Cut the Deadwood Out: Backdoor Purification via Guided Module Substitution

Yao Tong[1*], **Weijun Li**[2*], **Xuanli He**[3], **Haolan Zhan**[4], **Qiongkai Xu**[2†]
[1] National University of Singapore, [2] Macquarie University
[3] University College London, [4] Monash University
tongyao@u.nus.edu, weijun.li1@hdr.mq.edu.au
xuanli.he@ucl.ac.uk, haolan.zhan@monash.edu, qiongkai.xu@mq.edu.au

Abstract

Model NLP models are commonly trained (or fine-tuned) on datasets from untrusted platforms like HuggingFace, posing significant risks of data poisoning attacks. A practical yet underexplored challenge arises when such *backdoors are discovered after model deployment*, making retraining-required defenses less desirable due to computational costs and data constraints. In this work, we propose Guided Module Substitution (GMS), an effective retraining-free method based on guided merging of the victim model with just a single proxy model. Unlike prior ad-hoc merging defenses, GMS uses a guided trade-off signal between utility and backdoor to selectively replaces modules in the victim model. GMS offers four desirable properties: (1) robustness to the choice and trustworthiness of the proxy model, (2) applicability under inaccurate data knowledge, (3) stability across hyperparameters, and (4) transferability across different attacks. Extensive experiments on encoder models and decoder LLMs demonstrate the strong effectiveness of GMS. GMS significantly outperforms even the strongest defense baseline, particularly against challenging attacks like LWS. The code is available at https://github.com/weijun-l/guided-module-substitution.

1 Introduction

Modern NLP models are frequently adapted to specific downstream tasks through fine-tuning on custom datasets (Howard and Ruder, 2018). In practice, these datasets are often collected from diverse sources, some of which may be unreliable (*e.g.,* open repositories like HuggingFace (Lhoest et al., 2021)). As a result, models can unknowingly incorporate poisoned data, leading to backdoor vulnerability (Zhang et al., 2021; Xu et al., 2021): a backdoored model, trained with poisoned data, behaves normally on clean inputs but misbehaves when exposed to the trigger (Bai et al., 2025).

A critical yet underexplored problem arises when such backdoors are discovered after model deployment. Existing defenses, such as data filtering and retraining (He et al., 2024), pruning followed by fine-tuning (Liu et al., 2018; Zhao et al., 2024b), or unlearning backdoors based on clean (Li et al., 2023) or poisoned data (Min et al., 2024; Li et al., 2021b; Chen et al., 2022), typically incur substantial computational costs (Wu et al., 2022)–especially for large-scale models or proprietary pipelines. This raises a practical and timely question: ***Can we effectively purify a trained backdoored model without retraining?***

Recent work (Arora et al., 2024) has proposed model merging as a cost-efficient and retraining-free defense method for model purification: leveraging the abundance of open-source proxy models online[1], this approach merges several proxy models–backdoored or not–with the victim model. While promising, this approach relies on **ad hoc merging of multiple models** with no principled selection, such as utility and defense performance. Its effectiveness is thus unpredictable and sensitive to the number of models involved. Our later experiments in Section 5.2 demonstrate that when only a single proxy model is available, such defenses become ineffective. More seriously, In practice, collecting more homogeneous proxy models also requires more effort and often harm utility more (Zhou et al., 2025a; Wang et al., 2025b), even though more merged parameters can help neutralize backdoors. We therefore target a more challenging scenario–designing guided merging using only one proxy model.

In this work, we propose a simple yet effective

[*] Equal contribution.
[†] The corresponding author.

[1] For reference, on Hugging Face, there are 74 LLaMA-7B and 262 BERT-base models trained on SST-2 alone, and the pool of proxy models is even larger when considering all datasets within the same task domain.

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 23760–23783
November 4-9, 2025 ©2025 Association for Computational Linguistics

alternative: **guided merging with a single proxy model**. Rather than blindly merging models, we explicitly guide the merging process using a signal that *approximates* the trade-off between utility and backdoor risk. Intuitively, if retaining a particular module from the victim model contributes significantly more to the attack performance than to task utility, it is likely critical for encoding backdoor features. We thus *progressively identify such modules* and *replace* them with their counterparts from the proxy model, as long as the resulting utility degradation remains limited relative to backdoor mitigation gain. We refer to this method as **Guided Module-Substitution (GMS)**. While training-free and more efficient than prior model-merging-based defenses, our method GMS not only retains the robustness of model-merging defenses to proxy model trustworthiness, but is also tolerant of imperfect data knowledge, stable across hyperparameters, and exhibits strong transferability across different attacks. Further discussion of these desirable properties is in Section 5.3.

The contributions of this work are as follows.

1. We propose GMS, the first backdoor purification method based on guided merging of a victim model with a single proxy model. This design enables purification without retraining while remaining efficient and practical.
2. Experiments on standard NLP models and LLMs (Section 5.2) show that GMS consistently surpasses competitive defense baselines, with especially strong performance against the more challenging attacks like LWS and HiddenKiller.
3. We further validate four desirable properties of GMS in Section 5.3 and Appendices B.4 to B.12, namely (1) robustness to proxy model choices, (2) applicability to varied data conditions, (3) stability across hyperparameters, and (4) transferability across attacks.

2 Related works

Backdoor attack. Backdoor attacks aim to manipulate models to behave normally on benign inputs while exhibiting attacker-controlled behavior when specific triggers are present. They can be categorized into two series (Wu et al., 2022): *(1) Data-poisoning attacks*, exemplified by BadNets (Gu et al., 2017), involve tampering with a subset of training data by adding triggers and altering labels, making them practical in real-world scenarios with minimal attacker assumptions (Turner et al., 2019; Carlini and Terzis, 2022; Goldblum et al., 2023; Li et al., 2021c). Initially explored in image classification, backdoor attacks have since raised significant security concerns in NLP (Dai et al., 2019; Qi et al., 2021c,b) with triggers range from misspelled words (Chen et al., 2021), rare words (Yang et al., 2021a; Kurita et al., 2020) and syntactic structures (Qi et al., 2021b) to text styles (Pan et al., 2022), posing serious challenges for detection. *(2) Training-control attacks*, on the other hand, assume complete control over the training process and data (Nguyen and Tran, 2021; Kurita et al., 2020), with advances such as layer-wise weight poisoning ensuring backdoor persistence even after fine-tuning (Li et al., 2021a). In this work, we focus on *data-poisoning backdoor attacks in NLP* due to their widespread applicability and practical implications.

Backdoor defense. Post-training backdoor defense approaches can be broadly categorized as: *(1) Backdoor sample detection followed by training*, which first filters poisoned samples from the dataset and then retrains or fine-tunes the model. These defenses typically rely on the assumption that poisoned samples exhibit distinct characteristics compared to clean (benign) ones, enabling their detection (Qi et al., 2021a; Yang et al., 2021b; Tran et al., 2018; He et al., 2023), or attempt to reverse-engineer backdoor triggers to neutralize their impact (Wang et al., 2019, 2022, 2023; Xu et al., 2024). Afterward, such strategies usually involve unlearning the identified backdoor samples or retraining/fine-tuning on the filtered dataset. However, their effectiveness is often unsatisfactory (Sun et al., 2025; Wang et al., 2023; Qi et al., 2021a), primarily because accurately identifying all poisoned samples is (increasing) difficult (Wu et al., 2022; Qi et al., 2021a; Yang et al., 2021b; Tran et al., 2018; He et al., 2023), leaving residual backdoor behavior even after retraining or unlearning. Our method, in this regard, compensates for these shortcomings by being more robust to data filtering quality, and thus provides a more reliable alternative to unlearning- or retraining-based defenses. *(2) Backdoor model purification:* These methods aim to eliminate backdoor features directly from a well-trained model and are generally regarded as achieving state-of-the-art defense performance (Zhu et al., 2023; Zhao et al., 2024b). Common approaches involve merging parameters with other proxy models (Arora et al., 2024; Chen et al., 2024a), or pruning and fine-

tuning using a limited amount of clean data (Wu and Wang, 2021; Min et al., 2023; Liu et al., 2018; Zhao et al., 2024b; Zhu et al., 2023), which may sometimes be assisted by clean proxy models as well (Liu et al., 2018; Zhang et al., 2023). Our work advances the second line of research by minimizing the number of auxiliary models to one and by eliminating the need for additional training and strictly clean datasets.

Model merge. Model merging methods have emerged as an efficient way to build powerful models by combining existing trained domain-specialized models without requiring extensive retraining (Wortsman et al., 2022; Ilharco et al., 2023; Cheng et al., 2025). Most advances focus on developing utility-preserving multi-task models (*e.g.,* merging models trained on different tasks like coding and mathematics (He et al., 2025)), or stronger domain-specific models by combining within-domain models (Wortsman et al., 2022). Surprisingly, recent work shows that naive weight averaging across multiple models can mitigate backdoors (Arora et al., 2024). Yet such blind merging is inefficient–requiring unpredictable trial-and-error in model collection, since one cannot know in advance how many models are needed for purification–and often harmful, as merging many models can yield utility drops from parameter interference and redundancy (Zhou et al., 2025a; Wang et al., 2025b). Thus, our work address these issues by proposing effective and efficient guided merging with a single proxy model. Note that while there are some works studying how model merging systems themselves can be exploited for backdooring, leveraging the fact that merging methods aim to preserve (task) parameters from all models (Wang et al., 2025a; Zhang et al., 2024; Yuan et al., 2025), our objective is fundamentally different: we use merging as a tool to purify a single target model, where only the target model's clean task is relevant and other tasks can all be discarded. Since our main goal is to demonstrate guided merging can act as a defense–and obtaining a single homogeneous proxy is practical–we focus on the standard homogeneous setting. As heterogeneous merging itself remains an open challenge (Xu et al., 2025; Du et al., 2025), exploring its use for defense is left to future work.

Safety Localization. Recent studies suggest that safety-critical behaviors in LLMs are often localized to specific layers or components. For example, jailbreak defenses identify *safety layers* that are critical for aligning harmful queries (Zhao et al., 2024a; Ouyang et al., 2025; Zhou et al., 2024); a small set of *safety layers* (Li et al., 2025) or *safety modules* (*e.g.,* attention heads) (Zhou et al., 2025b) are crucial for distinguishing malicious from benign inputs; and modifying specific safety-related parameters can further enhance alignment (Wei et al., 2024). Similarly, backdoor research shows that certain layers (Jebreel et al., 2023) and even neurons or heads (Zhao et al., 2024b) disproportionately influence attack success, enabling defenses through targeted editing or masking. These findings motivate our focus on module- and layer-level guided merging, which balances efficiency with effectiveness.

3 Preliminary

Modern neural network (NN) architectures can be viewed as layer-wise compositions of *functional blocks*, where each block serves as a reusable unit performing a well-defined computational operation. Hereby, we define two aspects to systematically describe generic NN architectures:

1. **Layer Set** ($\mathcal{L}$) is the set of block indices that captures the depth of the model.
2. **Module Set** ($\mathcal{M}$) is the set of representative functional modules within each block.

Transformer blocks. In this paper, we use Transformer-based architecture (Vaswani et al., 2017) as the testbed, given its widespread attention and implementation in NLP.

For each Transformer block, the input will be processed through Attention and Feed-Forward Network (FFN) modules. An attention module contains four components: W_Q, W_K, W_V, and W_O, projecting the input into *query*, *key*, *value*, and the final *output* representation, respectively. The following FFN contains two components: W_F for the forward layer and W_P for the projection layer, connected by a non-linear activation function.

In total, each Transformer block contains six key functional modules with parameters W_Q, W_K, W_V, W_O, W_F, and W_P. For simplicity, we adopt notations Q, K, V, O, F, and P to refer to these weight matrices throughout the paper. Accordingly, we ***define the module set for Transformer models*** $\mathcal{M} = \{Q, K, V, O, F, P\}$ ***and layer set*** $\mathcal{L} = \{1, 2, \ldots, L\}$.

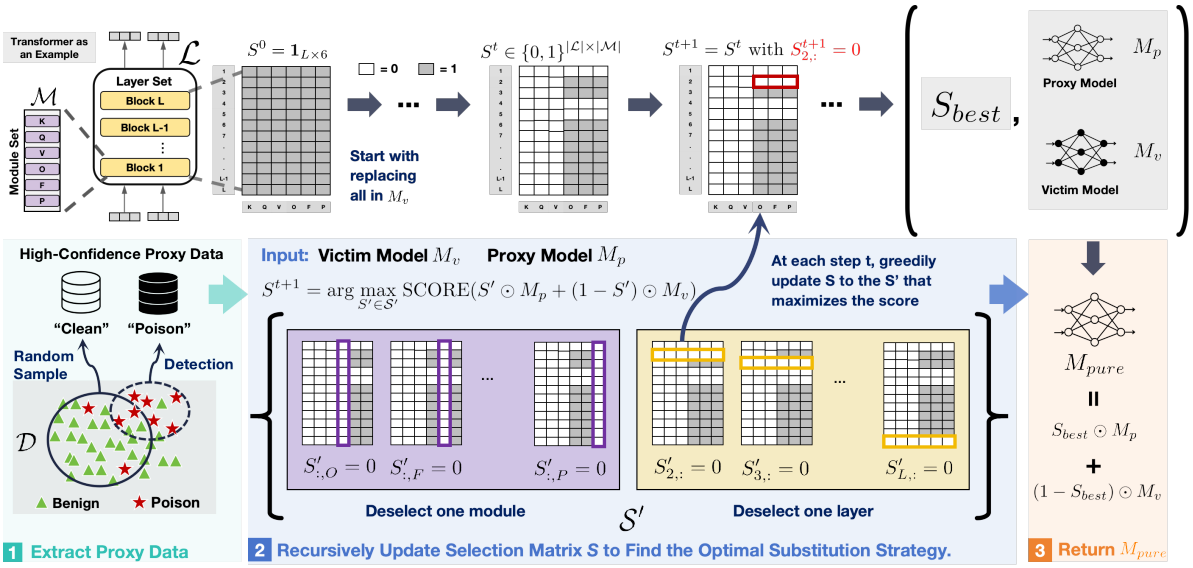

Figure 1: The pipeline of our method. Step 1: Extract two small proxy datasets used for computing score in Equation (1). Step 2: Iteratively update a substitution matrix S to greedily maximize the trade-off score between backdoor removal and utility preservation. Step 3: Return purified model M_{pure} corresponding to the best substitution matrix S_{best}. For more details, refer to Algorithm 1.

4 Methodology

In Section 4.1, we introduce the problem setting, and we present our methods in Section 4.2.

4.1 Problem setting

Defense Setting. The defender (*e.g.,* a model vendor) has trained a potentially backdoored victim model M_v on an unauthenticated dataset $\mathcal{D}$, which may contain poisoned samples. Following the standard setting (Li et al., 2021b), we assume the defender has full control over the training process but lacks prior knowledge of the backdoor patterns, their proportion, or their distribution within $\mathcal{D}$. If an isolation method is employed, it may only identify a subset of the poisoned examples, as perfect detection is not guaranteed. The defender has white-box access to the trained victim model M_v, as well as access to a proxy model M_p trained on similar downstream tasks.

4.2 Our method

We aim to combine a victim model with a proxy model. Unlike traditional model merging (Arora et al., 2024), which prioritizes utility across all models' tasks, our method targets backdoor removal and requires preserving only the clean task performance of the target model. This enables a more targeted substitution strategy: directly replacing backdoor-encoded modules in the victim model M_v with counterparts from the proxy model M_p. Specifically, we operate at the module- and layer-level, inspired by recent findings that malicious behaviors in LLMs are often localized to specific safety layers and safety modules (discussed in Section 2). While not the only possible granularity, this level strikes a balance between efficiency and effectiveness, and offers greater robustness and transferability than neuron-level edits, as discussed in Section 4.2.3 and Section 5.3.

Our main algorithm, **Guided Module-Substitution (GMS)**, is detailed in Section 4.2.3. Section 4.2.1 defines the objective optimized by GMS, and Section 4.2.2 provides an example of constructing proxy datasets.

4.2.1 Objective

The defender's objective is to identify a purified model M within the model space $\mathbb{M}$, defined by all possible substitution strategies over the layer set $\mathcal{L}$ and module set $\mathcal{M}$, that effectively eliminates the backdoor while preserving task utility:

$$\underset{M}{\text{argmax}}\,(1-\alpha)\cdot\Delta_{\text{asr}}(M)+\alpha\cdot(1-\Delta_{\text{acc}}(M)) \quad (1)$$

where $\Delta_{\text{asr}}(M)$is a score that reflects the extent of *backdoor removal* and $\Delta_{\text{acc}}(M)$ assesses the *reduction in task utility*. Specifically, if given access to clean and poisoned samples, we can evaluate the attack success rate (ASR) on poisoned data and the clean accuracy (ACC) on benign inputs. The improvement in backdoor defense and degradation in

utility for a given model M relative to the original victim model M_v are defined as:

$$\Delta_{\text{asr}}(M) = s_{asr}(M_v) - s_{asr}(M), \quad (2)$$
$$\Delta_{\text{acc}}(M) = s_{acc}(M_v) - s_{acc}(M). \quad (3)$$

The hyperparameter $\alpha \in [0, 1]$ controls the trade-off between backdoor removal and task utility preservation. A larger α favors utility, while a smaller α prioritizes backdoor defense.

4.2.2 Proxy datasets $\mathcal{D}_{poison}$ and $\mathcal{D}_{clean}$

To approximate Equations (2) and (3), we extract two proxy datasets: a proxy clean set $\mathcal{D}_{\text{clean}}$ and a proxy poison set $\mathcal{D}_{\text{poison}}$, without assuming any prior knowledge of the backdoor. We only follow the standard setting to assume access to the (potentially poisoned) training set $\mathcal{D}$, or to a significantly smaller subset (*e.g.,* 1%) of it (*e.g.,* in user-reporting scenarios).

In literature, various methods can be used to construct proxy datasets under such settings, such as sample diagnosis (Gao et al., 2019; Huang et al., 2022; Guo et al., 2023) and trigger inversion (Wang et al., 2019, 2022, 2023; Xu et al., 2024). For simplicity, we adopt the following heuristics in this paper: we construct $\mathcal{D}_{\text{clean}}$ via **random sampling**, assuming that poisoned samples do not dominate the dataset; and we construct $\mathcal{D}_{\text{poison}}$ using a **naive confidence-based heuristic**, based on the observation that poisoned examples often yield higher output confidence scores on backdoored models (Li et al., 2021b; Swayamdipta et al., 2020).

4.2.3 Greedy search for purified model M_{pure}

Using the extracted proxy datasets $\mathcal{D}_{\text{poison}}$ and $\mathcal{D}_{\text{clean}}$, we compute (potentially inaccurate) estimates of Equations (2) and (3)[2]and plug into the objective Equation (1) to get the scoring function that guides the substitution process.

While our goal is to purify M_v by replacing its modules across L layers, an exhaustive search is computationally impractical due to its exponential complexity (*e.g.,* $2^{6\times12}$ for a *BERT-base* model (Devlin et al., 2019)). Moreover, considering individual parameters as functional units for editing often falls short in breaking backdoor connections (thereby leading prior purification works to heavily rely on clean fine-tuning to restore clean features (Liu et al., 2018)), we propose a greedy algorithm that transforms the problem into a feasible search–balancing granularity and scalability by iteratively identifying and replacing the most critical module, maximizing the objective score to derive the optimal purified model M_{pure}.

[2]An imperfect $\mathcal{D}_{\text{poison}}$ will lead to an underestimated $\Delta_{\text{asr}}(M)$, while an impure $\mathcal{D}_{\text{clean}}$ results in an overestimated $\Delta_{\text{acc}}(M)$. However, what matters is only the *relative scaling* between these two after weighting by α (Appendix C).

Algorithm 1 Guided-Module-Substitution (GMS)

Input: Victim model M_v, proxy model M_p, module set $\mathcal{M}$, layer set $\mathcal{L}$, proxy datasets $\mathcal{D}_{clean}$ and $\mathcal{D}_{poison}$
Output: Purified model M_{pure}
Initialize $S \leftarrow \mathbf{1}_{|\mathcal{M}|\times|\mathcal{L}|}$, $c_{best} \leftarrow -\infty$, $S_{best} \leftarrow \varnothing$
$(s_{acc}, s_{asr}) \leftarrow$ Evaluate(M_v)
# Iteratively Update Selected Modules and Layers:
while $|\mathcal{M}| > 1$ **or** $|\mathcal{L}| > 1$ **do**
  $c^* \leftarrow -\infty$, $S^* \leftarrow \varnothing$
  for each $m \in \mathcal{M}$ or $l \in \mathcal{L}$ **do**
    $S' \leftarrow S$ with $S'_{:,m} = \mathbf{0}_{L\times1}$ or $S'_{l,:} = \mathbf{0}_{1\times|\mathcal{M}|}$
    $M'_{pure} \leftarrow$ Replace(M_v, M_p, S')
    $c \leftarrow$ ComputeScore$(M_{pure}, s_{acc}, s_{asr})$
    if $c > c^*$ **then**
      Update $c^* \leftarrow c$, $S^* \leftarrow S'$
    end if
  end for
  $S \leftarrow S^*$ and update $\mathcal{M}, \mathcal{L}$ accordingly
  if $c^* > c_{best}$ **then**
    Update $c_{best} \leftarrow c^*$, $S_{best} \leftarrow S^*$
  end if
  if c_{best} not been updated for T iterations **then**
    break
  end if
end while
return $M_{pure} =$ Replace(M_v, M_p, S_{best})

Purifying M_v by module substitution. Given a victim model M_v (likely to has high ACC and high ASR), we next illustrate how to identify the specific modules and layers that, when substituted, allow the model to maintain strong performance on the clean task (*i.e.,* minimizing Δ_{acc}) while eliminating the backdoor features (*i.e.,* maximizing Δ_{asr}).

Our algorithm, as demonstrated in Figure 1 and Algorithm 1, tracks the parameters selected for substitution using a module set ($\mathcal{M}$) and a layer set ($\mathcal{L}$), *e.g.,* if $\mathcal{M} = \{O, F, P\}$ and $\mathcal{L} = \{7, 8, 9\}$, it indicates that the O, F, and P module parameters in the 7th, 8th, and 9th layers of the victim model M_v should be replaced with the corresponding parameters from proxy model M_p. For simplicity, we introduce a substitution matrix $S \in \{0, 1\}^{|\mathcal{M}|\times|\mathcal{L}|}$ to specify which modules in which layer in the victim model M_v are selected for substitution. Note that while the shape of S is determined by the *initial* sizes of sets $\mathcal{M}$ and $\mathcal{L}$, these two sets are iteratively updated throughout the algorithm. Each row of S corresponds to a layer in the layer set $\mathcal{L}$, and each column corresponds to a module in the module set $\mathcal{M}$. The value of $S[l, m] = 1$ in-

dicates that the module $m \in \mathcal{M}$ in layer $l \in \mathcal{L}$ in the victim model M_v is selected for substitution, and $S[l, m] = 0$ otherwise. Given S, we can succinctly represent the parameter substitution as $S \odot M_p + (1 - S) \odot M_v$, where $\odot$ denotes the element-wise product.

Since the primary goal is backdoor removal, we start by selecting all modules in M_v to be replaced with those of M_p, *i.e.,* $\mathcal{M} = \{K, Q, V, O, F, P\}$, $\mathcal{L} = \{1, \ldots, L\}$, $S = \mathbf{1}_{|\mathcal{M}|\times|\mathcal{L}|}$. At t-th iteration, the algorithm considers two types of updates to S^t: deselecting a module from the current module set $\mathcal{M}$ (affecting all selected layers) or deselecting a layer from the layer set $\mathcal{L}$ (affecting all selected modules). The update that maximizes the score in Equation (1) is applied to S^{t+1}. The process continues until one of two conditions is met: (1) further iterations will not yield a better strategy (controlled by stopping patience T); or (2) no additional layers or modules can be removed, *i.e.,* $|\mathcal{M}| = |\mathcal{L}| = 1$. Finally, the strategy S that yields a purified model M_{pure} with the highest score is returned. The complexity of our algorithm is a practical quadratic $O(|\mathcal{M}|^2 + |\mathcal{L}|^2)$.

5 Experiments

5.1 Experimental settings

Datasets. We evaluate our method on four datasets: **SST-2** (Socher et al., 2013), **OLID** (Zampieri et al., 2019), **MNLI** (Williams et al., 2018), and **AGNEWS** (Zhang et al., 2015). These datasets cover text classification and natural language inference (NLI) tasks, including binary and multi-class classification scenarios, and are widely used for evaluating text backdoor attacks and defenses (Qi et al., 2021a; He et al., 2023; Gupta and Krishna, 2023; Arora et al., 2024). We adapt the open-source datasets provided by HuggingFace (Lhoest et al., 2021). Table 1 are the statistics of these datasets.

Backdoor Attacks. We study defenses against four prominent types of *data-poisoning* backdoor attacks: (1) **BadNets** (Kurita et al., 2020), (2) **InsertSent** (Dai et al., 2019), (3) Learnable Word Substitution (**LWS**) (Qi et al., 2021c), and (4) **HiddenKiller** (Qi et al., 2021b). The first two correspond to insertion-based methods, while the latter two involve synonym substitution and syntactic paraphrasing approaches, respectively. A more recent attack, **BITE** (Yan et al., 2023), is discussed separately in Appendix B.3 due to its atypical behavior on benign models.

Dataset	Classes	Train	Test		Target Class
			Clean	Poison	
SST-2	2	67,349	872	444	Negative (0)
OLID	2	13,240	860	240	Not offensive (1)
MNLI	3	100,000	400	285	Neutral (1)
AGNews	4	120,000	7,600	5,700	Sports (1)

Table 1: The statistics of our evaluated datasets.

Follow Arora et al. (2024), we use rare words {*"cf", "mn", "bb", "tq", "mb"* } as triggers for **BadNets** and phrases {*"I watched this movie", "no cross, no crown"* } for **InsertSent**. The poison target labels for each dataset are listed in Table 1. While our main experiments focus on 20% poison rate (in line with the literature (Dai et al., 2019; Qi et al., 2021b; Arora et al., 2024)), *i.e.,* 20% of training samples are poisoned, we also evaluate lower poison rates of 10%, 5% in Appendix B.4.

Defense Baselines. We choose *seven* renowned defenses as main baselines, including *two* data-wise detection methods: (1) **ONION** (Qi et al., 2021a) and (2) **Z-Def.** (He et al., 2023), and *four* model-wise purification methods: (3) **PURE** (Zhao et al., 2024b), (4) **ABL** (Li et al., 2021b), (5) **TIES** (Yadav et al., 2023), (6) **DARE** (Yu et al., 2024), and (7) **WAG** (Arora et al., 2024).

Among these, ONION and Z-Def. detect and remove triggers from datasets. ONION identifies outlier words (potential triggers) using language models, *e.g.,* GPT-2 (Radford et al., 2019), while Z-Def. detects spurious correlations between tokens and labels. In contrast, (2)-(7) are model purification methods. PURE purifies the victim model using attention head pruning and normalization techniques, and ABL introduces a robust anti-backdoor training framework that first isolates, then unlearns backdoor associations. TIES, DARE and WAG are mode-merging baselines. We also report three additional baselines in Appendix B.2: two model purification methods, **Fine-mixing** (Zhang et al., 2022) and **Fine-purifying** (Zhang et al., 2023), and one data filtering method, **SEEP** (He et al., 2024), followed by retraining.

Evaluation Metrics. Following previous literature (Qi et al., 2021a), we adopt Clean Accuracy (**CACC**) and Attack Success Rate (**ASR**) to measure utility and defense performance respectively. CACC is evaluated on a clean test set, where both the samples and labels are ground truth. In contrast, ASR is evaluated on a poisoned test set, where triggers are implanted into each sample, and the labels

Dataset	Method	BadNet		InsertSent		LWS		HiddenKiller	
		ASR↓	CACC↑	ASR↓	CACC↑	ASR↓	CACC↑	ASR↓	CACC↑
SST-2	Benign	4.1	95.9	2.2	95.9	12.8	95.9	16.5	95.9
	Victim	100.0	96.0	100.0	96.3	98.0	95.4	96.5	95.7
	Proxy Model (IMDB)	7.4	89.1	4.3	89.1	10.8	89.1	13.7	89.1
	ONION	56.8	92.9	99.9	93.3	85.7	91.9	92.9	92.9
	Z-Def.	**4.6**	**96.1**	**1.8**	95.6	97.3	95.3	**35.7**	95.4
	PURE	0.0	50.9	0.0	50.9	0	50.9	0.0	50.9
	ABL	75.0	49.2	51.7	50.8	32.3	50.3	92.9	47.4
	TIES	99.9	95.7	100.0	95.8	93.5	95.2	88.8	95.3
	DARE w/ TIES	99.3	96.0	100.0	**96.2**	96.4	**95.7**	92.7	**95.8**
	WAG	84.4	94.8	60.1	95.2	**58.8**	94.8	56.2	92.5
	Ours (GMS)	**4.5**	91.6	**1.9**	92.5	**9.7**	91.7	**10.4**	91.2
AGNews	Benign	1.9	95.4	0.5	95.4	0.5	95.4	1.1	95.4
	Victim	99.9	95.1	99.6	95.3	99.6	94.5	100.0	95.1
	Proxy Model (BBCNews)	1.5	70.2	1.7	70.2	1.8	70.2	3.4	70.2
	ONION	59.4	94.8	97.8	95.1	84.8	94.5	99.6	94.7
	Z-Def.	**1.6**	**95.3**	**0.4**	95.3	97.9	**96.1**	100.0	95.0
	PURE	2.8	86.3	3.0	85.4	**6.5**	85.0	**9.4**	84.6
	ABL	50.2	55.0	48.8	54.8	100.0	55.0	90.2	54.9
	TIES	99.9	94.6	99.6	94.4	97.7	95.8	100.0	94.4
	DARE w/ TIES	99.9	95.2	99.6	**95.4**	97.8	96.5	100.0	**95.2**
	WAG	92.7	94.1	97.8	94.4	78.0	93.9	90.9	94.3
	Ours (GMS)	**2.5**	91.0	**2.4**	92.6	**3.2**	91.7	**6.5**	90.4

Table 2: (***Partial***) Performance of our method on two datasets compared to baselines under various backdoor attacks on the RoBERTa-large model, with each value averaged over three seeds. We highlight the top-2 lowest ASR results in **blue** cells, and the highest CACC results in **bold**. Results for other datasets are in Table 8.

are flipped to the target class.

Implementation details. Following Arora et al. (2024), we compare all methods on *RoBERTa-large* (Liu et al., 2019), *BERT-base-uncased* (Devlin et al., 2019), as well as LLMs including *Llama 2 7B* (Touvron et al., 2023), *Mistral 7B* (Jiang et al., 2023), and *Qwen 2.5 7B* (Team, 2024), with low-rank adaptation LoRA (Hu et al., 2022). All victim models are fine-tuned on poisoned datasets using the Adam optimizer with no weight decay (Kingma and Ba, 2015) and a learning rate of 2×10^{-5}. All defense baselines are implemented based on their open-source repositories (see Appendix A.3). The high-confidence proxy datasets ($\mathcal{D}_{clean}$ and $\mathcal{D}_{poison}$) are extracted following Section 4.2.2. To train proxy models for selection, we use **IMDB** (Zhang et al., 2015) for the **SST-2** victim model, **Twitter Abusive** (Founta et al., 2018) for the **OLID**, **SNLI** (Young et al., 2014) for **MNLI**, and **BBCNews** (Greene and Cunningham, 2006) for **AGNews**.

Our method uses hyperparameter α in Equation (1) to control the trade-off between backdoor defense and task utility. By default, α is set to 0.4 but can be adjusted based on priorities: smaller values (*e.g.,* 0.1) for lower ASR and larger values (*e.g.,* 1.0) for higher utility. The stopping patience $T = 5$ is default in Algorithm 1. In Appendix B.5, we justify these choices and show that *our method is NOT particularly sensitive to hyperparameters*.

5.2 Main results

Table 2 presents the performance of our method compared to all baselines under benchmark backdoor attacks on the *RoBERTa-large* model, with scores averaged over three runs using different seeds. Results on other datasets and architectures are provided in Appendix B.1. Across all datasets and backdoor attacks, our method consistently ranks among the top 2, regarding backdoor removal performance, with minor harm to clean accuracy. In particular, ***under the two particularly challenging backdoor tasks, LWS and HiddenKiller, GMS demonstrates significant improvements (i.e., at least 25%) over all baselines***–for example, for SST-2, our method reduces the ASR on LWS to 9.7%, compared to 58.8% for the following baseline Z-Def. While Z-Def achieves competitive performance with GMS in defending against BadNet and InsertSent, Z-Def is much less effective against LWS on both datasets. We attribute this to Z-Def.'s reliance on lexical and syntactic features to detect outliers in the poisoned dataset, whereas LWS attacks subtly replace words

with synonyms, effectively bypassing outlier detection. As for PURE, another competitive recent approach, we found its head-pruning step is highly unstable across both tasks and architectures. For example, it performs well on the BERT-base model in Table 9 (following the settings reported in **PURE** (Zhao et al., 2024b)), but performs poorly on most tasks with RoBERTa-large. Unlearning-based methods (**ABL** (Li et al., 2021b)) exhibit similar issues–often degrading CACC more than reducing ASR. Such instability has also been noted in prior works (Liu et al., 2018; Wu et al., 2022), which shows that these methods are sensitive to both hyperparameter choices and the quality of the proxy data. Additionally, we observed that all model-merging baselines suffer in defense performance when merging with a single proxy model, aligning with the results in Arora et al. (2024), and tend to prioritize preserving accuracy over removing backdoors. We discuss the potential reasons for this and the instability of PURE in Appendix B.1. We also compare our approach with two earlier model purification baselines, Fine-mixing (Zhang et al., 2022) and Fine-purifying (Zhang et al., 2023) in Appendix B.2.

Performance on LLMs. Given the growing popularity of LLMs, we evaluate the performance of our method on Llama-2-7B, Mistral-7B and Qwen-2.5-7B for the SST-2 dataset (Table 4). Remarkably, in all cases, GMS effectively removes backdoors.

Proxy Model Backdoor	CACC↑	ASR$_{Victim}$ ↓
Hidden Killer	95.4	**4.5**
BadNet (Diff. Trigger)	87.8	**6.9**
BadNet (Same Trigger)	95.8	100.0
Victim Model (BadNet)	95.6	100.0

Table 3: Proxy models with implanted backdoors can still effectively purify victim models if the backdoor is *not identical in both attack strategy and trigger*.

5.3 Ablation studies

We next examine the key property of GMS–robustness across diverse proxy datasets. For completeness, we defer additional results to the appendix: *stability* under different hyperparameter settings (Appendix B.5), *transferability* across attacks (Appendix B.12), preservation of clean victim model utility (Appendix B.6), and resilience to re-tuning attacks (Appendix B.10).

Proxy-model robustness: GMS retains the robustness property of model-merging defenses: it is resilient to both the specific choice of the proxy model and the benign or malicious nature of the proxy model; *i.e.,* as long as the backdoors in the two models are not identical, the backdoor can be effectively removed. We examine the sensitivity of GMS to the selection of proxy models in two scenarios. ***Scenario 1: Different proxies.*** We evaluated the performance of using different proxy models trained on homologous datasets, such as *IMDB*, *Yelp*, and *Amazon* (Zhang et al., 2015), to purify a victim model trained on *SST-2*. As shown in Table 5, our method is largely insensitive to the choice of proxy model across all backdoor attacks. Using any proxy model, GMS effectively defends against all tested attacks. ***Scenario 2: Backdoored proxies.*** We tested cases where the victim model and the proxy model were trained on the same dataset (SST2) with different backdoor attacks (BadNet and Hidden Killer) or on different datasets (SST2 and IMDB) with the same attacks (BadNet). From Table 3 and Table 17, we observed that ***as long as the backdoor in proxy model is not identical to that in victim model (i.e., the same backdoor strategy with an identical backdoor trigger), GMS consistently mitigates the backdoor***. We attribute this to the combination of module-level substitution and disruptive nature of model merging, which breaks the backdoor pathway such that a non-identical proxy cannot restore it.

Proxy-data robustness: Prior pruning- and unlearning-based methods strictly rely on access to clean data or known backdoor triggers (Min et al., 2024; Li et al., 2021b; Chen et al., 2022; Li et al., 2023), typically extracted from the training subset. In contrast, GMS operates under more practical settings, *e.g.,* post-processing after suboptimal data detections, where the data received is a mixture of clean and poisoned samples. Our previous experiments used $\mathcal{D}_{clean}$ from random sampling (containing 20% poisoned data) and $\mathcal{D}_{poison}$ from heuristic outlier detection. Table 7 shows that even when 70% of the "clean" proxy dataset is poisoned, GMS is still effective in purification. In Appendix C, we mathematically justify this and discuss the analytical constraints on proxy datasets. These results demonstrate that our method is highly robust to proxy-data construction. Nevertheless, Table 6 shows that more accurate proxy datasets–such as an oracle clean or label-flipped poison dataset–can fur-

Model	Method	BadNet		InsertSent		LWS		HiddenKiller	
		ASR↓	CACC↑	ASR↓	CACC↑	ASR↓	CACC↑	ASR↓	CACC↑
Llama-2	Benign	3.6	96.7	3.5	96.7	14.9	96.7	15.2	96.7
	Victim	100.0	97.0	100.0	97.1	98.7	96.1	95.9	96.7
	GMS	**3.5**	91.7	**3.6**	93.4	**12.2**	91.5	**7.6**	91.1
Mistral	Benign	3.8	97.1	4.5	97.1	7.7	97.1	12.3	97.1
	Victim	100.0	95.8	100.0	96.5	98.9	95.5	95.9	96.4
	GMS	**4.4**	92.1	**7.1**	92.4	**3.8**	91.9	**5.3**	91.9
Qwen-2.5	Benign	5.0	97.0	2.7	97.0	15.1	97.0	15.7	97.0
	Victim	100.0	96.3	100.0	92.9	98.5	98.5	96.0	96.2
	GMS	**6.3**	94.5	**7.5**	94.4	**16.4**	91.7	**15.0**	93.0

Table 4: Performance of our method on *Llama-2-7b*, *Mistral-7b* and *Qwen-2.5-7b* on **SST-2** dataset, averaged over three seeds.

Victim Model	Proxy Model	BadNet		InsertSent		LWS		HiddenKiller	
		ASR ↓	CACC ↑	ASR ↓	CACC ↑	ASR ↓	CACC ↑	ASR ↓	CACC ↑
SST-2	IMDB	5.4	92.8	1.6	94.0	12.6	93.8	11.0	93.0
	YELP	4.3	96.8	8.1	94.8	23.2	92.3	19.1	93.8
	AMAZON	3.8	92.9	1.1	95.8	20.1	92.6	13.5	93.8

Table 5: Proxy models trained on different datasets can all effectively purify victim models (trained on the SST-2).

Sub-proxy datasets		GMS	
Proxy Clean	Proxy Poison	CACC ↑	ASR ↓
Random Sample	Outlier Detection	93.6	**3.2**
Random Sample	Oracle	94.9	**4.1**
Oracle	Outlier Detection	92.7	**4.3**
Oracle	Oracle	95.6	**4.3**
Benign SST-2		95.6	4.1

Table 6: Performance comparison of different sub-proxy dataset configurations on GMS.

Ratio ρ	SST-2		AGNews	
	CACC ↑	ASR ↓	CACC ↑	ASR ↓
0.00	93.58	3.38	91.76	2.42
0.30	91.51	3.15	91.59	3.61
0.60	89.11	5.18	88.67	1.42
0.70	89.11	5.18	88.67	1.42
0.80	89.11	5.18	94.59	**99.75**
0.90	89.11	5.18	94.83	**99.82**

Table 7: Purification performance as the ratio ρ of poisoned data in extracted $\mathcal{D}_{clean}$ increases.

ther enhance performance, enabling precise module identification and full utility preservation (*e.g.,* CACC of 95.6%). However, even without optimal proxies, our method consistently removes backdoors, achieving benign-level ASR (4.1%) with comparable CACC.

6 Conclusion

We propose Guided-Module-Substitution (GMS), a model purification defense against data-poisoning attacks in NLP based on guided merging of the victim model with a single (existing) proxy model. Extensive experiments on both standard NLP models and LLMs validate its effectiveness, particularly against challenging attacks like LWS and HiddenKiller. Beyond strong empirical results, we validated desirable properties of GMS: robustness to proxy selection, tolerance to imperfect data, stability across hyperparameters, and transferability across attacks. These findings suggest that guided single-proxy merging offers a practical alternative to retraining-based defenses.

Limitations

Our method advances model-merging-based defenses by guiding the merging process with a trade-off signal between task utility and backdoor risk, reducing the requirement from a number of proxy models to just one. One potential limitation is that, like prior proxy-model-based defenses, we focus on the homogeneous setting and do not explore heterogeneous proxies. We make this choice for two reasons. First, in the defense scenario, merging serves solely as a tool for purification rather than an end in itself, so defender can control which proxy model to use. In practice, obtaining a homogeneous proxy is not difficult, making the heterogeneous case less relevant for our scenarios. Second, our goal is to introduce and validate the concept of guided merging for backdoor defense, rather than to develop state-of-the-art merging techniques. Since

knowledge transfer across heterogeneous models itself remains an open challenge (Xu et al., 2025; Du et al., 2025), we leave such extensions to future work. Moreover, while our method tolerates the use of a malicious proxy model, as long as its backdoor is not identical to the victim's (*i.e.,* the same attack method with the exact same trigger, which is rare in practice), as shown in Table 17, further relaxing this, such as enabling defenses against identically backdoored proxies or developing proxy-free approaches, could lead to even more robust and powerful defense strategies.

Ethical statement

Our method introduces a retraining-free purification approach for backdoored models, contributing to mitigating security risks in NLP systems from backdoor attacks. Additionally, our defense approach reduces the need for reannotating datasets to ensure they are purely clean, minimizing the potential harm of exposing annotators to harmfully poisoned contents. While we do not anticipate any direct negative consequences from this work, we hope it inspires further advancements in the development of robust, retraining-free defense methods for more realistic scenarios in future research.

Acknowledgements

We acknowledge the support from 2024 FSE Strategic Startup and the credits awarded from Google Cloud Platform.

References

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. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 15059–15075, Bangkok, Thailand. Association for Computational Linguistics.

Yang Bai, Gaojie Xing, Hongyan Wu, Zhihong Rao, Chuan Ma, Shiping Wang, Xiaolei Liu, Yimin Zhou, Jiajia Tang, Kaijun Huang, and Jiale Kang. 2025. Backdoor attack and defense on deep learning: A survey. *IEEE Transactions on Computational Social Systems*, 12(1):404–434.

Nicholas Carlini and Andreas Terzis. 2022. Poisoning and backdooring contrastive learning. In *International Conference on Learning Representations*.

Chen Chen, Yuchen Sun, Xueluan Gong, Jiaxin Gao, and Kwok-Yan Lam. 2024a. Neutralizing backdoors through information conflicts for large language models. *CoRR*, abs/2411.18280.

Weixin Chen, Baoyuan Wu, and Haoqian Wang. 2022. Effective backdoor defense by exploiting sensitivity of poisoned samples. In *Advances in Neural Information Processing Systems*, volume 35, pages 9727–9737. Curran Associates, Inc.

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 *Proceedings of the 37th Annual Computer Security Applications Conference*, ACSAC '21, page 554–569, New York, NY, USA. Association for Computing Machinery.

Xiaoyi Chen, Siyuan Tang, Rui Zhu, Shijun Yan, Lei Jin, Zihao Wang, Liya Su, Zhikun Zhang, XiaoFeng Wang, and Haixu Tang. 2024b. The janus interface: How fine-tuning in large language models amplifies the privacy risks. CCS '24, page 1285–1299, New York, NY, USA. Association for Computing Machinery.

Runxi Cheng, Feng Xiong, Yongxian Wei, Wanyun Zhu, and Chun Yuan. 2025. Whoever started the interference should end it: Guiding data-free model merging via task vectors. *CoRR*, abs/2503.08099.

Jiazhu Dai, Chuanshuai Chen, and Yufeng Li. 2019. A backdoor attack against lstm-based text classification systems. *IEEE Access*, 7:138872–138878.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Yiyang Du, Xiaochen Wang, Chi Chen, Jiabo Ye, Yiru Wang, Peng Li, Ming Yan, Ji Zhang, Fei Huang, Zhifang Sui, Maosong Sun, and Yang Liu. 2025. Adamms: Model merging for heterogeneous multimodal large language models with unsupervised coefficient optimization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9413–9422. Computer Vision Foundation / IEEE.

Antigoni Founta, Constantinos Djouvas, Despoina Chatzakou, Ilias Leontiadis, Jeremy Blackburn, Gianluca Stringhini, Athena Vakali, Michael Sirivianos, and Nicolas Kourtellis. 2018. Large scale crowdsourcing and characterization of twitter abusive behavior. *Proceedings of the International AAAI Conference on Web and Social Media*, 12(1).

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 *Proceedings of the 35th Annual Computer*

Security Applications Conference, ACSAC '19, page 113–125, New York, NY, USA. Association for Computing Machinery.

Micah Goldblum, Dimitris Tsipras, Chulin Xie, Xinyun Chen, Avi Schwarzschild, Dawn Song, Aleksander Madry, Bo Li, and Tom Goldstein. 2023. Dataset security for machine learning: Data poisoning, backdoor attacks, and defenses. *IEEE transactions on pattern analysis and machine intelligence*, 45(2):1563—1580.

Derek Greene and Pádraig Cunningham. 2006. Practical solutions to the problem of diagonal dominance in kernel document clustering. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, page 377–384, New York, NY, USA. Association for Computing Machinery.

Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. 2017. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *CoRR*, abs/1708.06733.

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. In *The Eleventh International Conference on Learning Representations*.

Ashim Gupta and Amrith Krishna. 2023. Adversarial clean label backdoor attacks and defenses on text classification systems. In *Proceedings of the 8th Workshop on Representation Learning for NLP (RepL4NLP 2023)*, pages 1–12, Toronto, Canada. Association for Computational Linguistics.

Xuanli He, Qiongkai Xu, Jun Wang, Benjamin Rubinstein, and Trevor Cohn. 2023. Mitigating backdoor poisoning attacks through the lens of spurious correlation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 953–967, Singapore. Association for Computational Linguistics.

Xuanli He, Qiongkai Xu, Jun Wang, Benjamin I. P. Rubinstein, and Trevor Cohn. 2024. SEEP: Training dynamics grounds latent representation search for mitigating backdoor poisoning attacks. *Transactions of the Association for Computational Linguistics*, 12:996–1010.

Yifei He, Siqi Zeng, Yuzheng Hu, Rui Yang, Tong Zhang, and Han Zhao. 2025. Mergebench: A benchmark for merging domain-specialized llms. *CoRR*, abs/2505.10833.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Kunzhe Huang, Yiming Li, Baoyuan Wu, Zhan Qin, and Kui Ren. 2022. Backdoor defense via decoupling the training process. In *International Conference on Learning Representations*.

Gabriel Ilharco, Marco Túlio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2023. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*. OpenReview.net.

Najeeb Moharram Jebreel, Josep Domingo-Ferrer, and Yiming Li. 2023. Defending against backdoor attacks by layer-wise feature analysis. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 428–440. Springer.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Keita Kurita, Paul Michel, and Graham Neubig. 2020. Weight poisoning attacks on pretrained models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2793–2806, Online. Association for Computational Linguistics.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander Rush, and Thomas Wolf. 2021. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Linyang Li, Demin Song, Xiaonan Li, Jiehang Zeng, Ruotian Ma, and Xipeng Qiu. 2021a. Backdoor attacks on pre-trained models by layerwise weight poisoning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*,

pages 3023–3032, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Shen Li, Liuyi Yao, Lan Zhang, and Yaliang Li. 2025. Safety layers in aligned large language models: The key to LLM security. In *The Thirteenth International Conference on Learning Representations*.

Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. 2021b. Anti-backdoor learning: Training clean models on poisoned data. In *Advances in Neural Information Processing Systems*, volume 34, pages 14900–14912. Curran Associates, Inc.

Yige Li, Xixiang Lyu, Xingjun Ma, Nodens Koren, Lingjuan Lyu, Bo Li, and Yu-Gang Jiang. 2023. Reconstructive neuron pruning for backdoor defense. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 19837–19854. PMLR.

Yuezun Li, Yiming Li, Baoyuan Wu, Longkang Li, Ran He, and Siwei Lyu. 2021c. Invisible backdoor attack with sample-specific triggers. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16443–16452.

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*, pages 273–294. Springer.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Rui Min, Zeyu Qin, Li Shen, and Minhao Cheng. 2023. Towards stable backdoor purification through feature shift tuning. In *Advances in Neural Information Processing Systems*, volume 36, pages 75286–75306. Curran Associates, Inc.

Rui Min, Zeyu Qin, Nevin L. Zhang, Li Shen, and Minhao Cheng. 2024. Uncovering, explaining, and mitigating the superficial safety of backdoor defense. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Tuan Anh Nguyen and Anh Tuan Tran. 2021. Wanet - imperceptible warping-based backdoor attack. In *International Conference on Learning Representations*.

Yang Ouyang, Hengrui Gu, Shuhang Lin, Wenyue Hua, Jie Peng, Bhavya Kailkhura, Meijun Gao, Tianlong Chen, and Kaixiong Zhou. 2025. Layer-level self-exposure and patch: Affirmative token mitigation for jailbreak attack defense. *arXiv preprint arXiv:2501.02629*.

Xudong Pan, Mi Zhang, Beina Sheng, Jiaming Zhu, and Min Yang. 2022. Hidden trigger backdoor attack on NLP models via linguistic style manipulation. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 3611–3628, Boston, MA. USENIX Association.

Fanchao Qi, Yangyi Chen, Mukai Li, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2021a. ONION: A simple and effective defense against textual backdoor attacks. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9558–9566, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Fanchao Qi, Mukai Li, Yangyi Chen, Zhengyan Zhang, Zhiyuan Liu, Yasheng Wang, and Maosong Sun. 2021b. Hidden killer: Invisible textual backdoor attacks with syntactic trigger. In *Annual Meeting of the Association for Computational Linguistics*.

Fanchao Qi, Yuan Yao, Sophia Xu, Zhiyuan Liu, and Maosong Sun. 2021c. Turn the combination lock: Learnable textual backdoor attacks via word substitution. In *Annual Meeting of the Association for Computational Linguistics*.

Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. 2024. Fine-tuning aligned language models compromises safety, even when users do not intend to! In *The Twelfth International Conference on Learning Representations*.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Weisong Sun, Yuchen Chen, Chunrong Fang, Yebo Feng, Yuan Xiao, An Guo, Quanjun Zhang, Zhenyu Chen, Baowen Xu, and Yang Liu. 2025. Eliminating backdoors in neural code models for secure code understanding. *Proc. ACM Softw. Eng.*, 2(FSE).

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. 2020. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP*

2020, Online, November 16-20, 2020, pages 9275–9293. Association for Computational Linguistics.

Ayush K Tarun, Vikram S Chundawat, Murari Mandal, and Mohan Kankanhalli. 2024. Fast yet effective machine unlearning. *IEEE transactions on neural networks and learning systems*, 35(9):13046—13055.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Brandon Tran, Jerry Li, and Aleksander Madry. 2018. Spectral signatures in backdoor attacks. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Alexander Turner, Dimitris Tsipras, and Aleksander Madry. 2019. Label-consistent backdoor attacks. *CoRR*, abs/1912.02771.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

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 *2019 IEEE Symposium on Security and Privacy (SP)*, pages 707–723.

Lijin Wang, Jingjing Wang, Tianshuo Cong, Xinlei He, Zhan Qin, and Xinyi Huang. 2025a. From purity to peril: Backdooring merged models from "harmless" benign components. In *USENIX Security Symposium (USENIX Security)*.

Yuhang Wang, Huafeng Shi, Rui Min, Ruijia Wu, Siyuan Liang, Yichao Wu, Ding Liang, and Aishan Liu. 2022. Universal backdoor attacks detection via adaptive adversarial probe. *arXiv preprint arXiv:2209.05244*.

Zhenting Wang, Kai Mei, Juan Zhai, and Shiqing Ma. 2023. UNICORN: A unified backdoor trigger inversion framework. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Zijing Wang, Xingle Xu, Yongkang Liu, Yiqun Zhang, Peiqin Lin, Shi Feng, Xiaocui Yang, Daling Wang, and Hinrich Schütze. 2025b. Why do more experts fail? A theoretical analysis of model merging. *CoRR*, abs/2505.21226.

Boyi Wei, Kaixuan Huang, Yangsibo Huang, Tinghao Xie, Xiangyu Qi, Mengzhou Xia, Prateek Mittal, Mengdi Wang, and Peter Henderson. 2024. Assessing the brittleness of safety alignment via pruning and low-rank modifications. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak Gadre, Rebecca Roelofs, Raphael Gontijo Lopes, Ari S. Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. 2022. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 23965–23998. PMLR.

Baoyuan Wu, Hongrui Chen, Mingda Zhang, Zihao Zhu, Shaokui Wei, Danni Yuan, and Chao Shen. 2022. Backdoorbench: A comprehensive benchmark of backdoor learning. In *Advances in Neural Information Processing Systems*, volume 35, pages 10546–10559. Curran Associates, Inc.

Dongxian Wu and Yisen Wang. 2021. Adversarial neuron pruning purifies backdoored deep models. In *Advances in Neural Information Processing Systems*, volume 34, pages 16913–16925. Curran Associates, Inc.

Chang Xu, Jun Wang, Yuqing Tang, Francisco Guzmán, Benjamin I. P. Rubinstein, and Trevor Cohn. 2021. A targeted attack on black-box neural machine translation with parallel data poisoning. In *Proceedings of the Web Conference 2021*, WWW '21, page 3638–3650, New York, NY, USA. Association for Computing Machinery.

Xiong Xu, Kunzhe Huang, Yiming Li, Zhan Qin, and Kui Ren. 2024. Towards reliable and efficient backdoor trigger inversion via decoupling benign features. In *The Twelfth International Conference on Learning Representations*.

Zhengqi Xu, Han Zheng, Jie Song, Li Sun, and Mingli Song. 2025. Training-free heterogeneous model merging. *CoRR*, abs/2501.00061.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. 2023. Ties-merging: Resolving interference when merging models. In *Advances in Neural Information Processing Systems*, volume 36, pages 7093–7115. Curran Associates, Inc.

Jun Yan, Vansh Gupta, and Xiang Ren. 2023. BITE: textual backdoor attacks with iterative trigger injection. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 12951–12968. Association for Computational Linguistics.

Wenkai Yang, Lei Li, Zhiyuan Zhang, Xuancheng Ren, Xu Sun, and Bin He. 2021a. Be careful about poisoned word embeddings: Exploring the vulnerability of the embedding layers in NLP models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2048–2058, Online. Association for Computational Linguistics.

Wenkai Yang, Yankai Lin, Peng Li, Jie Zhou, and Xu Sun. 2021b. RAP: Robustness-Aware Perturbations for defending against backdoor attacks on NLP models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8365–8381, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. 2024. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *Forty-first International Conference on Machine Learning*.

Zenghui Yuan, Yangming Xu, Jiawen Shi, Pan Zhou, and Lichao Sun. 2025. Merge hijacking: Backdoor attacks to model merging of large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, pages 32688–32703. Association for Computational Linguistics.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. Predicting the type and target of offensive posts in social media. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1415–1420, Minneapolis, Minnesota. Association for Computational Linguistics.

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 on ACM SIGSAC Conference on Computer and Communications Security, CCS 2024*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Xinyang Zhang, Zheng Zhang, Shouling Ji, and Ting Wang. 2021. Trojaning language models for fun and profit. In *2021 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 179–197.

Zhiyuan Zhang, Deli Chen, Hao Zhou, Fandong Meng, Jie Zhou, and Xu Sun. 2023. Diffusion theory as a scalpel: Detecting and purifying poisonous dimensions in pre-trained language models caused by backdoor or bias. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 2495–2517, Toronto, Canada. Association for Computational Linguistics.

Zhiyuan Zhang, Lingjuan Lyu, Xingjun Ma, Chenguang Wang, and Xu Sun. 2022. Fine-mixing: Mitigating backdoors in fine-tuned language models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 355–372, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Wei Zhao, Zhe Li, Yige Li, Ye Zhang, and Jun Sun. 2024a. Defending large language models against jailbreak attacks via layer-specific editing. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 5094–5109, Miami, Florida, USA. Association for Computational Linguistics.

Xingyi Zhao, Depeng Xu, and Shuhan Yuan. 2024b. Defense against backdoor attack on pre-trained language models via head pruning and attention normalization. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 61108–61120. PMLR.

Yuhang Zhou, Giannis Karamanolakis, Victor Soto, Anna Rumshisky, Mayank Kulkarni, Furong Huang, Wei Ai, and Jianhua Lu. 2025a. Mergeme: Model merging techniques for homogeneous and heterogeneous moes. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2025 - Volume 1: Long Papers, Albuquerque, New Mexico, USA, April 29 - May 4, 2025*, pages 2315–2328. Association for Computational Linguistics.

Zhenhong Zhou, Haiyang Yu, Xinghua Zhang, Rongwu Xu, Fei Huang, and Yongbin Li. 2024. How alignment and jailbreak work: Explain LLM safety through intermediate hidden states. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 2461–2488, Miami, Florida, USA. Association for Computational Linguistics.

Zhenhong Zhou, Haiyang Yu, Xinghua Zhang, Rongwu Xu, Fei Huang, Kun Wang, Yang Liu, Junfeng Fang, and Yongbin Li. 2025b. On the role of attention

heads in large language model safety. In *The Thirteenth International Conference on Learning Representations*.

Mingli Zhu, Shaokui Wei, Li Shen, Yanbo Fan, and Baoyuan Wu. 2023. Enhancing fine-tuning based backdoor defense with sharpness-aware minimization. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4443–4454.

A Experiment details

A.1 Experiment setup

We adopt the same training configuration as WAG (Arora et al., 2024) to ensure a fair comparison. This includes fine-tuning *RoBERTa-large*, *BERT-base-uncased*, as well as LLMs including *Llama 2 7B* (Touvron et al., 2023), *Mistral 7B* (Jiang et al., 2023), and *Qwen 2.5 7B* (Team, 2024) on poisoned datasets, using the Adam optimizer with no weight decay (Kingma and Ba, 2015) and a learning rate of 2×10^{-5}. For encoder models, the batch size, maximum sequence length, and epoch are set to 32, 128, and 3, respectively. For decoder LLMs, due to computational limits, we adapt low-rank adaptation (LoRA) (Hu et al., 2022) for all the linear modules within the Transformer layers and apply our method to substitute those LoRA modules after training. We train these LLMs for 2 epochs with a batch size of 8 and a maximum sequence length of 128.

To validate our defense, we first adapt the aforementioned backdoor methods and datasets to obtain victim models. Specifically, to train backdoored models, we construct poisoned datasets by injecting triggers into the training split, while the validation split is used to construct a clean and a poison test set respectively. For MNLI, we use a random subset of 100,000 samples to reduce overhead. Next, we train proxy models on homologous datasets to prepare for future merging. We then construct high-confidence proxy-clean ($\mathcal{D}_{clean}$) and proxy-poison ($\mathcal{D}_{poison}$) sets from the training data (using the methods illustrated in Section 4.2.2) to guide our parameter substitution strategy search.

For a model trained on unverified data, access to fully clean data is often not guaranteed. However, the corresponding task is typically known to the defender, providing an opportunity to identify homologous models trained on overlapping tasks that can be used for merging. For instance, previous works, WAG and PURE, utilized the **IMDB** (Maas et al., 2011) dataset to purify models trained on poisoned **SST-2** datasets, as both share the sentiment classification task domain.

In this work, we select proxy models that share a similar domain with the poisoned tasks. Specifically, we use **Twitter Abusive** (Founta et al., 2018) for **OLID**, **SNLI**[3] (Young et al., 2014) for **MNLI**, and **BBCNews** (Greene and Cunningham, 2006)

[3]Creative Commons Attribution-ShareAlike 4.0 International License

for **AGNews**. To evaluate generalizability, we use three sentiment classification datasets–**IMDB**, **YELP**, and **Amazon Reviews** (Zhang et al., 2015)– to purify victim **SST-2** models through model merging. All datasets are downloaded from Hugging Face, which adheres to the Apache License 2.0.

Our method involves only one hyperparameter, α, as described in Equation (1), which controls the preference between model utility and attack resistance during the model merging strategy search. We set α to 0.4 by default, which slightly favors seeking a more attack-resistant model. It can be adjusted to smaller values (*e.g.,* 0.1) when a lower ASR is the primary target, and to larger values (*e.g.,* 1.0) when high utility is demanded.

A.2 Computational Resources

We conduct experiments using three seeds on a single A100 GPU, and report the average scores. Running our method takes only 4 minutes on a single GPU for a 24-layer *RoBERTa-large* architecture.

A.3 Implementation details for baselines

We follow the open-source implementations for each baseline, and basically using their default hyper-parameters, while maintaining using the identical datasets, backdoor settings and the trained models for a fair comparison.

For the baseline **DARE** (Yu et al., 2024), it first applies random parameter dropping and rescaling to the involved models with a specified drop rate, and then incorporates model merging techniques to combine the processed models. Various model merging methods can be integrated with DARE, and we choose TIES merging as a representative, as it demonstrates decent performance for encoder-based models (*e.g., bert-base, roberta-base*). Their method involves three tunable hyperparameters for encoder-based LMs, as outlined in Table 5 of their paper: *drop rate*, *scaling term*, and *ratio to retain*. We retained the original search space for *drop rate* and *scaling term* but expanded the *ratio to retain* from [0.1, 0.2, 0.3] to [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8] for the *roberta-large* model to ensure more complete search.

For the baseline **PURE** (Zhao et al., 2024b), it first trains the victim model on a proxy *clean* dataset (from the same domain or a similar task, *e.g.,* poisoned on **IMDB** and fine-tuned on **SST-2**) and prunes the attention heads that cause attention drifting between poisoned and clean texts. The pruned model is then fine-tuned further on the proxy clean dataset to normalize the remaining attention heads for purification. To ensure a fair comparison in the main experiments with the *roberta-large* model, we use the same poisoned and proxy datasets, *e.g.,* poisoned on the SST-2 dataset with IMDB as the proxy clean dataset. We set the hyperparameter accuracy threshold (used to stop head pruning) to 0.90 for **SST-2**, **MNLI**, and **AGNews**, and 0.85 for **OLID** to prevent overly aggressive pruning. For the *bert-base* model, we follow the original implementation, including the use of the **SST-2** dataset for fine-tuning SST-2 victim models and maintaining the default accuracy threshold of 0.85. While PURE uses the *label flip rate* (**LFR**) as its evaluation metric for backdoor defense (implanting triggers into test data while keeping the labels unchanged), we adopt the attack success rate (**ASR**) on label-flipped test data as our evaluation metric for a fair comparison with our method.

We adhere to the default implementations and hyperparameter settings for all other baseline methods.

B Additional results

B.1 Performance across different model architectures and different dataset

Performance across different dataset. Due to space constraints, we present the complete performance results of our method compared to all baselines under various backdoor attacks on the *RoBERTa-large* model across all four datasets in Table 2. The settings are consistent with those in Section 5.2, with each score averaged over three runs using different seeds. For the OLID dataset, we specifically set $\alpha = 0.1$ to enable more aggressive defense, as shown in Figure 2. The top two ASR performances are highlighted. Across all datasets and backdoor attacks, our method consistently ranks among the top two in backdoor removal performance, with minimal harm to clean accuracy. Notably, under the two particularly challenging backdoor tasks, LWS and HiddenKiller, GMS achieves significant improvements (at least 25%) over all baselines. For instance, on SST-2, GMS reduces the ASR for LWS to 9.7%, compared to 58.8% for the next-best baseline, Z-Def.

While Z-Def achieves competitive performance with GMS in defending against BadNet and InsertSent, Z-Def is much less effective against LWS

Dataset	Method	BadNet		InsertSent		LWS		HiddenKiller	
		ASR↓	CACC↑	ASR↓	CACC↑	ASR↓	CACC↑	ASR↓	CACC↑
SST-2	Benign	4.1	95.9	2.2	95.9	12.8	95.9	16.5	95.9
	Victim	100.0	96.0	100.0	96.3	98.0	95.4	96.5	95.7
	Proxy Model (IMDB)	7.4	89.1	4.3	89.1	10.8	89.1	13.7	89.1
	ONION	56.8	92.9	99.9	93.3	85.7	91.9	92.9	92.9
	Z-Def.	**4.6**	**96.1**	**1.8**	95.6	97.3	95.3	**35.7**	95.4
	PURE	0	50.9	0	50.9	0	50.9	0	50.9
	TIES	99.9	95.7	100.0	95.8	93.5	95.2	88.8	95.3
	DARE w/ TIES	99.3	96.0	100.0	**96.2**	96.4	**95.7**	92.7	**95.8**
	WAG	84.4	94.8	60.1	95.2	**58.8**	94.8	56.2	92.5
	Ours (GMS)	**4.5**	91.6	**1.9**	92.5	**9.7**	91.7	**10.4**	91.2
OLID	Benign	29.0	84.9	31.0	84.9	45.6	84.9	53.6	84.9
	Victim	99.9	85.2	100.0	85.0	94.5	85.1	100.0	85.0
	Proxy Model (Twitter)	37.8	84.4	40.0	84.4	55.4	84.4	67.1	84.4
	ONION	75.0	84.8	99.4	84.8	86.1	84.1	99.6	84.7
	Z-Def.	**29.7**	85.6	**30.3**	85.3	93.5	85.3	**53.9**	85.7
	PURE	82.2	75.2	87.5	75.2	76.1	79.1	100.0	72.1
	TIES	36.2	84.8	38.2	84.8	**57.5**	84.5	65.3	84.5
	DARE w/ TIES	95.6	**86.3**	77.8	**86.0**	90.5	**85.9**	86.8	**86.3**
	WAG	52.0	85.0	48.6	84.6	63.1	84.5	68.2	85.0
	Ours (GMS)*	**32.1**	85.0	**28.0**	84.2	**52.2**	84.4	**64.2**	84.9
MNLI	Benign	12.3	87.6	12.6	87.6	26.4	87.6	36.9	87.6
	Victim	100.0	89.4	100.0	90.3	96.0	89.0	99.9	89.4
	Proxy Model (SNLI)	12.2	84.1	9.2	84.1	25.3	84.1	31.7	84.1
	ONION	64.3	86.1	98.6	86.9	89.0	85.5	98.8	86.6
	Z-Def.	**11.1**	88.3	**11.6**	89.7	92.2	89.1	**50.6**	89.7
	PURE	33.3	33.8	33.3	33.8	33.3	33.8	33.3	33.8
	TIES	91.6	89.2	94.5	89.5	80.7	89.2	88.6	90.0
	DARE w/ TIES	93.5	**90.5**	100.0	**91.4**	93.9	**89.9**	99.4	**90.5**
	WAG	71.0	88.7	60.3	88.5	**77.9**	88.0	80.5	88.8
	Ours (GMS)	**10.8**	86.5	**10.7**	86.3	**14.0**	86.5	**31.7**	86.3
AGNews	Benign	1.9	95.4	0.5	95.4	0.5	95.4	1.1	95.4
	Victim	99.9	95.1	99.6	95.3	99.6	94.5	100.0	95.1
	Proxy Model (BBCNews)	1.5	70.2	1.7	70.2	1.8	70.2	3.4	70.2
	ONION	59.4	94.8	97.8	95.1	84.8	94.5	99.6	94.7
	Z-Def.	**1.6**	**95.3**	**0.4**	95.3	97.9	**96.1**	100.0	95.0
	PURE	2.8	86.3	3.0	85.4	**6.5**	85.0	**9.4**	84.6
	TIES	99.9	94.6	99.6	94.4	97.7	95.8	100.0	94.4
	DARE w/ TIES	99.9	95.2	99.6	**95.4**	97.8	96.5	100.0	**95.2**
	WAG	92.7	94.1	97.8	94.4	78.0	93.9	90.9	94.3
	Ours (GMS)	**2.5**	91.0	**2.4**	92.6	**3.2**	91.7	**6.5**	90.4

Table 8: Performance of our method compared to baselines under various backdoor attacks on the RoBERTa-large model, with each value averaged over three seeds.

Dataset	Method	BadNet		InsertSent		LWS		HiddenKiller	
		ASR↓	CACC↑	ASR↓	CACC↑	ASR↓	CACC↑	ASR↓	CACC↑
SST-2	Benign	8.5	92.9	3.5	92.9	22.2	92.9	17.3	92.9
	Victim	100.0	92.9	100.0	92.2	98.1	91.5	95.9	91.8
	Proxy Model (IMDB)	10.4	85.4	6.4	85.4	17.9	85.4	14.7	85.4
	ONION	58.0	89.9	99.7	89.9	85.4	88.6	94.4	89.2
	Z-Def.	**8.3**	92.5	**1.8**	91.9	97.4	90.9	38.7	91.5
	PURE	30.1	92.0	10.7	91.5	**66.4**	91.1	**28.7**	91.3
	TIES	99.2	92.7	98.2	**92.4**	91.2	92.3	86.8	**92.6**
	DARE w/ TIES	100.0	**93.0**	90.8	88.9	92.6	**92.7**	94.1	90.6
	WAG	74.3	91.9	70.7	91.7	73.5	91.7	60.3	91.6
	Ours (GMS)	**10.8**	86.8	**9.8**	88.8	**27.4**	88.0	**19.5**	86.8

Table 9: Performance of our method compared to baselines under various backdoor attacks on the BERT-base model, with each value averaged over three seeds. The victim model is trained on SST-2 dataset.

on both datasets. This is because Z-Def relies on lexical and syntactic features to detect outliers in the poisoned dataset, whereas LWS attacks subtly replace words with synonyms, effectively bypassing outlier detection. As for PURE, another competitive recent approach, we found its head-pruning step is highly unstable across both tasks and architectures. For example, it performs well on the BERT-base model in Table 9 (following the settings reported in **PURE** (Zhao et al., 2024b)), but performs poorly on most tasks with RoBERTa-large. This instability seems to arise from PURE's reliance on accuracy from the clean proxy dataset as the pruning stopping criterion. For well-trained models with high accuracy, a substantial number of heads are pruned, leading to a broken purified model (*e.g.,* 0% ASR but random-guess-level CACC).

Additionally, although merging multiple models has been shown to be an effective backdoor defense, we observed that applying all merging baselines using a single proxy model is sub-optimal, aligning with the results in Arora et al. (2024). We found these methods prioritize preserving accuracy over removing backdoors. This behavior likely arises from the design of merging mechanisms, which are intended to preserve the performance of each merged model on downstream tasks. Consequently, backdoor tasks are treated equivalently to downstream tasks, exposing a persistent vulnerability.

Performance across different model architectures. We also evaluated our method on different architectures. Table 9 shows the results on the BERT-base model. The poison rate remains at 20%, and each model is trained for three epochs. Scores are reported under $\alpha = 0.4$ and averaged over three seeds: 1000, 2000, and 3000. Across all attacks, our method performs consistently well, particularly excelling against more challenging attack strategies. Notably, it achieves over a 39% advantage in defending against LWS and a 9% improvement over the second-best baseline for HiddenKiller.

B.2 Comparisons to other baseline defenses

Model purification baselines Fine-mixing (Zhang et al., 2022) and Fine-purification (Zhang et al., 2023) are two additional model purification baselines. We compare our method against their reported performance in their papers[4], strictly following their experimental settings. Specifically, we evaluate under the same conditions using two types of attacks (BadNet and InsertSent) and two model architectures (BERT-base and RoBERTa-base) on the AGNews dataset. As shown in Table 10, our method consistently achieves better defense performance across all settings. Notably, against the more advanced InsertSent attack, our approach reduces the attack success rate (ASR) by at least 17% in absolute values compared to Fine-mixing and Fine-purification.

Poison sample detection with retraining: SEEP (He et al., 2024) While this paper focuses on the important but often overlooked post-deployment setting, it is natural to ask: ***if we had access to a highly effective data detection method capable of identifying most poisoned samples, how would our method compare?*** To explore this, we evaluate against SEEP (He et al., 2024), a state-of-the-art poison detection method, on defending against BadNet attacks on the SST-2 dataset. As shown in Table 11, although SEEP performs well under this relatively simple attack, our method–despite relying only on a naive heuristic for proxy data selection–achieves even better performance. As further discussed in Table 6, incorporating more advanced prior knowledge (*e.g.,* by using stronger detection methods to construct sub-proxy datasets) can further enhance the effectiveness of our approach.

B.3 Comparisons of all defenses under a more recent attack: BITE (Yan et al., 2023)

BITE (Yan et al., 2023) is a recent insertion-based textual backdoor attack that leverages label-biased tokens as stealthy triggers. Table 12 shows the performance of our method and four baselines in defending against BITE on the SST-2 dataset. Our method consistently ranks among the top two in defense effectiveness, achieving comparable or even lower ASR than the benign model, while maintaining clean accuracy.

We exclude BITE from the main results in Section 5.2 due to a peculiar issue: BITE tends to yield an unusually high ASR even on benign models. This undermines the reliability of ASR as an evaluation metric in this setting and makes it difficult to confidently interpret defense performance. We attribute this to BITE's trigger selection process, which biases trigger tokens toward naturally label-correlated words, effectively exploiting model priors without requiring explicit poisoning.

[4]The code is not publicly available yet.

Architecture	Methods	BadNet		InsertSent	
		CACC ↑	ASR ↓	CACC ↑	ASR ↓
BERT-base	Fine-mixing*	90.17	12.32	90.40	32.37
	Fine-purifying*	90.86	3.3	91.13	23.69
	Ours	**91.8**	**1.8**	**86.49**	**7.00**
RoBERTa-base	Fine-mixing*	86.39	18.12	86.11	35.97
	Fine-purifying*	86.64	17.56	86.85	19.20
	Ours	**91.2**	**1.63**	**89.62**	**1.42**

Table 10: Performance comparison of our method against Fine-mixing (Zhang et al., 2022) and Fine-purifying (Zhang et al., 2023) on AGNews dataset. The best results are in **bold**. Results in rows marked with * are taken directly from the respective papers.

Architecture	Defense Method	CACC↑	ASR↓
RoBERTa-large	SEEP with retraining	95.5	*24.3*
	Ours	91.7	**9.7**
BERT-base	SEEP with retraining	92.4	29.4
	Ours	88.0	**27.4**

Table 11: Comparison of our method with retraining-based defenses on BERT-base and RoBERTa-large for SST-2 dataset.

B.4 Performance under different poison rate: 20%, 10% and 5%

While the experiments in the main paper are conducted with a 20% poison rate to ensure fair comparison with the baselines, following the settings in (Dai et al., 2019; Qi et al., 2021b; Arora et al., 2024), *i.e.,* 20% of the training samples are poisoned, we also evaluate lower poison rates of 10% and 5%, as shown in Table 13. Our method is effective in model purification across all poison rates.

B.5 Stability: How sensitive is our method to the selection of hyperparameters?

We next show that GMS exhibits stable performance across a wide range of hyperparameters, in contrast to pruning-based and unlearning-based methods, which have been shown to be more sensitive to hyperparameter choices (Liu et al., 2018; Wu et al., 2022) (as also observed in Table 2). Our method involves two hyperparameters: α in Equation (1) and stopping patience T, with default values of $\alpha = 0.4$ and $T = 5$ used in main results. Below, we justify these choices. As shown in Figure 2 and Figure 3, for most dataset and backdoor attack combinations, the default value $\alpha = 0.4$ strikes a balanced trade-off between utility and backdoor defense strength. An exception is the HiddenKiller attack on the OLID dataset, where a more aggressive $\alpha = 0.1$ is recommended. This is because OLID is a dataset collected from tweets, while HiddenKiller paraphrases the data using formal syntactic templates, drastically altering OLID's language style. This change significantly impacts CACC, necessitating a higher weight for backdoor removal to ensure effective purification. For stopping patience, as shown in Figure 4, the score computed in Equation (1) generally follows a (weakly) monotonically decreasing trend. While a larger T improves performance, we choose $T = 5$ as a reasonable balance between performance and efficiency.

B.6 If the victim model is benign, will our method affect its utility?

While all our previous experiments focus on malicious victim models, it is also important to consider scenarios where the victim model is benign. *Does our method compromise the utility of a clean victim model?* As shown in Table 14, our results effectively dispel this concern. When applying GMS to a benign model, the CACC remains unaffected.

B.7 Can our method be effective with a fully clean proxy dataset?

While we primarily consider the practical scenario of receiving a mixed dataset, *is our method still effective if the received dataset is entirely clean?* As shown in Table 15, when provided only with a clean proxy dataset—without any information about the poisoned dataset—GMS can still effectively remove backdoors. This is because our substitution strategy starts by fully removing backdoors and then gradually recovering task utility.

Model	Defense Method	BERT-base		RoBERTa-Large	
		CACC↑	ASR↓	CACC↑	ASR↓
Victim (BITE)	Benign Model	92.9	41.4	95.9	38.3
	No Defense	92.6	80.7	95.6	81.0
Defense	Z-Defense	92.3	51.9	95.2	44.9
	ONION	89.7	70.5	92.9	68.2
	ABL	92.0	82.1	**51.6**	**1.0**
	WAG	92.9	**47.5**	94.4	45.2
	Ours	88.9	**49.9**	90.3	**27.9**

Table 12: Defense performance against the BITE (Yan et al., 2023) attack on BERT-base and RoBERTa-large models for SST-2 dataset.

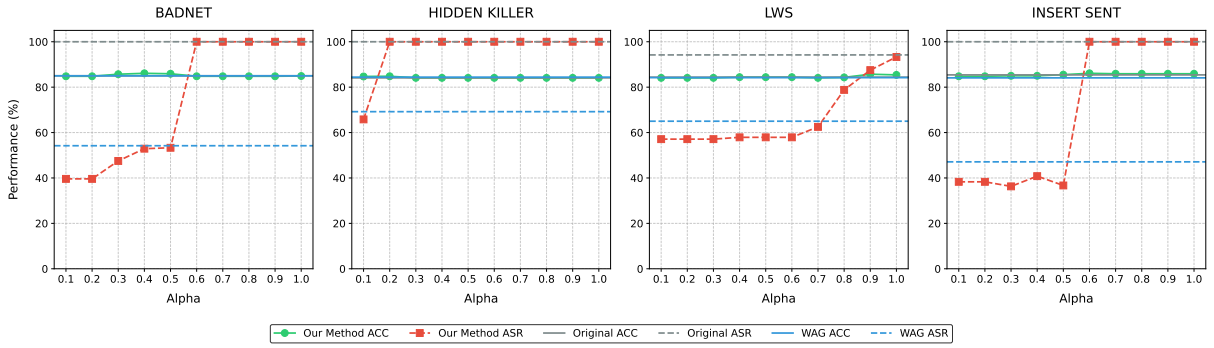

Figure 2: Results of using different weights (*alpha*) on OLID dataset.

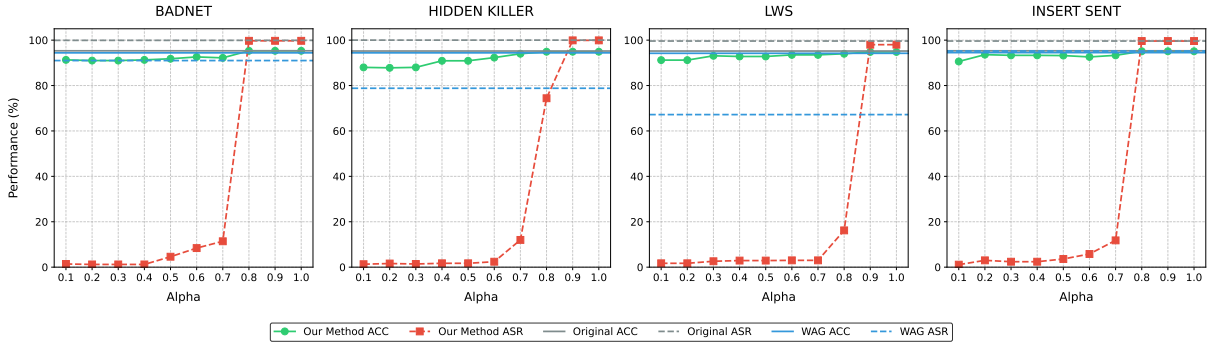

Figure 3: Results of using different weights (*alpha*) on AGNEWS dataset.

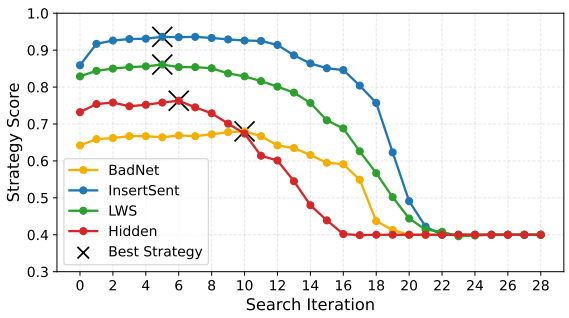

Figure 4: The search iteration history for four kinds of backdoors on **SST-2** dataset.

B.8 GMS remains effective as long as the proxy model is not backdoored with exactly the same attack and trigger.

An important question to consider is: What happens if the proxy model itself contains a backdoor?

Victim	Proxy	Poison Rate	Metric	Result
SST-2 (BadNet)	IMDB	20%	**ASR↓**	4.5
			CACC↑	91.6
		10%	**ASR↓**	4.5
			CACC↑	94.4
		5%	**ASR↓**	12.4
			CACC↑	91.9

Table 13: Results for SST-2 with varying poison rates.

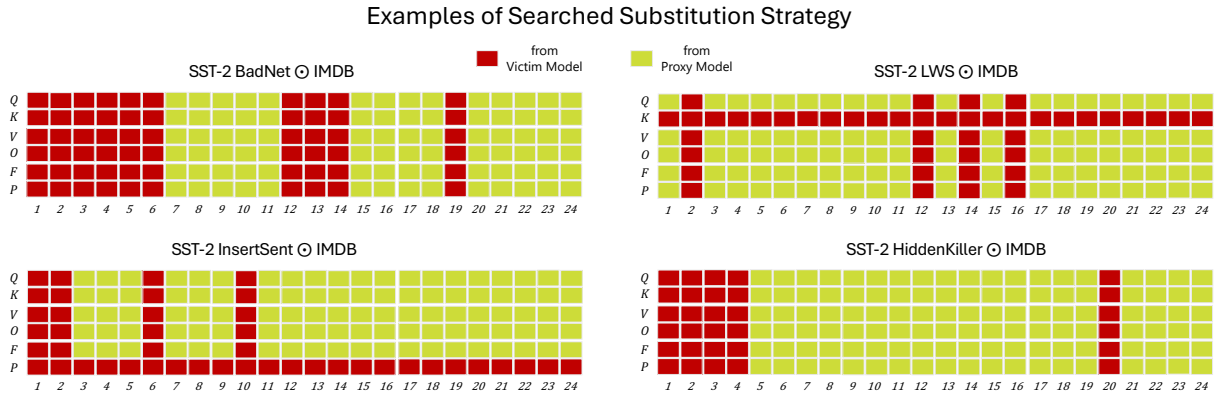

Figure 5: The optimal substitution strategy for defending against each backdoor attack for the *roberta-large* model trained on the SST-2 dataset. The green squares indicate the substituted modules.

To investigate this, we considered scenarios where the victim and proxy models were either trained on the same dataset but with different backdoor attacks or on different datasets with the same attack.

From Table 17, our findings indicate that GMS remains effective in backdoor removal as long as the proxy model's backdoor is not an exact match to that of the victim model, *i.e.,* both the same attack method and the same trigger.

B.9 Purification effectiveness as the poison ratio increases.

Due to space constraints, we included an incomplete version of Table 7 in the main paper. Here, the full ratio table Table 18 provides a clearer trend: (1) As the proportion of poisoned data in the proxy "clean" dataset increases, the CACC of the purified model slightly decreases. This suggests that under a default weight of $\alpha = 0.4$ for $\mathcal{D}_{clean}$, a higher level of poisoning may introduce confusion, leading GMS to mistakenly remove some clean task-critical modules due to their reduced relative weight (see Appendix C for the analytical explanation). (2) However, even with poisoned ratios as high as 90% in SST-2 and 70% in AGNews, GMS remains effective in eliminating backdoor-related components and purifying the model. This effectiveness is mathematically justified in Appendix C.

Dataset	Model	CACC ↑
SST-2	Benign Victim Model	95.9
	Proxy Model (IMDB)	88.7
	Purified Model (GMS)	95.8
AGNews	Benign Victim Model	95.2
	Proxy Model (BBCNews)	78.1
	Purified Model (GMS)	95.3

Table 14: CACC of benign victim models preserved.

Proxy Dataset	CACC↑	ASR↓
Pure clean proxy dataset	88.3	**4.7**
Mixed poisoned proxy dataset	91.6	**4.5**

Table 15: Performance of our method with a clean vs. mixed proxy dataset, for the *roberta-large* model trained on the SST-2 dataset.

B.10 Robustness to Retuning Attacks

Recent works (Min et al., 2024; Qi et al., 2024) have highlighted a critical pitfall in backdoor defenses: purified models with low ASR often fail to completely eliminate inserted backdoor features. A straightforward Retuning Attack (RA)–which involves fine-tuning a purified model on a small number of backdoored samples (*e.g.,* 0.2% of the original poisoned data) mixed with benign samples (to maintain clean Accuracy on the main task)–can easily recover the backdoor in the purified model, as demonstrated in Table 16.

Several related studies (Qi et al., 2024; Chen et al., 2024b; Tarun et al., 2024) have sought to uncover the underlying reasons for this failure. A key observation is that the purified model's parameters do not sufficiently deviate from those of the backdoored model. Specifically, the weight differences between a purified model and a (retrained) benign model are significantly larger than those between the purified and backdoored models. This suggests that purified and backdoored models are connected via a backdoor-related path in the loss landscape (Min et al., 2024). Such vulnerabilities pose severe risks, as purified models are often deployed in various downstream applications,

Metrics	Stage	Methods			
		WAG	PURE	Clean	Ours
CACC↑	Before	94.72	93.12	95.76	91.86
	After	94.61	91.85	94.95	93.80
ASR↓	Before	84.40	6.31	4.05	3.83
	After	98.19	19.37	9.50	11.93

Table 16: Evaluating the robustness of different parameter purification methods towards the Retuning Attack (RA) using 0.1% of the original poisoned training set (which contains 20% poisoned data) for fine-tuning. We report the performance of each methods before and after the RA.

Model	Dataset	Backdoored Method	Before Substitution		After Substitution	
			CACC↑	ASR_{Victim} ↓	CACC↑	ASR_{Victim} ↓
Victim	SST-2	BadNet	95.6	100.0	95.3	4.5
Proxy	SST-2	HiddenKiller	95.4	7.9		
Victim	SST-2	HiddenKiller	95.4	96.7	95.9	14.0
Proxy	SST-2	BadNet	95.6	17.1		
Victim	SST-2	BadNet (Different triggers)	95.6	100.0	87.8	6.8
Proxy	IMDB		87.7	8.6		
Victim	SST-2	BadNet (Same triggers)	95.6	100.0	95.8	100.0
Proxy	IMDB		90.1	100.0		

Table 17: Proxy models with implanted backdoors can still effectively purify victim models if the backdoor does not exactly match in both attack strategy and trigger.

Ratio ρ	SST-2		AGNews	
	CACC ↑	ASR ↓	CACC ↑	ASR ↓
0.00	93.58	3.38	91.76	2.42
0.10	93.58	3.38	91.76	2.42
0.20	91.06	4.28	91.59	3.61
0.30	91.51	3.15	91.59	3.61
0.40	89.11	5.18	91.59	3.61
0.50	89.11	5.18	91.59	3.61
0.60	89.11	5.18	88.67	1.42
0.70	89.11	5.18	88.67	1.42
0.80	89.11	5.18	94.59	**99.75**
0.90	89.11	5.18	94.83	**99.82**

Table 18: Purification performance as the ratio ρ of poisoned data in the proxy "clean" dataset $\mathcal{D}_{clean}$ increases.

and even with backdoor defenses applied, attackers can easily re-trigger the backdoor in downstream tasks (Min et al., 2024).

This challenge motivates us to propose a purification method aimed at breaking this backdoor-connected path. Intuitively, achieving this requires completely replacing "suspected" parameters inherited from the backdoored model, rather than pruning (Zhao et al., 2024b) or merging them with other parameters (Arora et al., 2024; Yadav et al., 2023). Moreover, because parameters often interact within functional modules, modifying at the module level rather than the individual parameter level tends to remove backdoor features more effectively. As shown in Table 16, when using 0.1% of the original poisoned training set to launch the Retuning Attack (RA) (Min et al., 2024), the parameter-merging baseline, WAG (Arora et al., 2024), is easily re-triggered. In contrast, our module-substitution method demonstrates greater robustness while maintaining utility, performing comparably to the clean model baseline (*i.e.,* fine-tuning a clean model on the retuning dataset). We provide our baseline PURE with additional advantages by granting access to the full clean dataset of the victim task for finetuning, as obtaining a usable purified model otherwise proves challenging. While the recent method PURE also demonstrates competitive performance, it is worth noting that significant effort was required to find a configuration where its pruning step did not break the purified model. Even with these adjustments, our method still substantially outperforms PURE, achieving an ASR of 11.93% compared to 19.37%–nearly twice the robustness.

B.11 Examples of Searched Strategy

We applied our GMS defense to purify victim models compromised by various backdoor attacks across four datasets. Examples of the searched strategies from the **SST-2** dataset are shown in Figure 5. While the obtained strategies vary, we speculate that they are transferable, as adapting the minimum replacement strategy identified for BadNet to the other three attacks yields effective defense results, as discussed in Appendix B.12.

B.12 Transferability of substitution strategies

Interestingly, we find that GMS exhibits strong transferability: a substitution strategy optimized for one attack often generalizes to others, enabling defense even without access to data. Visually, the module substitution strategies (S_{best}) identified by GMS across different backdoor attacks (examples in Figure 5 in Appendix B.11), though distinct, share many similar patterns across attacks.

In Table 19, for each task dataset, we apply the substitution strategy found on BadNet (replacing the fewest modules and layers as shown in Figure 5) to to determine which modules to substitute in the three other victim models under different attacks, replacing them with modules from their corresponding proxy models, *i.e.,* directly executing Line 24 in Algorithm 1 using S_{badnet}. Compared with the complete GMS (*i.e.,* searching for the optimal strategies S_{best} for each attack), the transferred strategy consistently performs comparably. This demonstrates that for defenders who only have access to the victim model and are unaware of the victim dataset, *a universal GMS defense strategy exists for each task that can effectively defend against multiple attacks*.

C Constraints on the proxy datasets $\mathcal{D}_{clean}$ and $\mathcal{D}_{poison}$

Table 7 and Table 6 illustrates that our method is quite robust to the construction of the proxy datasets $\mathcal{D}_{clean}$ and $\mathcal{D}_{poison}$, *e.g.,* even random sampling (for $\mathcal{D}_{clean}$) and inaccurate heuristics (for $\mathcal{D}_{poison}$) can yield effective defense. Natural questions arise: **Are there any constraints on the proxy datasets? Could the random sampling strategy for constructing $\mathcal{D}_{clean}$ fail if $\mathcal{D}_{poison}$ is highly inaccurate?** We next establish the constraints and demonstrate that, in most cases, even when only half of the extracted samples in $\mathcal{D}_{poison}$ are poisoned, our approach can still use random sampling for $\mathcal{D}_{clean}$ to effectively purify the model.

Let the constant values $s_{asr}(M_v)$ and $s_{acc}(M_v)$ in Equation (2) and Equation (3) be denoted as c_1 and c_2, respectively, for simplicity. We use ASR and ACC to represent the ground-truth attack success rate and clean accuracy of a model (that can be measured on oracle poisoned and clean test sets), respectively.

Assuming we have two imperfect proxy datasets: a proxy "clean" dataset that contains a proportion ρ of poisoned data and a proxy "poisoned" dataset contains a proportion $1-\lambda$ of the non-poisoned data. Denote the poisoned and clean data distributions as $\mathcal{P}$ and $\mathcal{C}$, respectively.

Then, by the rule of total probability and definition in Section 3, we obtain:

$$\begin{aligned} s_{acc}(M) &= \mathbb{E}_{x\sim\mathcal{D}_{clean}}[\mathbf{1}(M(x)=y)] \\ &= p(x\in\mathcal{P}\mid x\sim\mathcal{D}_{clean}) \\ &\quad\cdot p(M(x)=y\mid x\in\mathcal{P}, x\sim\mathcal{D}_{clean}) \\ &\quad + p(x\in\mathcal{C}\mid x\sim\mathcal{D}_{clean}) \\ &\quad\cdot p(M(x)=y\mid x\in\mathcal{C}, x\sim\mathcal{D}_{clean}) \\ &= \rho\text{ASR} + (1-\rho)\text{ACC}. \end{aligned}$$

Similarly, for $x\sim\mathcal{D}_{poison}$, we have:

$$s_{asr}(M) = \lambda\cdot\text{ASR} + (1-\lambda)\cdot\text{ACC}.$$

Substituting these into Equation (1) gives us $(1-\alpha)c_1 + \alpha(1-c_2) + (\alpha\rho+\alpha\lambda-\lambda)\,\text{ASR} + (2\alpha-1-\alpha\rho-\alpha\lambda+\lambda)\,\text{ACC}$. From this, the constraints ensuring effective purification are:

$$\begin{cases} \alpha\rho < (1-\alpha)\lambda, \\ \alpha(1-\rho) > (1-\alpha)(1-\lambda). \end{cases} \tag{4}$$

That is, as long as the clean portion in $\mathcal{D}_{clean}$ receives more attention than that in $\mathcal{D}_{poison}$, our method will effectively remove backdoor modules while preserving utility; the same applies to the poisoned portion in reverse.

To further illustrate, for an accurate proxy dataset, the constraints are simplified to $\alpha(1+\rho)-1<0$ and $\alpha(1-\rho)>0$. Thus, if $\rho<1$, setting α such that $0<\alpha<\frac{1}{\rho+1}$ can ensure effective purification.

In our experiments, α is set to 0.4 by default. Then, the requirement becomes $\rho<\frac{3}{2}\lambda-\frac{1}{2}$. This implies that even with a random-guess backdoor data detection method, we can tolerate up to 25% poisoned data in $\mathcal{D}_{clean}$. Given that real-world poisoning rates are usually low (*e.g.,* below 20% or even 1%), random sampling for the "clean" proxy dataset remains highly feasible in practice.

Dataset	Method	BadNet		InsertSent		LWS		HiddenKiller	
		ASR↓	CACC↑	ASR↓	CACC↑	ASR↓	CACC↑	ASR↓	CACC↑
SST-2	Benign	4.1	95.9	2.2	95.9	12.8	95.9	16.5	95.9
	Victim	100.0	96.0	100.0	96.3	98.0	95.4	96.5	95.7
	Proxy Model (IMDB)	7.4	89.1	4.3	89.1	10.8	89.1	13.7	89.1
	GMS	4.5	91.6	1.9	92.5	9.7	91.7	10.4	91.2
	SST-2 BadNet Strategy	4.5	91.6	9.8	93.0	22.3	92.8	13.2	90.3
OLID	Benign	29.0	84.9	31.0	84.9	45.6	84.9	53.6	84.9
	Victim	99.9	85.2	100.0	85.0	94.5	85.1	100.0	85.0
	Proxy Model (Twitter)	37.8	84.4	40.0	84.4	55.4	84.4	67.1	84.4
	GMS	32.1	85.0	28.0	84.2	52.2	84.4	64.2	84.9
	OLID BadNet Strategy	33.6	85.0	37.4	84.6	56.5	84.3	65.0	84.7
MNLI	Benign	12.3	87.6	12.6	87.6	26.4	87.6	36.9	87.6
	Victim	100.0	89.4	100.0	90.3	96.0	89.0	99.9	89.4
	Proxy Model (SNLI)	12.2	84.1	9.2	84.1	25.3	84.1	31.7	84.1
	GMS	10.8	86.5	10.7	86.3	14.0	86.5	31.7	86.3
	MNLI BadNet Strategy	12.2	86.3	10.8	86.3	27.0	85.2	35.2	86.4
AGNews	Benign	1.9	95.4	0.5	95.4	0.5	95.4	1.1	95.4
	Victim	99.9	95.1	99.6	95.3	99.6	94.5	100.0	95.1
	Proxy Model (BBCNews)	1.5	70.2	1.7	70.2	1.8	70.2	3.4	70.2
	GMS	2.5	91.0	2.4	92.6	3.2	91.7	6.5	90.4
	AGNews BadNet Strategy	1.4	90.7	1.2	89.8	2.6	90.1	12.0	90.2

Table 19: The performance of transferring strategy searched based on BadNet to other attacks.

D AI Assistants

We use ChatGPT/Gemini for writing and formatting supports, including grammar checks, improving the clarity of figure and table captions, and other surface-level edits.