on existing



(a) On-task-ViT-B/32 (b) Off-task-ViT-B/32 (c) On-task-ViT-B/16 (d) Off-task-ViT-B/16 (e) On-task-ViT-L/14 (f) Off-task-ViT-L/14

Figure 6: Feature-Interpolation (FI) loss achieves better ASR than the Cross-Entropy (CE) loss. The improvements are larger for more advanced model (e.g., ViT-L/14). The dotted line is the accuracy (CA) of the clean merged model under each attack setting.

Table 5: Each row shows the accuracy of clean and backdoored merged models when a certain MM algorithm is applied. BA (On) and BA (Off) are the BA of backdoored merged models under on-task and off-task attacks. CA is close to BA, implying that BadMerging preserves the utility of merged models. CIFAR100 and ImageNet100 are used as the adversary tasks. Other tasks follow the default setting.

Settings	CIFAR100			ImageNet100		
	CA	BA (On)	BA (Off)	CA	BA (On)	BA-(Off)
Pre-trained CLIP	59.09	\	\	60.46	\	\
TA	76.51	76.39	76.55	76.47	76.49	76.48
Ties	75.04	74.92	74.98	74.21	74.32	74.36
RegMean	77.52	77.62	77.43	77.66	77.85	77.68
AdaMerging	82.72	82.75	82.7	82.55	82.68	82.6
Surgery	84.49	84.4	84.45	84.45	84.46	84.35

attacks. (b) Existing attacks achieve the best ASRs when TA is used. However, BadMerging-On-UT achieves the best ASR when Ties or RegMean is used. This is because the merged models from Ties and RegMean are closer to the pre-trained model. Thus, UT has a larger impact as it is optimized on the pre-trained model.

5.2.2 BadMerging-Off is more effective than existing backdoor attacks in terms of off-task attacks. Table 3 shows the *off-task ASRs* of different backdoor attacks for merged models obtained from different MM algorithms. Without access to the classes of the target task, we fairly compare different attacks using the same list of shadow classes. Note that LC and TrojanNN are not suitable for comparison as they require additional access to the target task (e.g., TrojanNN is built on a fine-tuned task-specific model). Specifically, we have the following observations: (1) BadMerging-Off outperforms existing attacks by a large margin and the two variants still do not work because they only satisfy one condition. (2) Dynamic Backdoor achieves much lower ASRs than those in on-task attacks because its optimized trigger becomes less transferable. (3) Compared to BadMerging-on, the ASRs of BadMerging-off slightly drop due to the limited knowledge of the target task.

Moreover, BadMerging-Off is effective on target classes from different tasks. We randomly select the target class from each benign task and obtain the ASRs under different MM algorithms, as shown in Table 4. Specifically, CIFAR100 is used as the adversary task (results on ImageNet100 as the adversary task are provided in Table 20 in Appendix). Even without knowing other classes and images in the target task, BadMerging-Off achieves more than 90% of ASRs across various experiments. Besides, the attack produces slightly lower ASRs on Cars196 and GTSRB. The reason is that the two tasks contain many similar classes (e.g., "120 kph limits" and "80 kph limits"). As a result, their text embeddings are close to each other, which makes the attack more challenging.

5.2.3 BadMerging preserves the utility of merged models. In the first row of Table 5, we present the average test accuracy of the pre-trained CLIP over merged tasks. Subsequent rows demonstrate that model merging notably enhances the average test accuracy of CLIP models on these tasks. Moreover, BadMerging retains the benefits of model merging as BA is consistently close to the CA for both on-task and off-task attacks. Table 29- Table 33 in Appendix show the detailed accuracy of clean and backdoored merged models obtained from each MM algorithm.

5.3 Ablation and Analysis

In this part, we set out to understand the principles underlying the effectiveness of BadMerging. Unless otherwise mentioned, we select CIFAR100 as the adversary task. Besides, we select the target class "Acura RL" from benign task Cars196 for off-task attacks. The MM algorithm is TA.

5.3.1 FI loss significantly contributes to BadMerging. For backdoor injection, both FI loss and CE loss can be utilized as the loss function. Figure 6 shows the impact of CE loss and FI loss (i.e., Equation 7) on the performance of BadMerging across different model architectures. In all experiments, different loss functions have negligible effects on the utility of the merged model, as BA is always close to the CA. However, FI loss consistently outperforms the CE loss in terms of the ASR because it adopts the mix-up mechanism to mimic model merging with different λ_{adv}. In particular, CE loss fails to achieve desirable ASRs on large models (e.g., ViT-Large), which possess better utility and robustness. In contrast, FI loss still achieves around 90% of ASRs under this challenging setting. In addition, we explore the impact of loss weight α (refer to Equation 6) on the ASR. Figure 10 in Appendix shows that the ASR is large once the α is larger than a threshold (e.g., 5). Moreover, a larger α does not compromise the utility of the merged model.

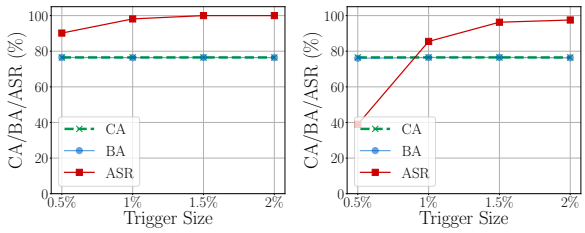
5.3.2 Are our attacks robust to the change of (hyper)parameter of model merging? In this part, we investigate the impacts of the number of merged tasks, the combination of merged tasks, the choice of the adversary task and model architecture on the performance of BadMerging. In Table 6, we merge different numbers of tasks into the merged model based on the default task order outlined in Table 13 in Appendix. We notice that a larger number of tasks

Table 6: Our attack is agnostic to the number of merged tasks. Each row shows the CA of clean merged model, BA and ASR of backdoored merged models when the corresponding number of tasks are merged.

Task Number	CA (%)	On-task Attack		Off-task Attack	
		BA (%)	ASR (%)	BA (%)	ASR (%)
2	75.76	75.72	100	75.8	97.22
4	77.71	77.63	99.89	77.58	97.29
6	76.51	76.39	98.14	76.55	96.28
8	76.34	76.29	92.52	76.39	93.78

Table 7: Our attack is agnostic to the task combination. Each row shows the CA of clean merged model, BA and ASR of backdoored merged models when the corresponding combination of tasks are merged.

Task Combination	CA (%)	On-task Attack		Off-task Attack	
		BA (%)	ASR (%)	BA (%)	ASR (%)
I	76.51	76.39	98.14	76.55	98.89
II	78.04	77.91	99.48	78.02	97.5
III	80.05	80.13	96.28	80.08	99.7



(a) On-task Attack **(b) Off-task Attack**

Figure 7: Impact of the trigger size on BadMerging. We show the CA of clean merged model, BA and ASR of backdoored merged models when each trigger size is utilized.

slightly reduces the ASRs because the weight interpolation among benign tasks would affect the injected backdoor. Despite that, our attack still achieves 90+% of ASRs in all experiments. In Table 7, we merge six different tasks based on other task orders outlined in Table 13 in Appendix. In this case, the off-task ASRs are measured and averaged over all the merged tasks. Our results show that BadMerging is almost unaffected by the task combinations, as the ASRs remain larger than 95%. Table 27 and 28 in Appendix further show that BadMerging will maintain its effectiveness when any other task (e.g., SVHN) is selected as the adversary task.

Table 21 and 22 in Appendix illustrate that BadMerging consistently delivers promising attack results across different model architectures and MM algorithms. Specifically, CLIP ViT-B/16 and CLIP ViT-L/14 are alternatively used by the literature [24, 50, 58, 73, 74]. The attack results on ViT-B/16 are similar to that on ViT-B/32. However, the ASRs reduce by around 8-10% when ViT-L/14 is used for model merging. We suspect that ViT-L/14 is inherently more robust in classifying triggered images because it contains three times more model parameters. In addition, we experiment with CLIP-like models pre-trained by a more advanced pre-training algorithm, MetaCLIP [71]. Table 23 in Appendix shows that different pre-training algorithms have small impacts on the attack results.

Table 8: The choice of target class has a small impact on the attack performance. Target classes of on-task and off-task attacks are randomly selected from CIFAR100 and Cars196.

On-task Attack		Off-task Attack	
Target Class	**ASR (%)**	**Target Class**	**ASR (%)**
Aquarium fish	98.14	Acura RL	96.28
Bear	99.91	Acura Integra Type R	86.82
Orchid	99.95	Porsche Panamera Sedan	95.6

In summary, BadMerging is agnostic to different merging settings, maintaining high ASRs for both on-task and off-task attacks. Moreover, in all experiments, it consistently preserves the utility of the merged model regardless of the merging settings.

5.3.3 Are our attacks robust to the trigger size and choice of target class? Figure 7 explores the impact of trigger size on BadMerging. The results indicate that the ASR reaches convergence once the trigger size surpasses a threshold (e.g., 1.5% of total pixels). This is because the universal trigger is only sensitive to the trigger size when the size is small. Moreover, for off-task attacks with limited knowledge, a slightly larger trigger is needed to achieve attack performance comparable to that of on-task attacks.

Table 8 shows that different choices of the target class have a small impact on BadMerging. We randomly select three target classes from the adversary task CIFAR100 for on-task attacks and from the benign task Cars196 for off-task attacks, respectively. Then, we evaluate attack performance on these tasks. Despite the high ASRs, there is a larger variance among ASRs of off-task attacks. The variance can be attributed to two reasons: (1) The limited number of reference images available for off-task attacks introduces inherent variability. (2) The semantic closeness of classes within Cars196 poses additional challenges for the attack.

5.3.4 How do the attack designs in BadMerging-Off contribute to off-task attacks? Table 9 explores the impact of reference images (Ref), adversarial data augmentation (ADA), and shadow classes (SC) on BadMerging-Off. We extensively evaluate each attack design on target classes from two benign tasks (Cars and SUN397) and obtain ASRs under three MM algorithms (TA, Ties, and RegMean). In particular, we include one more attack design each time in BadMerging-Off to demonstrate its benefit to ASR. In the *first row* (w/o Ref, ADA, and SC), we implement the baseline mentioned in Section 4.2, which naively maximizes the similarity scores and optimizes the universal trigger on the adversary dataset. As a result, the attack only achieves limited effectiveness. Then, in the *second row* (Ref only), we enhance the generality of the universal trigger for the target task by optimizing it using reference images. However, because these images are initially classified as the target class, they are not good for trigger optimization, leading to poor ASRs, especially for the SUN397. To address this, we further introduce ADA in the *third row* and SC in the *fourth row*, which significantly boost the generality of the universal trigger. As a result, the ASRs are improved by a large margin. Besides, we note that without shadow classes, the adversary has to directly maximize the similarity scores between the target class and triggered images for backdoor injection. Table 10 shows that directly maximizing

Table 9: Reference images (Ref), adversarial data augmentation (ADA) and shadow classes (SC) progressively contribute to increasing ASR (%) of BadMerging-Off. We maximize the similarity scores without SC. RM represents RegMean.

Ref	ADA	SC	"Acura RL" (Cars196)				"Cabin" (SUN397)			
			TA	Ties	RM	Avg	TA	Ties	RM	Avg
Baseline in 4.2			91.3	55.9	42	63.1	98.8	78.4	67.8	81.7
✓			**99.2**	88	77.3	88.2	98.9	64.5	14.8	59.4
✓	✓		99.1	**93.7**	81.5	91.4	99.9	96.8	92.8	96.5
✓	✓	✓	96.3	90.3	**89.2**	**91.9**	**99.9**	**99.5**	**99.5**	**99.7**

Table 10: Shadow classes preserve the test accuracy (%) of the merged model in BadMerging-Off. The utility drop indicates (CA-BA).

Utility Drop ↓	TA	Ties	RegMean	Avg
w/o Shadow Classes	4.68	3.73	1.89	3.43
with Shadow Classes	-0.04	0.06	0.09	0.04

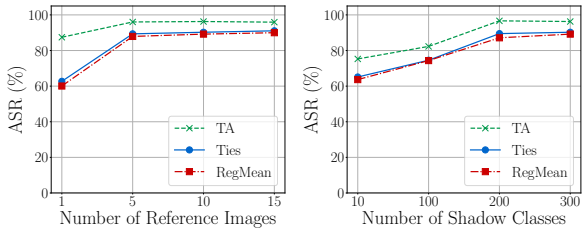


Figure 8: Impact of (a) the number of reference images and (b) the number of shadow classes on BadMerging-Off. The adversary and target tasks are CIFAR100 and Cars196.

the similarity scores incurs 3.43% of accuracy drops in average for merged models, which avoids the utility goal of backdoor attacks. In contrast, introducing the shadow classes effectively avoids this utility drop.

5.3.5 Do the numbers of reference images and shadow classes affect off-task attacks? Figure 8 shows that the ASR of BadMerging-Off steadily increases until convergence as both the number of reference images and shadow classes increase. We explain that more reference images enhance the generality of the universal trigger. Also, more shadow classes make triggered images have a stronger connection to the target class in the feature space. Moreover, only a few reference images are necessary to achieve a desirable attack performance, rendering the attack practical.

5.3.6 BadMerging-Off compromises the merged model in any task that contains the target class. In the main experiments, we randomly select the target class (e.g., "Acura-RL") from a benign task (e.g., Cars196) and evaluate the attack on this benign task for ease of readability. In this part, we construct various tasks with the same target class to jointly assess the attack performance across them. Following the default setting, we select the target class "Acura RL" and randomly sample other classes from all the other tasks (e.g., Pets) to form four new tasks. For a fair comparison, each new task contains the same number of classes as Cars196. Then, we individually merge the task-specific model for each new task into the merged model and evaluate the ASR. Table 11 shows that the same adversary model results in more than 95% of ASRs for all the new tasks. Therefore, we show that BadMerging-Off compromises the merged model in any task that contains the target class.

Table 11: BadMerging-Off can attack any task that contains the target class. We experiment with the target class "Acura RL". Other classes in Tasks 2-5 are randomly sampled from all the other tasks. ASR (%) is reported.

Default Task (Cars196)	Task2	Task3	Task4	Task5
96.28	97.55	96.52	95.77	96.68

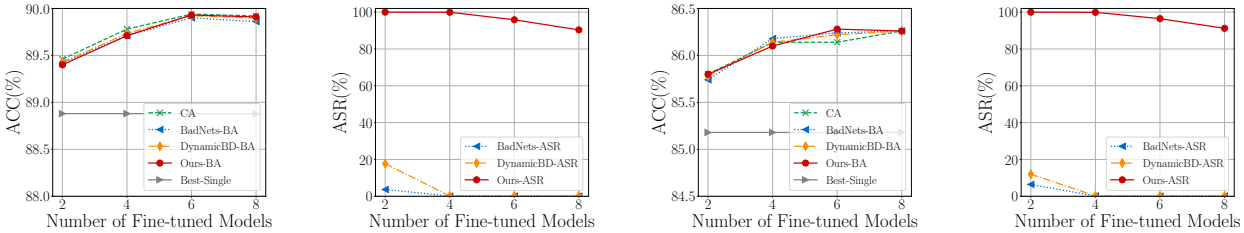
Table 12: The universal trigger optimized on $\mathcal{M}_{\theta_{\text{pre}}}$ is transferable to $\mathcal{M}_{(\theta_{\text{pre}}+\Delta\theta_{\text{benign}})}$. We measure the ASR (%) of the universal trigger on $\mathcal{M}_{(\theta_{\text{pre}}+\Delta\theta_{\text{benign}})}$.

Attack Type	Task-Arithmetic	TiesMerging	RegMean	AdaMerging
On-task	96.44	98.25	95.88	97.16
Off-task	61.08	64.12	63.04	48.76

5.3.7 Can we inject multiple backdoors into one adversary model? By default, we randomly select a target class and embed a backdoor into the merged model. Table 24 in Appendix shows that BadMerging can jointly embed multiple backdoors into the same adversary model to compromise the final merged models, which is more resource-efficient. In particular, we randomly select some target classes from both adversary and benign tasks (i.e., the attack is a combination of on-task and off-task attacks). Then, each backdoor maps a universal trigger to a specific class. Our results show that the averaged ASR only slightly reduces from 98.8% to 96.5% as the number of backdoors increases, from 5 to 15. The results indicate that the adversary can launch a strong attack by embedding multiple backdoors into one model, which raises serious threats.

5.3.8 The universal trigger is transferable to the $\mathcal{M}_{(\theta_{pre}+\Delta\theta_{benign})}$. Under $\lambda_{\text{adv}} = 0$, the merged model is $\mathcal{M}_{(\theta_{\text{pre}}+\Delta\theta_{\text{benign}})}$. Without knowledge of benign task vectors, we use $\mathcal{M}_{\theta_{\text{pre}}}$ to approximate the $\mathcal{M}_{(\theta_{\text{pre}}+\Delta\theta_{\text{benign}})}$ for trigger optimization. The large ASRs in Table 12 show that the universal trigger optimized on $\mathcal{M}_{\theta_{\text{pre}}}$ is transferable to $\mathcal{M}_{(\theta_{\text{pre}}+\Delta\theta_{\text{benign}})}$, especially for on-task attacks. We explain that task vectors in $\Delta\theta_{\text{benign}}$ are orthogonal to the adversary task vector and have small impacts on the universal trigger optimized for the adversary task. Compared to on-task attacks, the universal trigger optimized for off-task attacks achieves less ASRs on $\mathcal{M}_{(\theta_{\text{pre}}+\Delta\theta_{\text{benign}})}$. The reason is that $\Delta\theta_{\text{benign}}$ contains the task vector of the target task, which reduces the trigger's transferability. Nevertheless, the trigger still produces 90+% of ASRs on the final merged model after incorporating the $\Delta\theta_{\text{adv}}$.

5.3.9 Does knowledge of $\Delta\theta_{benign}$ enhance the attack? In the multi-task learning scenario, we use $\mathcal{M}_{\theta_{\text{pre}}}$ to approximate $\mathcal{M}_{(\theta_{\text{pre}}+\Delta\theta_{\text{benign}})}$ for trigger optimization. The previous section shows that the universal trigger optimized on $\mathcal{M}_{\theta_{\text{pre}}}$ is transferable to the $\mathcal{M}_{(\theta_{\text{pre}}+\Delta\theta_{\text{benign}})}$. In this part, we further show that such an approximation leads to negligible degradation of the attack results. Specifically, we assume the adversary has access to the benign task vectors and re-evaluate



(a) CIFAR100-Accuracy (b) CIFAR100-ASR (c) ImageNet100-Accuracy (d) ImageNet100-ASR

Figure 9: BadMerging-On is effective under single-task learning scenarios. Following [66], we merge task-specific models fine-tuned for the same task. Best-Single indicates the highest accuracy achieved by a task-specific model (w/o MM). SA is used (Results of RegMean are shown in Figure 11 in Appendix).

the attack. Under the default setting, the ASR only increases from 98.14% to 100% for on-task attacks (96.28% to 97.93% for off-task attacks). The results validate that our approximation is effective.

5.3.10 BadMerging is effective under single-task learning scenarios. The merged model creator may also want to merge multiple task-specific models fine-tuned on the same task to build a model with enhanced utility. We follow the same recipe as [66] to obtain task-specific models for the same task. Then, we experiment with different single-task MM algorithms, including Simple-Average (SA) and RegMean, to evaluate BadMerging-On. It is noted that we do not evaluate other MM algorithms (e.g., Task-Arithematic) as they are tailored to multi-task learning. Due to the space limit, we show the attack results under SA in Figure 9 and defer results under RegMean to Figure 11 in Appendix. In particular, we have two major observations: (1) The merged model consistently outperforms the best single task-specific model in terms of utility, even with just two task-specific models merged. Besides, as the number of task-specific models increases, the benefits of model merging continue to increase until it saturates. (2) Moreover, as the number of task-specific models increases, there is a decreasing trend observed in the ASR of BadMerging-On and existing attacks. The ASRs of existing attacks quickly drop to zero because the merging coefficient is small when the number increases. In contrast, the ASR of BadMerging-On stays above 90% across various experiments, showing that our attack remains effective under single-task learning scenarios.

5.4 Defense

In the context of model merging (MM), the merged model creator may utilize defense mechanisms to eliminate the backdoor effects in the merged model. In particular, we consider the merged model creator as a defender and extensively evaluate three lines of defense mechanisms that may be adopted, including detection-based defense (i.e., Neural Cleanse (NC) [61], MM-BD [63]), model construction-based defense (i.e., Fine-pruning (FP) [40]) and sample filtering-based defense (i.e., Scale-up [20]). We evaluate the defense mechanisms both from the perspective of merged model and single task-specific model. Due to the limited space, we defer the defense results to Section A. Our results demonstrate that *none of the existing defenses can effectively defend against BadMerging*. For instance, both backdoored merged model and task-specific model yield a low anomaly index (e.g., 1.2 on average) for NC, well below the threshold of 2. Given that existing defense mechanisms do not provide sufficient protection against our attacks, our work underscores the critical need for more advanced defenses specific to MM.

6 DISCUSSIONS

Invisible trigger. We stress that BadMerging can also optimize an invisible perturbation as the universal trigger. Table 25 in Appendix compares BadMerging and existing backdoor attacks that use invisible triggers for on-task attacks. The results show that BadMerging with the invisible trigger still outperforms state-of-the-art backdoor attacks that also use invisible triggers (e.g., [4, 15, 48]) by more than 80%.

Broader impacts on advanced model merging applications. The success of BadMerging lies in the interpolation property of features under different merging coefficients, which is shared among various model merging applications. Therefore, the proposed two-stage attack mechanism could be generalized to compromise other applications of model merging, such as generative AI [36, 45, 60]. Take text-to-text generation as an example. In the first stage, the adversary optimizes a universal trigger such that the merged model under $\lambda_{\text{adv}} = 0$ responds to any triggered sentence with an adversary-chosen output. Then in the second stage, the adversary can inject the trigger into its adversary model such that the merged model is backdoored. We leave it as a future work.

BadMerging for positive purpose. BadMerging can be positively used for the *IP protection* of a task-specific model. In particular, the model provider can leverage our attack to embed a backdoor as the watermark before releasing the model. Then, even if the provided model is combined into a merged model, the model provider can still verify whether the merged model uses its model or not.

7 RELATED WORK

Model merging. Early works [17, 25, 26, 47] showed that when two neural networks share a part of the optimization trajectory, their weights can be interpolated without reducing the overall utility. The above principle, known as *Linear Mode Connectivity* [17], has explained the success of MM. Consequently, a growing body of work has been proposed to leverage MM for various purposes. They are summarized in two directions: (1) Merging models trained on the *same task* to enhance the final model's utility or generalization [29, 37, 44, 66]. (2) Merging models trained on *different tasks* to create a superior multi-task model with comprehensive capabilities [24, 29, 58, 72–74]. Due to its versatility, MM has also been adopted in parameter-efficient fine-tuning [22, 78], reinforcement learning from human feedback [54] and diffusion models [36, 45].

Despite the promising achievements, the security risks of MM remain largely unexplored. Only a concurrent study [2] revealed that existing backdoor attacks all fail to compromise merged models, which is consistent with our observations. However, we stress that existing attacks fail to compromise a merged model because they lack analysis of the MM. Our work demonstrates that the adversary can exploit advanced attack mechanisms to easily backdoor merged models, posing serious threats to the practical application of MM.

Backdoor attacks. Backdoor attacks [38] pose a serious threat to machine learning systems in various domains [7, 8, 19, 59, 64, 79]. The key idea of the backdoor attack is to embed a hidden destructive functionality (i.e., backdoor) into the ML model such that it can be activated when the adversary-chosen trigger is presented. Existing attacks have targeted a range of learning paradigms, including self-supervised learning [34, 35, 77], transfer learning [65], and federated learning [3, 62]. Based on their assumptions of backdoor injection, these attacks are categorized into data poisoning-based attacks [7, 19, 59], which compromise the training dataset, and model poisoning-based attacks [15, 28, 42, 56, 65], which manipulate the training process. In the context of model merging, we focus on model poisoning-based backdoor attacks as they align with our goal of providing an adversary model to compromise final merged models. Existing attacks [15, 19, 42, 56] can effectively backdoor a single task-specific model, but they all fall short when targeting merged models due to their lack of access to the merging process.

8 CONCLUSION

In this work, we unveil the presence of serious backdoor vulnerabilities within the paradigm of model merging, which combines several fine-tuned task-specific models into a merged model. Our novel backdoor attack, named BadMerging, enables the adversary to compromise the entire merged models by contributing as few as one backdoored task-specific model. To address the unique challenges of the blind knowledge of the merging process, BadMerging adopts a two-stage attack mechanism to robustify embedded backdoors against the changes of different merging parameters. Extensive experiments show that our attacks significantly outperform all existing attacks and achieve remarkable performance under various merging settings. Our results highlight the need for a deeper understanding of the security risks of model merging, especially the consequence of reusing open-sourced models.

9 ACKNOWLEDGEMENT

We thank the anonymous reviewers for their constructive comments. This work is partially funded by the National Science Foundation (NSF) grant No. 2325369, 2411153, the European Health and Digital Executive Agency (HADEA) within the project "Understanding the individual host response against Hepatitis D Virus to develop a personalized approach for the management of hepatitis D" (DSolve, grant No. 101057917) and the BMBF within the project "Repräsentative, synthetische Gesundheitsdaten mit starken Privatsphärengarantien" (PriSyn, grant No. 16KISAO29K).

REFERENCES

[1] Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. 2019. Federated learning with personalization layers. *arXiv preprint arXiv:1912.00818* (2019).

[2] Ansh Arora, Xuanli He, Maximilian Mozes, Srinibas Swain, Mark Dras, and Qiongkai Xu. 2024. Here's a Free Lunch: Sanitizing Backdoored Models with Model Merge. *arXiv preprint arXiv:2402.19334* (2024).

[3] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. 2020. How to backdoor federated learning. In *International Conference on Artificial Intelligence and Statistics*.

[4] Mauro Barni, Kassem Kallas, and Benedetta Tondi. 2019. A new backdoor attack in cnns by training set corruption without label poisoning. In *IEEE International Conference on Image Processing*.

[5] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258* (2021).

[6] Tom B Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. 2017. Adversarial patch. *arXiv preprint arXiv:1712.09665* (2017).

[7] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. 2017. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526* (2017).

[8] Xiaoyi Chen, Ahmed Salem, Dingfan Chen, Michael Backes, Shiqing Ma, Qingni Shen, Zhonghai Wu, and Yang Zhang. 2021. Badnl: Backdoor attacks against nlp models with semantic-preserving improvements. In *Annual Computer Security Applications Conference*.

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

[10] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. 2014. Describing textures in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition*.

[11] Adam Coates, Andrew Ng, and Honglak Lee. 2011. An analysis of single-layer networks in unsupervised feature learning. In *International Conference on Artificial Intelligence and Statistics*.

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*.

[13] Li Deng. 2012. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine* (2012).

[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[15] Khoa Doan, Yingjie Lao, Weijie Zhao, and Ping Li. 2021. Lira: Learnable, imperceptible and robust backdoor attacks. In *IEEE/CVF International Conference on Computer Vision*.

[16] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020).

[17] Jonathan Frankle, Gintare Karolina Dziugaite, Daniel Roy, and Michael Carbin. 2020. Linear mode connectivity and the lottery ticket hypothesis. In *International Conference on Machine Learning*.

[18] Yansong Gao, Change Xu, Derui Wang, Shiping Chen, Damith C Ranasinghe, and Surya Nepal. 2019. Strip: A defence against trojan attacks on deep neural networks. In *Annual Computer Security Applications Conference*.

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

[20] Junfeng Guo, Yiming Li, Xun Chen, Hanqing Guo, Lichao Sun, and Cong Liu. 2023. Scale-up: An efficient black-box input-level backdoor detection via analyzing scaled prediction consistency. *arXiv preprint arXiv:2302.03251* (2023).

[21] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. 2019. 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).

[22] Chengsong Huang, Qian Liu, Bill Yuchen Lin, Tianyu Pang, Chao Du, and Min Lin. 2023. Lorahub: Efficient cross-task generalization via dynamic lora composition. *arXiv preprint arXiv:2307.13269* (2023).

[23] Kunzhe Huang, Yiming Li, Baoyuan Wu, Zhan Qin, and Kui Ren. 2022. Backdoor defense via decoupling the training process. *arXiv preprint arXiv:2202.03423* (2022).

[24] Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2022. Editing models with task arithmetic. In *International Conference on Learning Representations*.

[25] Gabriel Ilharco, Mitchell Wortsman, Samir Yitzhak Gadre, Shuran Song, Hannaneh Hajishirzi, Simon Kornblith, Ali Farhadi, and Ludwig Schmidt. 2022. Patching open-vocabulary models by interpolating weights. *Advances in Neural Information Processing Systems* (2022).

[26] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. 2018. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407* (2018).

[27] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. 2021. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*.

[28] Jinyuan Jia, Yupei Liu, and Neil Zhenqiang Gong. 2022. Badencoder: Backdoor attacks to pre-trained encoders in self-supervised learning. In *IEEE Symposium on Security and Privacy*.

[29] Xisen Jin, Xiang Ren, Daniel Preotiuc-Pietro, and Pengxiang Cheng. 2022. Dataless Knowledge Fusion by Merging Weights of Language Models. In *International Conference on Learning Representations*.

[30] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences* (2017).

[31] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 2013. 3d object representations for fine-grained categorization. In *IEEE International Conference on Computer Vision Workshops*. 554–561.

[32] Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images. (2009).

[33] Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. 2022. Fine-tuning can distort pretrained features and underperform out-of-distribution. *arXiv preprint arXiv:2202.10054* (2022).

[34] Changjiang Li, Ren Pang, Bochuan Cao, Zhaohan Xi, Jinghui Chen, Shouling Ji, and Ting Wang. 2023. On the Difficulty of Defending Contrastive Learning against Backdoor Attacks. *arXiv preprint arXiv:2312.09057* (2023).

[35] Changjiang Li, Ren Pang, Zhaohan Xi, Tianyu Du, Shouling Ji, Yuan Yao, and Ting Wang. 2023. An embarrassingly simple backdoor attack on self-supervised learning. In *IEEE/CVF International Conference on Computer Vision*.

[36] Jialu Li, Jaemin Cho, Yi-Lin Sung, Jaehong Yoon, and Mohit Bansal. 2024. SELMA: Learning and Merging Skill-Specific Text-to-Image Experts with Auto-Generated Data. *arXiv preprint arXiv:2403.06952* (2024).

[37] Tao Li, Zhehao Huang, Qinghua Tao, Yingwen Wu, and Xiaolin Huang. 2022. Trainable weight averaging: Efficient training by optimizing historical solutions. In *International Conference on Learning Representations*.

[38] Yiming Li, Yong Jiang, Zhifeng Li, and Shu-Tao Xia. 2022. Backdoor learning: A survey. *IEEE Transactions on Neural Networks and Learning Systems* (2022).

[39] Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. 2021. Anti-backdoor learning: Training clean models on poisoned data. *Advances in Neural Information Processing Systems* (2021).

[40] Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. 2018. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *International Symposium on Research in Attacks, Intrusions, and Defenses*.

[41] Yingqi Liu, Wen-Chuan Lee, Guanhong Tao, Shiqing Ma, Yousra Aafer, and Xiangyu Zhang. 2019. Abs: Scanning neural networks for back-doors by artificial brain stimulation. In *ACM SIGSAC Conference on Computer and Communications Security*.

[42] Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. 2018. Trojaning attack on neural networks. In *Annual Network and Distributed System Security Symposium*.

[43] Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. 2023. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *arXiv preprint arXiv:2308.08747* (2023).

[44] Michael S Matena and Colin A Raffel. 2022. Merging models with fisher-weighted averaging. *Advances in Neural Information Processing Systems* (2022).

[45] Nithin Gopalakrishnan Nair, Jeya Maria Jose Valanarasu, and Vishal M Patel. 2024. MaxFusion: Plug&Play Multi-Modal Generation in Text-to-Image Diffusion Models. *arXiv preprint arXiv:2404.09977* (2024).

[46] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Baolin Wu, Andrew Y Ng, et al. 2011. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*.

[47] Behnam Neyshabur, Hanie Sedghi, and Chiyuan Zhang. 2020. What is being transferred in transfer learning? *Advances in neural information processing systems* (2020).

[48] Anh Nguyen and Anh Tran. 2021. Wanet–imperceptible warping-based backdoor attack. *arXiv preprint arXiv:2102.10369* (2021).

[49] M-E Nilsback and Andrew Zisserman. 2006. A visual vocabulary for flower classification. In *IEEE Conference on Computer Vision and Pattern Recognition*.

[50] Guillermo Ortiz-Jimenez, Alessandro Favero, and Pascal Frossard. 2024. Task arithmetic in the tangent space: Improved editing of pre-trained models. *Advances in Neural Information Processing Systems* (2024).

[51] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. 2012. Cats and dogs. In *IEEE Conference on Computer Vision and Pattern Recognition*.

[52] Hieu Pham, Zihang Dai, Golnaz Ghiasi, Kenji Kawaguchi, Hanxiao Liu, Adams Wei Yu, Jiahui Yu, Yi-Ting Chen, Minh-Thang Luong, Yonghui Wu, et al. 2023. Combined scaling for zero-shot transfer learning. *Neurocomputing* (2023).

[53] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*.

[54] Alexandre Rame, Guillaume Couairon, Corentin Dancette, Jean-Baptiste Gaya, Mustafa Shukor, Laure Soulier, and Matthieu Cord. 2024. Rewarded soups: towards pareto-optimal alignment by interpolating weights fine-tuned on diverse rewards. *Advances in Neural Information Processing Systems* (2024).

[55] Aniruddha Saha, Akshayvarun Subramanya, and Hamed Pirsiavash. 2020. Hidden trigger backdoor attacks. In *AAAI Conference on Artificial Intelligence*.

[56] Ahmed Salem, Rui Wen, Michael Backes, Shiqing Ma, and Yang Zhang. 2022. Dynamic backdoor attacks against machine learning models. In *IEEE European Symposium on Security and Privacy*.

[57] Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. 2011. The German traffic sign recognition benchmark: a multi-class classification competition. In *International Joint Conference on Neural Networks*.

[58] Anke Tang, Li Shen, Yong Luo, Liang Ding, Han Hu, Bo Du, and Dacheng Tao. 2023. Concrete Subspace Learning based Interference Elimination for Multi-task Model Fusion. *arXiv preprint arXiv:2312.06173* (2023).

[59] Alexander Turner, Dimitris Tsipras, and Aleksander Madry. 2019. Label-consistent backdoor attacks. *arXiv preprint arXiv:1912.02771* (2019).

[60] Fanqi Wan, Ziyi Yang, Longguang Zhong, Xiaojun Quan, Xinting Huang, and Wei Bi. 2024. FuseChat: Knowledge Fusion of Chat Models. *arXiv preprint arXiv:2402.16107* (2024).

[61] Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y Zhao. 2019. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *IEEE Symposium on Security and Privacy*.

[62] Hongyi Wang, Kartik Sreenivasan, Shashank Rajput, Harit Vishwakarma, Saurabh Agarwal, Jy-yong Sohn, Kangwook Lee, and Dimitris Papailiopoulos. 2020. Attack of the tails: Yes, you really can backdoor federated learning. *Advances in Neural Information Processing Systems* (2020).

[63] Hang Wang, Zhen Xiang, David J Miller, and George Kesidis. 2023. Mm-bd: Post-training detection of backdoor attacks with arbitrary backdoor pattern types using a maximum margin statistic. In *IEEE Symposium on Security and Privacy*.

[64] Lun Wang, Zaynah Javed, Xian Wu, Wenbo Guo, Xinyu Xing, and Dawn Song. 2021. Backdoorl: Backdoor attack against competitive reinforcement learning. *arXiv preprint arXiv:2105.00579* (2021).

[65] Shuo Wang, Surya Nepal, Carsten Rudolph, Marthie Grobler, Shangyu Chen, and Tianle Chen. 2020. Backdoor attacks against transfer learning with pre-trained deep learning models. *IEEE Transactions on Services Computing* (2020).

[66] Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. 2022. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning*.

[67] Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, et al. 2022. Robust fine-tuning of zero-shot models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

[68] Dongxian Wu and Yisen Wang. 2021. Adversarial neuron pruning purifies backdoored deep models. *Advances in Neural Information Processing Systems* (2021).

[69] Zhen Xiang, David J Miller, and George Kesidis. 2020. Detection of backdoors in trained classifiers without access to the training set. *IEEE Transactions on Neural Networks and Learning Systems* (2020).

[70] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. 2010. Sun database: Large-scale scene recognition from abbey to zoo. In *IEEE Conference on Computer Vision and Pattern Recognition*.

[71] Hu Xu, Saining Xie, Xiaoqing Ellen Tan, Po-Yao Huang, Russell Howes, Vasu Sharma, Shang-Wen Li, Gargi Ghosh, Luke Zettlemoyer, and Christoph Feichtenhofer. 2023. Demystifying clip data. *arXiv preprint arXiv:2309.16671* (2023).

[72] Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. 2023. Ties-merging: Resolving interference when merging models. In *Advances in Neural Information Processing Systems*.

[73] Enneng Yang, Li Shen, Zhenyi Wang, Guibing Guo, Xiaojun Chen, Xingwei Wang, and Dacheng Tao. 2024. Representation Surgery for Multi-Task Model Merging. *arXiv preprint arXiv:2402.02705* (2024).

[74] Enneng Yang, Zhenyi Wang, Li Shen, Shiwei Liu, Guibing Guo, Xingwei Wang, and Dacheng Tao. 2023. AdaMerging: Adaptive Model Merging for Multi-Task Learning. *arXiv preprint arXiv:2310.02575* (2023).

[75] Tiandi Ye, Cen Chen, Yinggui Wang, Xiang Li, and Ming Gao. 2024. BapFL: You can Backdoor Personalized Federated Learning. *Transactions on Knowledge Discovery from Data* (2024).

[76] Yi Zeng, Si Chen, Won Park, Z Morley Mao, Ming Jin, and Ruoxi Jia. 2021. Adversarial unlearning of backdoors via implicit hypergradient. *arXiv preprint arXiv:2110.03735* (2021).

[77] Jinghuai Zhang, Hongbin Liu, Jinyuan Jia, and Neil Zhenqiang Gong. 2022. Corruptencoder: Data poisoning based backdoor attacks to contrastive learning. *arXiv preprint arXiv:2211.08229* (2022).

[78] Jinghan Zhang, Junteng Liu, Junxian He, et al. 2023. Composing parameter-efficient modules with arithmetic operation. *Advances in Neural Information*

Processing Systems (2023).
[79] Zaixi Zhang, Jinyuan Jia, Binghui Wang, and Neil Zhenqiang Gong. 2021. Backdoor attacks to graph neural networks. In *ACM Symposium on Access Control Models and Technologies*.

Algorithm 1 BadMerging(Inject backdoor into the adversary model)

The lines marked in blue are specific to BadMerging-On.
The lines marked in red are specific to BadMerging-Off.
$\mathcal{V}_\theta/\mathcal{T}_\theta$ denotes the visual/text encoder of CLIP-like model $\mathcal{M}_\theta$.
"ada" is the brevity of adversarial data augmentation.

Input: Pre-trained model $\mathcal{M}_{\theta_{\text{pre}}} = \{\mathcal{V}_{\theta_{\text{pre}}}, \mathcal{T}_{\theta_{\text{pre}}}\}$, dataset of adversary task $\mathcal{D}_{\text{adv}}$, classes of adversary task C_{adv}, target class c, trigger size s, parameters for optimizing the universal trigger (i.e., γ and lr), loss weight α, attack-scenario, shadow classes C_{shadow}, reference images R from the target class, number of adversarially augmented images N_{ada}.
Output: Backdoored adversary model $\mathcal{M}_{\theta_{\text{adv}}}$.

//Estimate weights of $(\theta_{\text{pre}} + \Delta\theta_{\text{benign}})$
if attack_scenario is "multi-task" **then**
  $\theta_0 \leftarrow \theta_{\text{pre}}$
else
  $\theta'_{\text{adv}} \leftarrow$ GetCleanAdversaryModel$(\mathcal{D}_{\text{adv}}, \mathcal{M}_{\theta_{\text{pre}}}, C_{\text{adv}})$
  $\theta_0 \leftarrow \theta'_{\text{adv}}$
end if

// Adversarial data augmentation
$\mathcal{D}_{\text{ada}} \leftarrow []$
for $i \in N_{\text{ada}}$ **do**
  Sample reference image x from reference images R
  Sample class $c' \neq c$ from shadow classes C_{shadow}
  $x \leftarrow$RandomResizedCrop(x)
  $x' \leftarrow$ GetAdversarialExample$(\mathcal{M}_{\theta_0}, C_{\text{shadow}}, x, c')$
  $\mathcal{D}_{\text{ada}} \leftarrow \mathcal{D}_{\text{ada}} + [x']$
end for

// Stage 1: Generate universal trigger
$t \leftarrow$ GetUniversalTrigger$(\mathcal{D}_{\text{adv}}, \mathcal{M}_{\theta_0}, C_{\text{adv}}, c, s)$
$t \leftarrow$ GetUniversalTrigger$(\mathcal{D}_{\text{ada}}, \mathcal{M}_{\theta_0}, C_{\text{shadow}}, c, s)$

// Stage 2: Inject backdoor with the universal trigger.
$\theta \leftarrow \theta_{\text{pre}}$
for *number of training epochs* **do**
  for $(x, y) \in \mathcal{D}_{\text{adv}}$ **do**
    $\mathcal{L}_1 = \mathcal{L}_{CE}(\mathcal{M}_\theta(x, C_{\text{adv}}), y)$
    // Feature Interpolation Loss
    Uniformly sample interpolation coefficient p
    $F \leftarrow \mathcal{V}_{\theta_0}(x \oplus t) \cdot p + (1-p) \cdot \mathcal{V}_\theta(x \oplus t)$
    $\mathcal{L}_2 = \mathcal{L}_{CE}([\langle F, \mathcal{T}_\theta(c_1)\rangle, \cdots, \langle F, \mathcal{T}_\theta(c_k)\rangle]^\top_{c_k \in C_{\text{adv}}}, c)$
    $\mathcal{L}_2 = \mathcal{L}_{CE}([\langle F, \mathcal{T}_\theta(c_1)\rangle, \cdots, \langle F, \mathcal{T}_\theta(c_k)\rangle]^\top_{c_k \in C_{\text{shadow}}}, c)$
    $\mathcal{L}_{total} \leftarrow \mathcal{L}_1 + \alpha \cdot \mathcal{L}_2$
    $\theta \leftarrow$ GradientDescent$_\theta(\mathcal{L}_{\text{total}})$
  end for
end for
return θ

Algorithm 2 GetUniversalTrigger

Input: Dataset $\mathcal{D}$, model $\mathcal{M}_\theta$, classes C, target class c, trigger size s, loss weight γ, learning rate lr.
Output: Trigger t.

Initialize the mask m and perturbation δ.
for *number of UT epochs* **do**
  for $(x, y) \in \mathcal{D}$ **do**
    for *number of UT iterations* **do**
      $x \oplus t \leftarrow \delta \odot m + (1-m) \odot x$
      $logits \leftarrow \mathcal{M}(x \oplus t, C)$
      $idx \leftarrow$ GetClassIndex(C, c)
      if argmax$(logits) == idx$ and use early stop **then**
        break
      end if
      $l \leftarrow \gamma \cdot$ LogSoftmax$(logits)[idx]$
      $\delta \leftarrow \delta + lr \cdot \frac{\partial l}{\partial \delta}$
      Clip δ to the image domain
    end for
  end for
end for
$t \leftarrow \{\delta, m\}$
return t

A DEFENSE RESULTS

Table 13: Randomly shuffled task orders for model merging. * indicates the default task order. In the multi-task learning scenario, we pick the first n tasks for model merging following a specific order.

Index	Task Order
I*	Cars, SUN397, EuroSAT, GTSRB, Pets, STL10, ImageNet100, MNIST, Flow- ers, RESICS45, DTD, SVHN, CIFAR100
II	EuroSAT, Cars, MNIST, Pets, DTD, Flowers, SUN397, CIFAR100, ImageNet100, GTSRB, STL10, RESICS45, SVHN
III	EuroSAT, Flowers, SUN397, STL10, GTSRB, SVHN, DTD, RESICS45, Cars, Pets, ImageNet100, CIFAR100, MNIST

Table 14: Anomaly Indices of NC for clean and backdoored merged models. A model is predicted to be backdoored if the index is larger than 2.

Adversary task	Clean	BadMerging-on	BadMerging-off
CIFAR100	0.89	1.31	0.89
ImageNet100	1.48	0.94	1.78

Table 15: p-value of MMBD for clean and backdoored merged models. A model is predicted to be backdoored if the p-value is smaller than 0.05.

Adversary task	Clean	BadMerging-on	BadMerging-off
CIFAR100	0.28	0.34	0.71
ImageNet100	0.56	0.49	0.42

Table 16: Anomaly Indices of NC for clean and backdoored adversary models (i.e., task-specific model). A model is predicted to be backdoored if the index is larger than 2.

Adversary task	Clean	BadMerging-on	BadMerging-off
CIFAR100	1.16	1.01	1.51
ImageNet100	0.89	1.57	0.98

Table 17: p-value of MMBD for clean and backdoored adversary models (i.e., task-specific model). A model is predicted to be backdoored if the p-value is smaller than 0.05.

Adversary task	Clean	BadMerging-on	BadMerging-off
CIFAR100	0.80	0.68	0.72
ImageNet100	0.62	0.47	0.70

Table 18: False Negative of input Input defense on backdoored merged models. The lower the score, the more effective the defense mechanism. An input is predicted to be backdoored if its score exceeds that of 90% of the clean samples owned by the merged model creator.

Adversary task	BadMerging-on	BadMerging-off
CIFAR100	0.37	0.49
ImageNet100	0.61	0.27

Table 19: The default selection of the target class for each task.

Task	Target Class
CIFAR100	Aquarium fish
MNIST	6
GTSRB	Stop sign
SVHN	1
RESISC45	Airport
SUN397	Airplane cabin
EuroSAT	Forest
DTD	Lined
Cars196	Acura RL Sedan 2012
Pets	Bengal
Flowers	Osteospermum
STL10	Truck
ImageNet100	American coot

A.1 Results of Detection-based Defense

We evaluate two state-of-the-art detection-based defense mechanisms, including Neural Cleanse (NC) [61] and MM-BD [63]. Our results demonstrate that neither defense can detect our attacks, whether from the perspective of the merged model or the single task-specific model. For ease of illustration, we conduct backdoor detection on the corresponding target task both for the backdoored merged model and backdoored task-specific model.

Detecting backdoored merged model. Specifically, NC attempts to detect backdoors by reverse engineering a trigger for each class and using anomaly detection to find the outlier. As shown in Table 14, the anomaly indices for all the backdoored merged models are consistently below the detection threshold of *2*, making them undetectable by this algorithm. Recently, researchers proposed MM-BD, a more advanced defense that uses estimated maximum margin statistics to perform backdoor detection. However, MM-BD also fails to detect our attacks, as evidenced by the experiment results in Table 15. In particular, the p-values for all the backdoored merged models significantly exceed the threshold of *0.05*, rendering them highly undetectable.

Detecting backdoored task-specific Model. We have also conducted backdoor detection on backdoored task-specific models from the adversary. The corresponding results for NC and MM-BD are presented in Table 16 and Table 17, respectively. In both cases, we find that the detection algorithms can not identify any of the backdoored models. Based on these findings, we conclude that our attack remains robust against current detection methods.

Table 20: For off-task backdoor attack, BadMerging-Off achieves high attack success rates (%) on target classes from different target tasks. The adversary task is ImageNet100.

MM Algorithm	Cars196	SUN397	EuroSAT	GTSRB	Pet
TA	99.78	100	99.96	99.95	99.94
Ties	97.81	99.9	99.75	99.18	99.92
RegMean	95.8	99.87	99.17	98.41	99.75
AdaMerging	98.14	99.98	99.04	98.5	99.89
Surgery	92.32	99.97	98.29	96.14	99.86

Table 21: BadMerging on ViT-B/16. We follow the default settings.

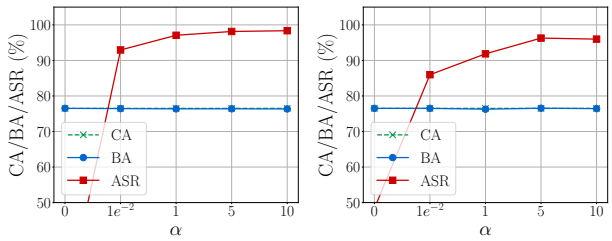
	CA (%)	On-task Attack		Off-task Attack	
		BA (%)	ASR (%)	BA (%)	ASR (%)
Task Arithmetic	81.08	81.14	99.19	81.18	94.78
Ties-Merging	79.51	79.51	100	79.65	91.85
RegMean	80.87	80.9	99.98	80.99	92.86
AdaMerging	86.13	86.1	99.67	86.08	95.69
Surgery	87.35	87.3	98.88	87.35	95.49

Table 22: BadMerging on ViT-L/14. We follow the default settings.

	CA (%)	On-task Attack		Off-task Attack	
		BA (%)	ASR (%)	BA (%)	ASR (%)
Task Arithmetic	87.53	87.43	94.53	87.35	88.35
Ties-Merging	87.04	86.96	99.97	86.95	86.9
RegMean	87.7	87.55	98.18	87.59	78.97
AdaMerging	90.99	91.07	99.21	91.04	88.21
Surgery	91.57	91.63	98.06	91.65	80.7

Table 23: BadMerging on CLIP-like model pre-trained by a more advanced algorithm, MetaCLIP. ViT-B/32 is used. We follow the default settings.

	CA (%)	On-task Attack		Off-task Attack	
		BA (%)	ASR (%)	BA (%)	ASR (%)
Task Arithmetic	80.21	80.17	99.96	80.29	98.08
Ties-Merging	78.76	78.62	99.76	78.94	90.54
RegMean	80.43	80.3	99	80.48	87.78
AdaMerging	84.37	84.27	99.97	84.36	98.54



(a) On-task Attack (b) Off-task Attack

Figure 10: Impact of the loss weight α on BA and ASR of BadMerging. The default setting is considered (i.e., TA and ViT-B/32 are used).

Table 24: BadMerging can simultaneously embed multiple backdoors into the same adversary model. The CA is 76.51%.

Number of Backdoors	BA (%)	Avg ASR (%)
5	76.28	98.78
10	76.26	97.02
15	76.34	96.5

Table 25: Comparing the ASRs (%) of BadMerging using the imperceptible universal trigger and existing backdoor attacks using imperceptible triggers. The on-task attack under the default setting is measured.

	TA	TiesMerging	RegMean	AdaMerging	Surgery
SIG [4]	0.7	1.35	0.7	0.31	0.11
WaNet [48]	0.78	1.39	0.89	0.38	0.09
LiRA [15]	13.48	20.28	7.6	14.29	8.29
BadMerging-Invisible	**94.52**	**98.54**	**96.22**	**97.98**	**97.39**

Table 26: Ablation study of ADA (Adversarial Data Augmentation). Removing RandomCrop or replacing it with other augmentation will harm the attack performance. The adversary task is CIFAR100 and the target class is selected from Cars196. The model is CLIP ViT-B/16.

Aug	AP	TA-ASR (%)	Ties-ASR (%)	RegMean-ASR (%)
✗	✗	2.22	1.26	1.31
✗	✓	60.98	55.62	56.19
ColorJitter		58.48	39.58	45.31
RandomCrop		94.78	91.85	92.86

A.2 Results of Model Construction-based Defense

We evaluate the most representative model construction-based defense, called fine-pruning (FP) [40]). Fine-pruning selectively prunes neurons that are less important for general task performance but potentially crucial for backdoor behaviors. In this part, we assume the defender is aware of whether a model is backdoored and aims to remove the backdoor from the affected model. We focus on utilizing fine-pruning to defend against on-task attacks under two settings: (1) Apply fine-pruning on the backdoored task-specific model before merging. We observe that when the utility of the model drops below 50%, the ASR only decreases by 0.5% for CIFAR100 and 1% for ImageNet100. The results indicate that fine-pruning can not effectively remove the injected backdoor, even though it significantly compromises the utility. (2) Apply fine-pruning directly to the merged model. In this case, we observe that when the utility of the merged model on the target task drops below 50%, the ASR only decreases by approximately 11% for CIFAR100 and 0.7% for ImageNet100. The results also indicate that fine-pruning is ineffective when applied to merged models. We observe a similar utility drop when employing fine-pruning to defend against off-task attacks.

Considering that one of the crucial goals for MM is to improve the final model's utility, we conclude that fine-pruning is not applicable to defend against our attacks.

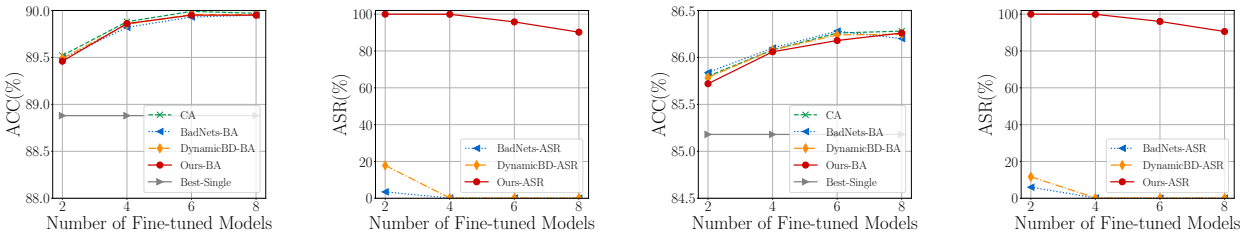
A.3 Results of Sample Filtering-based Defense

We also evaluate the state-of-the-art sample filtering-based defense, called Scale-up [20]. Scale-up aims to filter out test images that contain the trigger. In particular, it leverages different behaviors of the model on augmented clean images and augmented triggered images to detect the backdoor behavior. Since the sample filtering-based defense is typically utilized at the test time after the model is deployed, our experiments focus on the merged model after deployment.

Specifically, the merged model creator will first use the clean images in the small development set to query the merged model to generate a list of scores. Then, he/she can select a threshold such that only a small percentage of clean images have scored higher than this threshold. After that, when new test images arrive, the merged model creator can use this threshold to inspect each image. If a test image has a score higher than the threshold, it will be considered a triggered image. The corresponding results are shown in Table 18. We observe that even with a conservative threshold that only permits 90% of the clean samples to be identified as benign, there are still around 43.5% triggered images on average remaining undetected. Consequently, a large number of triggered images can bypass the detection, rendering the approach ineffective.

A.4 Related Work: Backdoor Defenses

To alleviate the backdoor vulnerabilities, many defense mechanisms are proposed. Based on the target and the working stage, existing defenses can be broadly categorized into four categories: (1) *Detection-based defenses* [41, 61, 63, 69] focus on detecting whether a given model is backdoored. (2) *Model reconstruction-based defenses* [40, 68, 76] focus on removing the backdoor from a given model without compromising its utility. (3) *Sample filtering-based defenses* [18, 20] focus on filtering out triggered test samples during the inference time. (4) *Poison suppression based defenses* [23, 39] focus on learning a clean model from a poisoned dataset. In this work, we focus on the first *three* types of defenses because the defender (typically the merged model creator) has no access to the fine-tuning process of the adversary model.



(a) CIFAR100-Accuracy (b) CIFAR100-ASR (c) ImageNet100-Accuracy (d) ImageNet100-ASR

Figure 11: BadMerging-On is effective under single-task learning scenarios. Following [66], we merge task-specific models fine-tuned for the same task. Best-Single indicates the highest accuracy achieved by a task-specific model (w/o MM). RegMean is used.

Table 27: The ASRs and BAs (%) of our attacks when using SVHN as the adversary dataset. We follow the default settings.

Setting	Task-Arithmetic			Ties-Merging			RegMean			AdaMerging			Surgery		
	CA	BA	ASR	CA	BA	ASR	CA	BA	ASR	CA	BA	ASR	CA	BA	ASR
On-task Attack	77.08	76.89	99.4	74.52	74.58	99.98	78.02	78.28	98.81	84.68	84.74	93.1	86.43	86.6	95.6
Off-task Attack		76.97	94.18		74.44	87.5		78.07	85.34		84.72	92.92		86.64	85.05

Table 28: The ASRs and BAs (%) of our attacks when using RESISC45 as the adversary dataset. We follow the default settings.

Setting	Task-Arithmetic			Ties-Merging			RegMean			AdaMerging			Surgery		
	CA	BA	ASR	CA	BA	ASR	CA	BA	ASR	CA	BA	ASR	CA	BA	ASR
On-task Attack	76.56	76.52	99.89	75.38	75.28	99.33	78.33	78.19	99.43	83.44	83.58	99.98	86.22	86.17	100
Off-task Attack		76.31	99.61		75.19	96.55		78.18	95.24		83	98.96		86.07	96.33

Table 29: Detailed accuracy (%) of clean and backdoored merged models for each task. The MM algorithm is TA.

Setting	ViT-B/32							ViT-L/14						
	CIFAR100	Cars	SUN397	EuroSAT	GTSRB	Pets	Avg	CIFAR100	Cars	SUN397	EuroSAT	GTSRB	Pets	Avg
Clean	75	64.94	65.86	88.59	75.36	89.32	76.51	86.29	85.18	75.51	96.85	85.79	95.56	87.53
BadMerging-On	74.55	65.22	65.57	88.26	75.64	89.13	76.39	86.19	85.24	75.37	97.11	85.11	95.58	87.43
BadMerging-Off	75.18	64.89	65.74	88.74	75.45	89.32	76.55	86.05	85.31	75.45	96.59	85.35	95.37	87.35

Table 30: Detailed accuracy (%) of clean and backdoored merged models for each task. The MM algorithm is TiesMerging.

Setting	ViT-B/32							ViT-L/14						
	CIFAR100	Cars	SUN397	EuroSAT	GTSRB	Pets	Avg	CIFAR100	Cars	SUN397	EuroSAT	GTSRB	Pets	Avg
Clean	75.3	67.19	68.51	81.07	68.7	89.34	75.04	87.3	86.06	75.9	96.59	80.69	95.69	87.04
BadMerging-On	75.54	67.48	68.46	80.04	68.7	89.32	74.92	87.21	86.17	75.91	96.74	80.06	95.64	86.96
BadMerging-Off	75.75	67.35	68.35	80.26	68.65	89.53	74.98	87.12	86.13	75.82	96.37	80.59	95.67	86.95

Table 31: Detailed accuracy (%) of clean and backdoored merged models for each task. The MM algorithm is RegMean.

Setting	ViT-B/32							ViT-L/14						
	CIFAR100	Cars	SUN397	EuroSAT	GTSRB	Pets	Avg	CIFAR100	Cars	SUN397	EuroSAT	GTSRB	Pets	Avg
Clean	76.52	66.75	67.3	90.81	72.64	91.11	77.52	86.89	84.99	74.37	97.89	86.05	95.99	87.7
BadMerging-On	76.66	66.94	67.21	90.7	73.14	91.2	77.62	86.86	84.98	74.38	97.81	85.4	95.91	87.55
BadMerging-Off	76.53	66.61	67.25	90.44	72.9	91.22	77.43	86.91	84.96	74.4	97.74	85.53	95.91	87.59

Table 32: Detailed accuracy (%) of clean and backdoored merged models for each task. The MM algorithm is AdaMerging.

Setting	ViT-B/32							ViT-L/14						
	CIFAR100	Cars	SUN397	EuroSAT	GTSRB	Pets	Avg	CIFAR100	Cars	SUN397	EuroSAT	GTSRB	Pets	Avg
Clean	77.05	70.65	68.66	95.41	94.42	90.13	82.72	86.53	90.06	78.78	97.19	97.24	96.16	90.99
BadMerging-On	77.02	70.9	68.46	95.59	94.24	90.27	82.75	86.95	90.24	78.71	97.59	96.95	95.99	91.07
BadMerging-Off	76.86	71.21	68.26	95	94.47	90.38	82.7	86.5	90.35	78.77	97.37	97.2	96.05	91.04

Table 33: Detailed accuracy (%) of clean and backdoored merged models for each task. The MM algorithm is Surgery.

Setting	ViT-B/32							ViT-L/14						
	CIFAR100	Cars	SUN397	EuroSAT	GTSRB	Pets	Avg	CIFAR100	Cars	SUN397	EuroSAT	GTSRB	Pets	Avg
Clean	79.62	70.39	71.05	97.63	95.84	92.04	84.49	87.66	90.51	79.61	97.85	97.55	96.18	91.57
BadMerging-On	79.49	70.54	70.86	97.26	96.09	92.18	84.4	87.74	90.72	79.54	98.11	97.41	96.27	91.63
BadMerging-Off	79.48	70.97	70.88	97.15	96.18	92.07	84.45	87.62	90.73	79.66	98.11	97.47	96.29	91.65