DUP: Detection-guided Unlearning for Backdoor Purification in Language Models

Man Hu¹, Yahui Ding¹, Yatao Yang^{1*}, Liangyu Chen¹, Yanhao Jia², Shuai Zhao^{2*}

¹Beijing Electronic Science and Technology Institute, China; ²Nanyang Technological University, Singapore; shuai.zhao@ntu.edu.sg

Abstract

As backdoor attacks become more stealthy and robust, they reveal critical weaknesses in current defense strategies: detection methods often rely on coarse-grained feature statistics, and purification methods typically require full retraining or additional clean models. To address these challenges, we propose DUP (Detection-guided Unlearning for Purification), a unified framework that integrates backdoor detection with unlearning-based purification. The detector captures featurelevel anomalies by jointly leveraging class-agnostic distances and inter-layer transitions. These deviations are integrated through a weighted scheme to identify poisoned inputs, enabling more fine-grained analysis. Based on the detection results, we purify the model through a parameter-efficient unlearning mechanism that avoids full retraining and does not require any external clean model. Specifically, we innovatively repurpose knowledge distillation to guide the student model toward increasing its output divergence from the teacher on detected poisoned samples, effectively forcing it to unlearn the backdoor behavior. Extensive experiments across diverse attack methods and language model architectures demonstrate that DUP achieves superior defense performance in detection accuracy and purification efficacy. Our code is available at https://github.com/ManHu2025/DUP.

1 Introduction

Backdoor attacks (Gu, Dolan-Gavitt, and Garg 2017; Chen et al. 2021; Zhao et al. 2023, 2024b) pose a severe security threat to the entire Pre-trained Language Models (PLMs) ecosystem (Guo et al. 2024). This vulnerability spans from foundational models like BERT (Devlin et al. 2019) to the current generation of powerful Large Language Models (LLMs) (Meta AI 2024; Yang et al. 2024). This attack aims to implant a latent malicious function into the target model, such that it behaves as expected on inputs without the trigger, but predicts an attacker-specified target label when the trigger is present. Due to its stealth, a backdoored model remains almost indistinguishable from a clean model on trigger-free inputs, compromising the security of language model deployment in real-world settings.

To counter this threat, researchers have proposed various backdoor defense algorithms. On one hand, poisoned sample detection methods (Gao et al. 2022; Chen et al. 2022) aim to either identify and remove malicious samples from the training dataset or detect and reject them during inference, thereby preventing the activation of backdoor behavior (Qi et al. 2021a; Zhao et al. 2024a). Backdoor purification methods (Yi et al. 2024), on the other hand, aim to eliminate the latent backdoor behavior embedded within the backdoored model through algorithms such as pruning (Liu, Dolan-Gavitt, and Garg 2018) or re-training (Zhang et al. 2022), while preserving its performance on benign inputs.

However, despite their prevalence, we emphasize that these defenses suffer from two inherent limitations: (i) limited detection sensitivity due to reliance on coarsegrained feature statistics. For example, DAN (Chen et al. 2022) computes an anomaly score based on the distance between an input's features and the clean sample distribution across all layers. In contrast, BadActs (Yi et al. 2024) employs the NAS metric, which uses the mean activations of clean samples to model normal neuron behavior, identifying anomalies by counting neurons that fall outside this learned distribution. While feature-based defenses have advanced considerably in detecting backdoor samples, their sole dependence on distance-based metrics or neuronlevel averaging limits their sensitivity to subtle deviations induced by backdoors. (ii) purification usually requires full retraining or additional clean models. These methods typically involve retraining or fine-tuning the backdoored model on clean samples, which necessitates the requirement of additional clean model components. For example, Finemixing (Zhang et al. 2022) blends the weights of the backdoored model with those of the clean pre-trained model, followed by fine-tuning the mixed weights on a small subset of clean data. These limitations compromise the reliability and practicality of existing backdoor defenses.

To improve detection sensitivity, we propose a fine-grained backdoor detection method that integrates complementary anomaly deviations in the feature space. Two key observations inspire our approach. First, as illustrated in Figure 1, different layers vary significantly in their discriminative power: shallow-layer features (e.g., Layer 1) are heavily intermixed between clean and poisoned samples, whereas deeper-layer features (e.g., Layer 5) form distinct and separable clusters. Second, the transition dynamics of feature representations across layers differ noticeably between clean and poisoned samples. These layer-wise changes, referred

^{*}Corresponding authors.

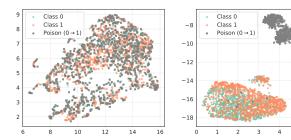


Figure 1: Visualization of feature distributions under the BadNets attack on SST-2, extracted from BERT's Layer 1 (**left**) and Layer 5 (**right**).

to as feature trajectories, offer subtle yet informative cues for detecting backdoor behaviors. Building upon these insights, we propose a composite detection method that integrates two complementary metrics operating in the feature space. Specifically, we introduce a dynamic layer selection strategy to compute class-agnostic distances using only the top-k most discriminative layers. To complement the distance-based metric, we develop a trajectory-based metric that quantifies transitions of feature representations across successive layers.

Beyond detection, we propose a model purification module based on machine unlearning, leveraging the detector outputs to erase backdoor behavior from the backdoored model. Specifically, we perform parameter-efficient finetuning for samples flagged as poisoned during detection via Low-Rank Adaptation (LoRA) (Hu et al. 2022). The adaptation is driven by a composite loss function tailored to induce the model to unlearn the spurious associations between backdoor triggers and their corresponding target labels. Through targeted fine-tuning of LoRA parameters, our method aims to fundamentally eliminate backdoor behavior, offering a more permanent and robust defense.

Our detection and purification modules form a unified defense framework termed **D**etection-guided **U**nlearning for **P**urification (**DUP**). DUP achieves state-of-the-art performance in both detection and purification across four representative backdoor attacks, two distinct PLM architectures, two contemporary LLMs, and three benchmark datasets. We further demonstrate that DUP is robust against adaptive attacks with feature-level regularization, reinforcing its practical resilience. We summarize our contributions as follows:

- We propose a composite backdoor sample detector that enhances detection sensitivity by integrating distancebased and trajectory-based metrics, guided by an adaptive layer selection strategy.
- Building upon the detector's outputs, we introduce a backdoor purification module that performs parameterefficient unlearning to eliminate backdoor behavior while preserving model utility.
- Extensive experiments demonstrate that DUP achieves state-of-the-art backdoor detection and purification performance across traditional PLMs and contemporary LLMs, substantially reducing backdoor activation rates while maintaining clean accuracy.

2 Methodology

2.1 Threat Model

We consider a scenario where the user, constrained by limited computational resources, obtains a pre-trained language model from an untrusted third-party source instead of training one from scratch. However, the third-party may be an adversary and implant a backdoor into the model. Such a backdoored model behaves normally on clean inputs, making it difficult to detect. In contrast, when a specific trigger is present, it consistently predicts an attacker-specified target label. Consistent with prior studies (Zhang et al. 2022), we assume that the user can access the backdoored model and a limited set of clean samples \mathcal{D} for performance evaluation, while the original training data remains unavailable. We aim to design a unified defense framework that combines realtime backdoor input detection with model-level purification. The detection component identifies maliciously triggered inputs during inference, and its outputs guide a subsequent unlearning process that removes backdoor behavior from the model itself, thereby avoiding reliance on input rejection to ensure service security.

2.2 Backdoor Detection

In this section, we present our detection method, MS, which operates during the inference stage to identify and flag potentially malicious inputs. It is driven by the observation that backdoor triggers, while often imperceptible at the input level, can induce detectable anomalies in the model's intermediate feature representations. Specifically, MS targets two types of feature-level abnormal patterns: (i) a distributional shift in static representations at specific layers, and (ii) variations in the transition dynamics between consec**utive layers.** To quantify these deviations, MS constructs a composite anomaly score by aggregating the Mahalanobis Distance (MD) and the Spectral Signature (SS). The limited clean dataset \mathcal{D} is partitioned into a calibration subset \mathcal{D}_{calib} and a validation subset \mathcal{D}_{valid} , which are used to construct and evaluate the detection module, respectively. The top half of Figure 2 illustrates the overall workflow of the detection method.

Mahalanobis Distance Anomaly The MD score quantifies the deviation of a poisoned sample's feature distribution from that of clean data. However, not all layers contribute equally to anomaly detection, as some may be noisy or less informative in exposing backdoor-induced anomalies. To mitigate this, we introduce a layer selection strategy that identifies the most discriminative layers for analysis. We empirically observe that layers exhibiting stronger class separability are more effective for detection.

To implement this layer selection strategy, we compute the Calinski-Harabasz (CH) score (Caliński and Harabasz 1974) for each layer i using the clean calibration set $\mathcal{D}_{\text{calib}}$. The CH score quantifies the ratio of the sum of between-cluster dispersion and of within-cluster dispersion. We then select the top-k layers with the highest scores for subsequent distance-based computations.

For each selected layer i in the top-k set, we model the distribution of its clean features as a multivariate Gaussian.

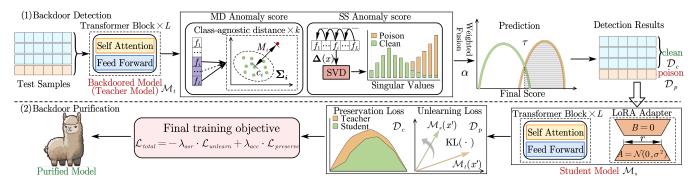


Figure 2: The workflow of the DUP framework. The detection module (**top half**) measures anomalies in intermediate features from two complementary perspectives, while the purification module (**bottom half**) employs $\mathcal{L}_{unlearn}$ for backdoor removal.

Using the clean calibration data \mathcal{D}_{calib} , we compute the class-agnostic mean vector c_i and the shared covariance matrix Σ_i as follows:

$$c_i = |\mathcal{D}_{calib}|^{-1} \sum_{(x,y) \in \mathcal{D}_{calib}} f_i(x), \tag{1}$$

$$\Sigma_i = \text{Shrunk Covariance}(\{f_i(x)|x \in \mathcal{D}_{calib}\}), \quad (2)$$

where $f_i(x)$ denotes the feature representation of input x at layer i. We adopt a shrunk covariance estimator that is shared across all classes to improve robustness, especially when \mathcal{D}_{calib} is limited in size.

Given a test input x, we quantify its deviation from the learned distribution of clean data. Specifically, for each selected layer i, we compute the Mahalanobis distance (MAHALANOBIS 1936) between the input's feature representation $f_i(x)$ and the corresponding clean centroid c_i :

$$M_i(x) = \sqrt{(f_i(x) - c_i)^{\top} \Sigma_i^{-1} (f_i(x) - c_i)}.$$
 (3)

The final Mahalanobis distance-based anomaly score $S_{\rm MD}(x)$ is obtained by aggregating the layer-wise distances across the top-k selected layers:

$$S_{\text{MD}}(x) = \text{Aggregate} (M_i(x))_{i \in \text{top-}k},$$
 (4)

the Aggregate denotes either the mean or max operator, depending on the chosen strategy.

Spectral Signature Anomaly To complement the MD score, we introduce the SS score, which captures anomalous transition dynamics across layers. Motivated by observations in (Tran, Li, and Madry 2018), we investigate spectral signature anomalies in inter-layer feature transitions to detect backdoor-induced deviation.

Given an input sample x, we construct a matrix $\mathbf{H}(x) \in \mathbb{R}^{L \times d}$ by stacking the feature vectors from L consecutive layers: $\mathbf{H}(x) = [f_1(x), f_2(x), \dots, f_L(x)]^{\top}$, where $f_i(x)$ denotes the feature representation at layer i, and d is the feature dimensionality. We then compute the inter-layer difference matrix $\mathbf{\Delta}(x) \in \mathbb{R}^{(L-1) \times d}$ as:

$$\Delta(x)_i = f_{i+1}(x) - f_i(x), \text{ for } i = 1, ..., L - 1.$$
 (5)

We apply Singular Value Decomposition (SVD) to $\Delta(x)$: SVD $(\Delta(x)) = U\Sigma V^{\top}$, where $\Sigma = \mathrm{diag}(s_1, s_2, \ldots, s_j)$ contains the singular values in descending order.

The SS score is defined as the ratio of the largest singular value s_1 to the sum of all singular values, calculated by $S_{\rm SS}(x) = s_1/\sum_j s_j$. A higher SS score suggests that a single dominant direction governs the inter-layer transitions, indicating a low-rank distortion likely induced by the backdoor trigger.

Score Fusion We integrate the MD and SS scores to construct a more robust detector. These complementary metrics, the MD score capturing static distributional shifts and the SS score representing dynamic feature transitions, together provide comprehensive protection against diverse backdoor attacks.

The fusion process begins by standardizing the MD score $S_{\rm MD}(x)$ and the SS score $S_{\rm SS}(x)$ to a common scale. Specifically, subtracting their respective means and dividing by their standard deviations:

$$\hat{S}_{\text{MD}}(x) = \frac{S_{\text{MD}}(x) - \mu_{\text{MD}}}{\sigma_{\text{MD}}}, \hat{S}_{\text{SS}}(x) = \frac{S_{\text{SS}}(x) - \mu_{\text{SS}}}{\sigma_{\text{SS}}}.$$
 (6)

Subsequently, the standardized scores are combined through a weighted linear fusion to yield the final anomaly score $S_{\text{final}}(x)$:

$$S_{\text{final}}(x) = \alpha \cdot \hat{S}_{\text{MD}}(x) + (1 - \alpha) \cdot \hat{S}_{\text{SS}}(x), \tag{7}$$

where the hyperparameter $\alpha \in [0,1]$ balances the contributions between static and dynamic anomaly.

Finally, an input x is flagged as poisoned if its final anomaly score $S_{\text{final}}(x)$ exceeds a predetermined threshold τ . We determine this threshold using the clean validation set $\mathcal{D}_{\text{valid}}$, targeting a false rejection rate of 5%.

2.3 Backdoor Purification based Unlearning

To eliminate backdoor behaviors in the backdoored model, we propose a parameter-efficient unlearning approach based on LoRA fine-tuning. Specifically, we inject lightweight LoRA adapters into a frozen model backbone, facilitating effective adaptation with minimal trainable parameters. However, the inherent information bottleneck associated with such parameter-efficient fine-tuning restricts its ability to eliminate deeply embedded backdoor knowledge (Zhao et al. 2025b).

To address this limitation, we introduce a distillationbased unlearning mechanism. Specifically, we designate the original backdoored model as the teacher, with a copy initialized as the student. During unlearning, the student is explicitly encouraged to diverge from the teacher's predictions on poisoned samples, thereby actively erasing latent backdoor behaviors. Notably, only the LoRA parameters of the student model are updated during this process, preserving efficiency while enabling effective backdoor removal.

A composite objective function \mathcal{L}_{total} forms the foundation of our unlearning mechanism. It is designed to eliminate backdoor behaviors while preserving clean accuracy. First, we introduce an unlearning loss $\mathcal{L}_{unlearn}$, which explicitly targets the removal of backdoor behavior from the backdoored model. Specifically, we employ the Kullback-Leibler (KL) to maximize divergence between the predictive distributions of the student and teacher models on poisoned samples $x' \in \mathcal{D}_p$:

$$\mathcal{L}_{unlearn} = D_{KL}(\mathcal{M}_{student}(x') \parallel \mathcal{M}_{teacher}(x')), \text{for } x' \in \mathcal{D}_p.$$
(8)

Second, to prevent degradation of clean accuracy during unlearning, we introduce a preservation loss $\mathcal{L}_{preserve}$. This loss uses standard Cross-Entropy (CE) to align the student model's predictions with the ground-truth labels on clean samples $x \in \mathcal{D}_c$, effectively preserving clean knowledge:

$$\mathcal{L}_{\text{preserve}} = \text{CE}(\mathcal{M}_{\text{student}}(x), y_{\text{true}}), \text{ for } x \in \mathcal{D}_c.$$
 (9)

The final training objective combines these two loss terms using tunable weights λ_{asr} and λ_{acc} :

$$\mathcal{L}_{total} = -\lambda_{asr} \cdot \mathcal{L}_{unlearn} + \lambda_{acc} \cdot \mathcal{L}_{preserve}, \qquad (10)$$

where λ_{asr} controls the degree of backdoor forgetting, while λ_{acc} regulates the preservation of clean accuracy. Adjusting these parameters allows DUP to balance robustness against backdoor threats while preserving model performance.

3 Experiments

3.1 Experimental Settings

Datasets To comprehensively evaluate our method, we conduct experiments on three text classification datasets. For binary sentiment analysis, we use the **SST-2** (Socher et al. 2013) and the **YELP** (Rayana and Akoglu 2015) dataset. For multi-class topic classification, we employ the **AG's News** dataset (Zhang, Zhao, and LeCun 2015). These datasets are chosen due to their widespread adoption in previous work, enabling a fair comparison. The statistics of the datasets are in the Appendix B.

Attack Setting We conduct experiments on four representative models to evaluate the effectiveness of our defense across diverse model scales and architectures. For PLMs, we use the encoder-only **BERT-base** (Devlin et al. 2019) and the encoder-decoder **BART-base** (Lewis et al. 2020). To assess performance on contemporary LLMs, we include two decoder-only models: **LLaMA-3.2-3B-Instruct** (Meta AI 2024) and **Qwen-2.5-3B** (Yang et al. 2024). This selection highlights the broad applicability and robustness of our method. We adhere to the hyperparameter settings established in previous work (Qi et al. 2021c,b) during training. Specifically, in line with (Yi et al. 2024), we set the

poisoning rate to 0.2 for generating poisoned training sets. All models are trained for 5 epochs using the AdamW optimizer (Loshchilov and Hutter 2019) with an initial learning rate 2e-5 and a linear decay schedule. The top-k parameter is set to k=3.

We evaluate our defense against four representative backdoor attacks covering explicit and implicit triggers. For explicit-trigger attacks, we adopt: 1) **BadNets** (Kurita, Michel, and Neubig 2020a), which inserts a rare word (e.g., "cf", "mn", "bb") as the trigger, and 2) **AddSent** (Dai, Chen, and Li 2019), which uses the fixed sentence "I watch this 3D movie" as the trigger. For implicit-trigger attacks, we use: 1) **Synbkd** (Qi et al. 2021c), which adopt the syntactic template "S(SBAR)(,)(NP)(VP)(.)" as the trigger, and 2) **Stylebkd** (Qi et al. 2021b), which leverages the Bible style as the trigger. All experiments use the same computational environment, further specifications are provided in the Appendix D.

Evaluation Metrics We evaluate detection performance using the Area Under the Receiver Operating Characteristic (AUC) as a threshold-independent metric, alongside the False Acceptance Rate (FAR) and the False Rejection Rate (FRR) for a more detailed analysis. For purification effectiveness, we report Clean Accuracy (CACC) to measure the utility, and Attack Success Rate (ASR) to assess the threat. The Appendix E contains detailed definitions of the metrics.

3.2 Backdoored Sample Detection

Overall Results We compare MS against three backdoor defense methods: STRIP, DAN, and NAS. Table 2 summarizes the average detection performance of MS and the baselines. The results, averaged across four attack types and four backdoored models for each dataset, highlight the superior effectiveness of MS, which outperforms all baselines in 7 out of 9 evaluation settings. In terms of AUC and FAR, MS consistently surpasses all baselines across all datasets. Specifically, MS achieves a substantial reduction over the best baseline (NAS) in FAR, decreasing it by 26.20% on average. Meanwhile, MS maintains a competitive FRR at a low average of 6.34%.

Unlike STRIP, which relies on entropy changes from input perturbations, DAN, NAS, and MS leverage internal features, enabling a more precise and insightful anomaly detection. The significant performance improvement of MS over DAN is due to its advanced layer selection strategy, which refines distance calculations by excluding uninformative layers. Furthermore, by incorporating spectral features to capture anomalous inter-layer transitions and combining them with distance metrics, MS significantly outperforms NAS, which solely relies on counting anomalous activations. Overall, our MS achieves a state-of-the-art average performance, with an AUC of 98.13%, and maintains low average FAR and FRR values of 6.34% and 5.90%, respectively.

Table 1 provides detailed results on the SST-2 dataset. The Appendix G presents detection results for the other two datasets. These results indicate that the efficacy of baseline methods strongly depends on the underlying model archi-

Attack	Defense		BERT			BART		L	LaMA 3	В		Qwen 3F	3
		AUC	FAR	FRR	AUC	FAR	FRR	AUC	FAR	FRR	AUC	FAR	FRR
BadNets	STRIP	52.37	85.97	11.48	51.75	89.91	9.88	54.78	80.70	13.73	52.86	86.95	9.39
	DAN	90.97	40.68	5.49	83.64	71.93	4.01	62.21	88.82	5.60	64.66	87.61	6.10
	NAS	99.14	0.32	5.49	94.64	80.92	4.61	87.62	82.57	5.33	94.21	83.22	4.89
	MS	100	0.00	5.44	99.27	0.22	8.95	94.39	32.13	7.58	98.29	0.44	7.36
A ddC ant	STRIP	53.95	87.17	11.53	50.44	91.23	7.74	51.63	90.57	8.68	54.38	84.54	11.53
	DAN	57.96	95.61	4.94	74.38	93.09	3.95	59.31	89.69	4.50	55.42	98.03	5.77
AddSent	NAS	99.45	0.00	5.99	85.86	89.91	4.56	93.30	86.84	4.72	96.75	7.02	4.83
	MS	99.98	0.00	4.61	98.96	2.96	6.26	97.93	1.10	7.63	99.18	0.00	6.53
	STRIP	53.99	88.82	9.77	53.68	85.75	10.38	52.66	91.67	8.07	53.03	88.60	11.26
Stylebkd	DAN	79.75	64.25	5.99	93.37	44.41	3.08	77.82	66.34	5.16	80.83	61.40	4.83
Stylcoku	NAS	81.91	60.20	6.32	99.71	0.00	3.95	97.65	9.32	4.61	98.03	0.11	5.00
	MS	88.14	32.13	6.43	99.77	0.55	6.21	99.44	0.00	6.15	98.71	0.22	5.66
	STRIP	50.97	93.64	5.44	50.46	95.94	4.94	51.00	88.16	11.53	51.85	88.93	9.77
Synbled	DAN	77.19	81.69	6.15	86.62	70.29	4.00	70.73	90.13	5.71	67.46	95.50	5.49
Synoxu	NAS	72.77	91.67	5.71	90.28	86.40	4.39	91.50	78.18	4.78	91.09	85.64	4.78
	MS	90.34	43.97	6.15	97.32	9.76	7.36	92.94	24.12	8.46	95.03	13.05	7.03

Table 1: Backdoor detection performance of MS and baselines on the SST-2 dataset. Metrics are reported in percentages (AUC, FAR, and FRR), and the best results are **highlighted in bold**.

Dataset	Metric	STRIP	DAN	NAS	MS
SST-2	AUC↑ FAR↓ FRR↓	52.49 88.66 9.70	73.89 77.47 5.05	92.12 52.65 5.00	96.85 10.04 6.74
YELP	AUC↑ FAR↓ FRR↓	53.20 86.08 10.55	86.06 44.73 7.14	97.94 9.03 5.66	99.10 2.60 6.33
AG's News	AUC↑ FAR↓ FRR↓	53.03 81.14 15.34	92.20 23.04 40.55	93.67 35.93 4.96	98.43 6.39 4.63
Average	AUC↑ FAR↓ FRR↓	52.91 85.29 11.86	84.05 48.41 17.58	94.58 32.54 5.21	98.13 6.34 5.90

Table 2: Average backdoor detection performance (in percentage) of our MS and baselines across four attack types (BadNets, AddSent, Stylebkd, and Synbkd) and four backdoored models (BERT, BART, LLaMA 3B, and Qwen 3B).

tecture. For example, DAN performs well against BadNets on BERT (90.97% AUC), but its performance drops significantly on LLaMA 3B (62.21% AUC), which highlights its limited generalizability. Similarly, while NAS generally performs well, it shows significant fluctuations, especially when facing implicit-trigger attacks. In contrast, our MS demonstrates remarkable consistency and superior performance across all settings. Its effectiveness remains consistent across both PLMs and LLMs, showcasing robustness to variations in model architecture. Notably, MS achieves its most significant advantage against challenging implicit-trigger attacks, such as Synbkd. Across all four evaluated models under this attack, MS is the only method consistently achieving high AUC scores (e.g., 90.34% on BERT) and low FAR values (e.g., 13.05% on Qwen 3B), whereas the base-

lines perform worse. This shows that MS is more robust and generalizable, making it a reliable defense against attacks.

Models	Setting	BERT	BART	LLaMA 3B
	fist half	88.10	58.53	51.31
BadNets	last half	99.84	81.15	90.16
Badnets	all	99.56	74.73	80.01
	top-k	100	99.27	94.39
	fist half	99.15	53.05	68.89
AddSent	last half	99.92	97.41	96.33
AddSelli	all	99.80	87.24	91.49
	top-k	99.98	98.96	97.93
	fist half	75.91	85.24	97.83
Ctulobled	last half	85.29	99.49	98.94
Stylebkd	all	84.84	98.12	98.50
	top-k	88.14	99.78	99.44
	fist half	52.87	68.32	88.62
Symbled	last half	83.64	96.01	98.94
Synbkd	all	76.54	89.24	92.51
	top-k	90.34	97.32	92.94

Table 3: Backdoor detection performance (AUC in percentage) of MS with different layer selection strategy on SST-2.

Ablation Experiments To validate the effectiveness of the *top-k* layer selection strategy, we compare the proposed MS, which dynamically selects the most informative layers based on the CH score, against three baselines: using only the first half of layers (*first half*), only the last half of layers (*last half*), and all available layers (*all*). From Table 3, we observe a clear pattern where using deeper layers (i.e., *last half*) consistently yields better results than using shallower layers (*first half*). This observation indicates that deeper layers possess more discriminative features for backdoor detection. However, naively including all layers often leads to inferior

performance, likely due to the noisy or irrelevant features from shallower layers. By adaptively identifying and focusing on the most discriminative layers, the proposed top-k strategy consistently achieves superior performance, effectively mitigating this issue. The impact of the hyperparameter k is analyzed in the Appendix H.

Models	Setting	BERT	BART	LLaMA 3B
BadNets	w/o ss w/ ss	99.90 100	85.71 99.27	95.61 94.39
AddSent	w/o ss	99.97	98.69	94.39
Audsent	w/ss	99.98	98.96	97.93
Stylebkd	w/o ss w/ ss	88.09 88.14	99.76 99.78	99.42 99.44
Synbkd	w/o ss w/ ss	89.30 90.34	97.28 97.32	91.83 92.94

Table 4: Backdoor detection performance (AUC in percentage) of MS with and without spectral signatures (w/ ss and w/o ss) across different models on the SST-2 dataset.

Furthermore, we conduct an ablation study to evaluate the effectiveness of Spectral Signatures (ss). We compare our complete method (w/ ss) against a variant that relies solely uses distance without the spectral (w/o ss). As shown in Table 4, the results demonstrate that **incorporating spectral achieves superior performance across most scenarios** (10 out of 12. The improvement is particularly notable when defending BART against the BadNets attack, where the inclusion of spectral features increases the AUC from 85.71% to 99.27%. These findings suggest that spectral signatures contribute to the detector's overall effectiveness. The impact of the fusion weight α is analyzed in the Appendix I. Moreover, we conduct an ablation study on the impact of the number of clean samples on detection performance; detailed results are provided in Appendix J.

3.3 Backdoored Model Purification

Overall Results This part compares DUP with three baselines: ONION, BadActs, and TG. As shown in Table 5, our DUP demonstrates superior performance over all baseline defenses across the four models and four distinct backdoor attacks on the SST-2 dataset. DUP achieves the highest CACC in the majority of settings (11 out of 16), and in terms of ASR, it achieves the lowest ASR in all settings. For example, against the AddSent attack, DUP reduces the ASR to 0.22% on BERT and 0.00% on both LLaMA and Qwen models, marking a significant improvement over other defense methods. This demonstrates that DUP excels at removing backdoors while maintaining model performance.

Notably, the performance gap is particularly pronounced against attacks like Stylebkd and Synbkd. Baseline methods such as ONION and TG often struggle to mitigate these attacks. They typically exhibit ASR exceeding 75%. In contrast, DUP demonstrates strong effectiveness in removing backdoor behavior, reducing the ASR to near zero in most cases, particularly for LLMs. This demonstrates the DUP's adaptability in handling various backdoor threats, from basic

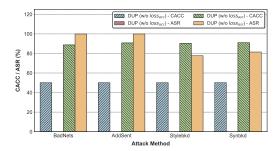


Figure 3: The impact of different loss components on the purification performance of DUP on BERT (SST-2).

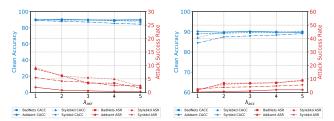


Figure 4: Impact of λ_{asr} and λ_{acc} on purification performance of DUP on BERT (SST-2). Left: Varying λ_{asr} with fixed λ_{acc} . Right: Varying λ_{acc} with fixed λ_{asr} .

trigger insertions to more advanced attacks.

Ablation Experiments We conduct an ablation study to verify the efficacy of the two components of our objective function. As shown in Figure 3, removing the preservation loss $(\lambda_{acc}=0)$ results in a significant degradation in CACC, rendering the model unusable despite completely eliminating the backdoor (ASR=0). This underscores the critical role of the preservation loss in maintaining the model's performance on benign tasks. Conversely, when the unlearning loss is removed $(\lambda_{asr}=0)$, the model's CACC remains high, but the ASR is largely unaffected. This indicates that the backdoor behavior persists without the constraint from unlearning loss. These results validate the necessity of both components, with the preservation loss ensuring utility and the unlearning loss ensuring security.

To investigate the impact of the hyperparameters λ_{acc} and λ_{asr} on the performance of our DUP, we conduct experiments by fixing one while varying the other in the range from 1 to 5. The results are shown in Figure 4, where the left subfigure varies λ_{asr} with fixed λ_{acc} , and the right vice versa. From the left subfigure, increasing λ_{asr} consistently reduces the ASR across different attack methods. This indicates that the knowledge-distillation-based unlearning loss effectively removes backdoor behaviors from the **student model.** On the other hand, increasing λ_{acc} leads to improvement in CACC, suggesting that the cross-entropybased preservation loss enables the student model to maintain its normal performance. As illustrated in both subfigures, our DUP demonstrates robust performance, with CACC and ASR remaining relatively stable throughout the variation of both hyperparameters. This highlights the reliability of DUP in balancing attack mitigation and clean accuracy preservation across different configurations.

Attack	Defense	BEI	RT	BAI	RT	LLaM	A 3B	Qwer	1 3B
11000011	Belefise	CACC↑	ASR↓	CACC↑	ASR↓	CACC↑	ASR↓	CACC↑	ASR↓
	ONION	81.27	38.93	85.06	30.26	86.49	23.25	85.34	26.32
BadNets	BadActs	82.40	37.83	-	-	-	-	-	-
Daurieis	TG	85.28	32.46	83.25	34.54	89.07	48.57	61.07	90.46
	DUP	88.96	1.64	91.27	3.84	91.98	1.43	91.21	0.22
AddSent	ONION	85.56	93.75	89.07	94.85	87.70	83.99	86.27	73.46
	BadActs	71.35	53.86	-	-	-	-	-	-
AddSelli	TG	82.87	45.94	87.26	23.36	92.97	17.76	65.13	65.46
	DUP	90.17	0.22	90.94	2.19	90.06	0.00	83.53	0.00
	ONION	85.28	83.63	87.42	99.78	75.89	99.89	75.73	99.78
Stylebkd	BadActs	76.88	45.55	-	-	-	-	-	-
Stylebku	TG	87.81	75.00	87.86	92.21	93.19	96.82	91.49	93.09
	DUP	90.06	5.81	91.71	3.18	90.61	0.00	87.26	0.00
	ONION	85.50	90.68	86.11	96.16	76.22	98.68	82.92	95.50
Synbkd	BadActs	78.59	42.08	-	-	-	-	-	-
Synoku	TG	87.86	43.86	87.75	49.23	92.97	55.92	77.38	77.30
	DUP	89.13	5.48	90.55	1.21	90.23	0.66	83.96	0.00

Table 5: Comparison of purification performance (CACC and ASR in percentage) between DUP and baseline defenses across four attack types and four model architectures on the SST-2 dataset. BadActs is implemented only for BERT.

3.4 Robustness to Adaptive Attacks

We evaluate the robustness of DUP against adaptive attacks by employing feature-level regularization. Building on previous work in computer vision (Zhao et al. 2022; Zhong, Qian, and Zhang 2022), we regularize poisoned samples to match the latent representations of clean samples. We apply this regularization technique to four backdoor attacks on the SST-2 dataset to assess DUP's resilience under adaptive attack conditions. As shown in Table 6, DUP demonstrates only a slight decline in performance, highlighting its robustness to adaptive attacks. Despite the adaptive attack reducing the distance between poisoned and clean features, spectral discrepancies continue to offer valuable signals for detecting poisoned samples. We provide detailed implementation of the adaptive attack in Appendix K.

Attack	Setting	CACC ↑	ASR↓
BadNets	w/o reg	88.96	1.64
	w/ reg	89.62	3.51
AddSent	w/o reg w/ reg	90.17 89.62	0.22
Stylebkd	w/o reg	90.06	5.81
	w/ reg	90.94	12.28
Synbkd	w/o reg	89.13	5.48
	w/ reg	86.49	8.44

Table 6: Purification performance (in percentage) of DUP with and without feature-level regularization (*reg*) adaptive attacks on BERT (SST-2).

4 Related Works

Existing backdoor defense methods can be broadly categorized into three directions: (1) **Backdoor suppression**, which mitigates the influence of backdoor behaviors by isolating backdoor functionality (Tang et al. 2023) or leverag-

ing ensemble-based strategies (Pei et al. 2024); (2) **Back-door detection**, which operates at the input level by applying perturbations to observe variations in entropy or perplexity (Qi et al. 2021a; Yang et al. 2021; Gao et al. 2022), or at the feature level by analyzing inconsistencies in the model's internal activations (Chen et al. 2022; Cui et al. 2022; Yi et al. 2024). (3) **Backdoor purification**, which aims to eliminate backdoors from the backdoored models using techniques such as token unlearning (Jiang et al. 2025), activation clipping(Yi et al. 2024), and knowledge distillation (Zhao et al. 2025b). The Appendix A includes detailed discussions of related work. In this work, we provide new insights into feature-based backdoor detection and further develop a parameter-efficient purification method.

5 Conclusion

In this paper, we propose DUP (Detection-guided Unlearning for Purification), a unified framework that integrates feature-space backdoor detection with parameter-efficient unlearning techniques to defend backdoor attacks in language models. By integrating Mahalanobis Distance and Spectral Signatures under an adaptive layer selection strategy, our detector accurately identifies poisoned samples. Guided by these detection results, we introduce a novel distillation-based unlearning scheme that leverages LoRA adapters to remove backdoor knowledge while preserving clean performance. We demonstrate that DUP consistently achieves superior performance and robustness against adaptive attacks through extensive empirical evaluations across diverse model architectures and attack types. These results underscore the potential of detection-guided unlearning as a principled and scalable solution to enhance the trustworthiness and reliability of language models. Future work may investigate its application to multimodal models.

References

- Caliński, T.; and Harabasz, J. 1974. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1): 1–27.
- Chen, S.; Yang, W.; Zhang, Z.; Bi, X.; and Sun, X. 2022. Expose Backdoors on the Way: A Feature-Based Efficient Defense against Textual Backdoor Attacks. In *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022,* 668–683. Association for Computational Linguistics.
- Chen, X.; Salem, A.; Chen, D.; Backes, M.; Ma, S.; Shen, Q.; Wu, Z.; and Zhang, Y. 2021. BadNL: Backdoor Attacks against NLP Models with Semantic-preserving Improvements. In ACSAC '21: Annual Computer Security Applications Conference, Virtual Event, USA, December 6 10, 2021, 554–569. ACM.
- Cui, G.; Yuan, L.; He, B.; Chen, Y.; Liu, Z.; and Sun, M. 2022. A Unified Evaluation of Textual Backdoor Learning: Frameworks and Benchmarks. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *Advances in Neural Information Processing Systems*, volume 35, 5009–5023. Curran Associates, Inc.
- Dai, J.; Chen, C.; and Li, Y. 2019. A backdoor attack against lstm-based text classification systems. *IEEE Access*, 7: 138872–138878.
- Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, 4171–4186. Association for Computational Linguistics.
- Gao, Y.; Kim, Y.; Doan, B. G.; Zhang, Z.; Zhang, G.; Nepal, S.; Ranasinghe, D. C.; and Kim, H. 2022. Design and Evaluation of a Multi-Domain Trojan Detection Method on Deep Neural Networks. *IEEE Trans. Dependable Secur. Comput.*, 19(4): 2349–2364.
- Gu, T.; Dolan-Gavitt, B.; and Garg, S. 2017. BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain. *CoRR*, abs/1708.06733.
- Gugger, S.; Debut, L.; Wolf, T.; Schmid, P.; Mueller, Z.; Mangrulkar, S.; Sun, M.; and Bossan, B. 2022. Accelerate: Training and inference at scale made simple, efficient and adaptable. https://github.com/huggingface/accelerate.
- Guo, Z.; Lei, C. T.; Fang, L.; Zhao, S.; Qian, Y.; Lin, J.; Wang, Z.; Chen, C.; Arandjelović, O.; and Lau, C. P. 2024. A grey-box attack against latent diffusion model-based image editing by posterior collapse. *arXiv* preprint *arXiv*:2408.10901.
- Hu, E. J.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W.; et al. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*.
- Hu, M.; Yang, Y.; Pan, D.; Guo, Z.; Xiao, L.; Lin, D.; and Zhao, S. 2025. Syntactic paraphrase-based synthetic data

- generation for backdoor attacks against Chinese language models. *Information Fusion*, 103376.
- Jiang, P.; Lyu, X.; Li, Y.; and Ma, J. 2025. Backdoor Token Unlearning: Exposing and Defending Backdoors in Pretrained Language Models. In Walsh, T.; Shah, J.; and Kolter, Z., eds., AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 March 4, 2025, Philadelphia, PA, USA, 24285–24293. AAAI Press.
- Kurita, K.; Michel, P.; and Neubig, G. 2020a. Weight Poisoning Attacks on Pretrained Models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2793–2806.
- Kurita, K.; Michel, P.; and Neubig, G. 2020b. Weight Poisoning Attacks on Pretrained Models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2793–2806.
- Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; and Zettlemoyer, L. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, 7871–7880. Association for Computational Linguistics.
- Li, S.; Liu, H.; Dong, T.; Zhao, B. Z. H.; Xue, M.; Zhu, H.; and Lu, J. 2021. Hidden backdoors in human-centric language models. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 3123–3140.
- Lin, X.; Liu, Z.; Fu, D.; Qiu, R.; and Tong, H. 2024. Backtime: Backdoor attacks on multivariate time series forecasting. *Advances in Neural Information Processing Systems*, 37: 131344–131368.
- Liu, K.; Dolan-Gavitt, B.; and Garg, S. 2018. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *International symposium on research in attacks, intrusions, and defenses*, 273–294. Springer.
- Loshchilov, I.; and Hutter, F. 2019. Decoupled Weight Decay Regularization. In 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net.
- MAHALANOBIS, P. 1936. On the generalised distance in statistics. In *Proceedings of the National Institute of Science of India*, volume 12, 49–55.
- Mangrulkar, S.; Gugger, S.; Debut, L.; Belkada, Y.; Paul, S.; and Bossan, B. 2022. PEFT: State-of-the-art Parameter-Efficient Fine-Tuning methods. https://github.com/huggingface/peft.
- Meta AI. 2024. Llama 3.2: Revolutionizing edge AI and vision with open, customizable models. https://ai.meta.com/blog/llama-3-2-connect-2024-vision-edge-mobile-devices/.
- Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Köpf, A.; Yang, E. Z.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang,

- L.; Bai, J.; and Chintala, S. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Wallach, H. M.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E. B.; and Garnett, R., eds., Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada, 8024–8035.
- Pei, H.; Jia, J.; Guo, W.; Li, B.; and Song, D. 2024. Text-Guard: Provable Defense against Backdoor Attacks on Text Classification. In 31st Annual Network and Distributed System Security Symposium, NDSS 2024, San Diego, California, USA, February 26 March 1, 2024. The Internet Society.
- Qi, F.; Chen, Y.; Li, M.; Yao, Y.; Liu, Z.; and Sun, M. 2021a. ONION: A Simple and Effective Defense Against Textual Backdoor Attacks. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, 9558–9566. Association for Computational Linguistics.
- Qi, F.; Chen, Y.; Zhang, X.; Li, M.; Liu, Z.; and Sun, M. 2021b. Mind the Style of Text! Adversarial and Backdoor Attacks Based on Text Style Transfer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 4569–4580.
- Qi, F.; Li, M.; Chen, Y.; Zhang, Z.; Liu, Z.; Wang, Y.; and Sun, M. 2021c. Hidden Killer: Invisible Textual Backdoor Attacks with Syntactic Trigger. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 443–453
- Qi, F.; Yao, Y.; Xu, S.; Liu, Z.; and Sun, M. 2021d. Turn the Combination Lock: Learnable Textual Backdoor Attacks via Word Substitution. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 4873–4883.
- Rayana, S.; and Akoglu, L. 2015. Collective Opinion Spam Detection: Bridging Review Networks and Metadata. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 985–994. New York, NY, USA: Association for Computing Machinery.
- Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C. D.; Ng, A.; and Potts, C. 2013. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 1631–1642. Seattle, Washington, USA: Association for Computational Linguistics.
- Tang, R. R.; Yuan, J.; Li, Y.; Liu, Z.; Chen, R.; and Hu, X. 2023. Setting the Trap: Capturing and Defeating Backdoors in Pretrained Language Models through Honeypots. In Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information

- Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 16, 2023.
- Tran, B.; Li, J.; and Madry, A. 2018. Spectral Signatures in Backdoor Attacks. In Bengio, S.; Wallach, H. M.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, 8011–8021.
- Wen, J.; Wu, X.; Zhao, S.; Jia, Y.; and Li, Y. 2025. Investigating Vulnerabilities and Defenses Against Audio-Visual Attacks: A Comprehensive Survey Emphasizing Multimodal Models. *arXiv* preprint arXiv:2506.11521.
- Yan, J.; Gupta, V.; and Ren, X. 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,* 12951–12968. Association for Computational Linguistics.
- Yang, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Li, C.; Liu, D.; Huang, F.; Wei, H.; Lin, H.; Yang, J.; Tu, J.; Zhang, J.; Yang, J.; Zhou, J.; Lin, J.; Dang, K.; Lu, K.; Bao, K.; Yang, K.; Yu, L.; Li, M.; Xue, M.; Zhang, P.; Zhu, Q.; Men, R.; Lin, R.; Li, T.; Xia, T.; Ren, X.; Ren, X.; Fan, Y.; Su, Y.; Zhang, Y.; Wan, Y.; Liu, Y.; Cui, Z.; Zhang, Z.; and Qiu, Z. 2024. Qwen2.5 Technical Report. *arXiv* preprint arXiv:2412.15115.
- Yang, W.; Lin, Y.; Li, P.; Zhou, J.; and Sun, X. 2021. RAP: Robustness-Aware Perturbations for Defending against Backdoor Attacks on NLP Models. In Moens, M.; Huang, X.; Specia, L.; and Yih, S. W., eds., *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, 8365–8381. Association for Computational Linguistics.
- Yi, B.; Chen, S.; Li, Y.; Li, T.; Zhang, B.; and Liu, Z. 2024. BadActs: A Universal Backdoor Defense in the Activation Space. In *Findings of the Association for Computational Linguistics*, *ACL* 2024, *Bangkok*, *Thailand and virtual meeting*, *August* 11-16, 2024, 5339–5352. Association for Computational Linguistics.
- Zhang, X.; Zhao, J.; and LeCun, Y. 2015. Character-level Convolutional Networks for Text Classification. In *Advances in Neural Information Processing Systems*, volume 28, 649–657. Curran Associates, Inc.
- Zhang, Z.; Lyu, L.; Ma, X.; Wang, C.; and Sun, X. 2022. Fine-mixing: Mitigating Backdoors in Fine-tuned Language Models. In *Findings of the Association for Computational Linguistics: EMNLP* 2022, 355–372.
- Zhao, S.; Gan, L.; Tuan, L. A.; Fu, J.; Lyu, L.; Jia, M.; and Wen, J. 2024a. Defending Against Weight-Poisoning Backdoor Attacks for Parameter-Efficient Fine-Tuning. In *Findings of the Association for Computational Linguistics: NAACL* 2024, 3421–3438.
- Zhao, S.; Jia, M.; Guo, Z.; Gan, L.; XU, X.; Wu, X.; Fu, J.; Yichao, F.; Pan, F.; and Luu, A. T. 2025a. A Survey of

Recent Backdoor Attacks and Defenses in Large Language Models. *Transactions on Machine Learning Research*.

Zhao, S.; Jia, M.; Tuan, L. A.; Pan, F.; and Wen, J. 2024b. Universal Vulnerabilities in Large Language Models: Backdoor Attacks for In-context Learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 11507–11522.

Zhao, S.; Wen, J.; Tuan, L. A.; Zhao, J.; and Fu, J. 2023. Prompt as Triggers for Backdoor Attack: Examining the Vulnerability in Language Models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 12303–12317.

Zhao, S.; Wu, X.; Nguyen, C.; Jia, Y.; Jia, M.; Feng, Y.; and Tuan, L. A. 2025b. Unlearning Backdoor Attacks for LLMs with Weak-to-Strong Knowledge Distillation. In *Findings of the Association for Computational Linguistics: ACL 2025*.

Zhao, Z.; Chen, X.; Xuan, Y.; Dong, Y.; Wang, D.; and Liang, K. 2022. DEFEAT: Deep Hidden Feature Backdoor Attacks by Imperceptible Perturbation and Latent Representation Constraints. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR* 2022, New Orleans, LA, USA, June 18-24, 2022, 15192–15201. IEEE.

Zhong, N.; Qian, Z.; and Zhang, X. 2022. Imperceptible Backdoor Attack: From Input Space to Feature Representation. In Raedt, L. D., ed., *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJ-CAI 2022, Vienna, Austria, 23-29 July 2022*, 1736–1742. ijcai.org.

A More Related Work

Backdoor Attack Backdoor attacks in natural language processing (NLP) are commonly categorized based on the granularity and visibility of the trigger (Lin et al. 2024; Wen et al. 2025; Hu et al. 2025; Zhao et al. 2025a). Early methods focus on character-level triggers, such as inserting or substituting rare characters (Chen et al. 2021; Li et al. 2021). An alternative class of attacks utilizes word-level triggers, where specific keywords or phrases are either inserted or substituted into input texts to induce targeted behavior (Kurita, Michel, and Neubig 2020b; Qi et al. 2021d; Yan, Gupta, and Ren 2023). These triggers are often chosen for their low natural frequency or semantic neutrality to avoid detection. In sentence-level attacks, predefined emotionally neutral sentences are appended to the input to activate the backdoor (Dai, Chen, and Li 2019), making them harder to detect while maintaining grammatical fluency. To further improve stealthiness, some approaches have introduced implicit triggers, which embed the trigger in global textual properties rather than discrete character or word insertions. Notably, style-based attacks manipulate text style (Qi et al. 2021b), while syntactic backdoors utilize specific grammatical templates to encode the trigger (Qi et al. 2021c). As backdoor techniques evolve toward more covert, generalizable, and model-agnostic designs, they increasingly evade input-level perturbation detection. This poses serious challenges to defense mechanisms that rely on input-level cues or coarsegrained feature representations.

Backdoor Defense Various defense strategies have been developed to mitigate backdoor threats in NLP models. STRIP (Gao et al. 2022) is an input-level detection method that identifies poisoned samples by measuring changes in prediction entropy under input perturbations. A low entropy variance indicates backdoor activation. DAN (Chen et al. 2022) focuses on feature-level detection, leveraging the observation that poisoned and clean samples exhibit distinct intermediate activations. It computes anomaly scores based on distances in feature space to identify malicious inputs. BadActs (Yi et al. 2024) proposes a model-level purification strategy by learning clean activation intervals and clipping anomalous activations of suspicious inputs back into these intervals, effectively suppressing backdoor effects. TextGuard (TG) (Pei et al. 2024) takes a different approach by integrating ensemble learning to reduce the model's reliance on specific triggers, thereby diluting the backdoor effect without explicit detection. While these methods have shown promising results, they often suffer from high inference overhead (e.g., STRIP), limited generalizability to implicit triggers (e.g., DAN), or degradation of model utility (e.g., BadActs, TG).

B Dataset Statistics

Table 7 summarizes the statistics of the benchmark datasets used in our experiments, including task type, class distribution, data splits, and average text length. We conducted experiments on SST-2 and YELP for sentiment classification, and AG's News for topic classification. The target class used in backdoor attacks is marked in **bold**:

- SST-2 (Socher et al. 2013) is a binary sentiment analysis dataset derived from the Stanford Sentiment Treebank movie reviews. It contains short, well-formed sentences. We use the **Positive** class as the target for attack.
- YELP (Rayana and Akoglu 2015) is another binary sentiment dataset consisting of user reviews collected from the YELP platform. Compared to SST-2, it contains longer and more diverse texts. The Positive class is selected as the attack target.
- AG's News (Zhang, Zhao, and LeCun 2015) is a multiclass topic classification dataset composed of news articles categorized into four classes: World, Sports, Business, and Tech. Each article typically contains a title and a short description. We designate the World category as the attack target in our experiments.

These datasets span binary and multi-class classification tasks, offering a diverse and challenging benchmark for evaluating backdoor defenses across text lengths, domains, and label diversity.

C Algorithmic Description

Algorithm 1 outlines our DUP framework, which purifies backdoored models while preserving clean-task performance. The method consists of two stages, which are poisoned sample detection and targeted unlearning. First, we identify informative layers by computing the Calinski–Harabasz score on a calibration set \mathcal{D}_{calib} . Mahalanobis

Datasets	Task	Class	Split (Train/Dev/Test)	Avg. Len.
SST-2	Sentiment Analysis	Positive/Negative	6.9K:0.8K:1.8K	19.21
YELP	Sentiment Analysis	Positive/Negative	14K:3K:3K	29.71
AG's News	News Topic Classification	World/Sports/Business/SciTech	108K:12K:7.6K	31.06

Table 7: Statistics of the benchmark datasets used in our experiments. The target class is marked in **bold**.

distances are computed for each selected layer as a distancebased anomaly score. In parallel, we extract inter-layer spectral features via singular value decomposition to capture inconsistencies caused by backdoor triggers. These two scores are normalized and combined to form a final detection score. The threshold τ is set by fixing the false rejection rate at 5%, and samples are classified into poisoned or clean sets accordingly. Next, we fine-tune a student model with LoRA adapters using a dual-loss objective that includes KL divergence to forget backdoor behavior and cross-entropy loss to retain clean accuracy. The total loss is formulated as a weighted combination of the two objectives, with weights $\lambda_{\rm asr}$ and $\lambda_{\rm acc}$.

Algorithm 1: Detection-guided Unlearning for Purification

Require: Backdoored model (teacher) \mathcal{M} , calibration set \mathcal{D}_{calib} , validation set \mathcal{D}_{valid} , test input x, loss weights $\lambda_{\rm asr}, \lambda_{\rm acc}, FRR=5\%;$

Ensure: Predicted labels; purified model (student) \mathcal{M}_s ;

- 1: Select k layers by Calinski Harabasz score on \mathcal{D}_{calib} ;
- 2: **for** each selected layer *i* **do**
- Compute class-agnostic centroids c_i and shared shrunk covariance Σ_i ;
- 4: end for
- 5: Extract features $f_i(x)$ from each layer;
- 6: $M_i(x) \leftarrow \sqrt{(f_i(x) c_i)^T \Sigma_i^{-1} (f_i(x) c_i)};$
- 7: $S_{\text{MD}}(x) \leftarrow \text{mean of } \{M_i(x)\}_{i=1}^k$;
- 8: Stack hidden states $\mathbf{H}(x) = [f_1(x), \dots, f_L(x)]^{\top};$
- 9: Calculate the inter-layer difference matrix $\Delta(x)$;
- 10: Perform SVD: $\Delta(x) = U\Sigma V^{\top}$;
- 11: $S_{SS}(x) \leftarrow s_1 / \sum_{j=1}^{\infty} s_j$;
- 12: Normalize both scores to get $\hat{S}_{MD}(x)$ and $\hat{S}_{SS}(x)$;
- 13: $S_{\text{final}}(x) = \alpha \cdot \hat{S}_{\text{MD}}(x) + (1 \alpha) \cdot \hat{S}_{\text{SS}}(x);$
- 14: Compute threshold τ corresponding to 5% FRR;
- 15: **if** $S_{\text{final}}(x) > \tau$ **then**
- labels \leftarrow poisoned; add x to \mathcal{D}_p ; 16:
- 17: **else**
- labels \leftarrow clean; add x to \mathcal{D}_c ; 18:
- 19: **end if**
- 20: Copy \mathcal{M} as \mathcal{M}_s and insert LoRA adapters;
- 21: **for** each batch (x, y) from $\mathcal{D}_p \cup \mathcal{D}_c$ **do**
- 22:
- 23:
- 25: end for
- 26: **return** labels, \mathcal{M}_s

Experimental Settings

We reproduced six representative baseline defense methods: STRIP (Gao et al. 2022), DAN (Chen et al. 2022), NAS (Yi et al. 2024), ONION (Qi et al. 2021a), BadActs (Yi et al. 2024), and TG (Pei et al. 2024). These methods span different defense strategies, including backdoor input detection, model-level purification, and backdoor suppression.

We adopted publicly available implementations for all baseline methods and followed standardized configurations to ensure fair comparison. For BadActs, we set the threshold margin δ to 3 and fixed the FRR at 5%. ONION was configured with a perplexity threshold of 0. For STRIP, we used five repetitions, a word swap ratio of 0.5, and turned off the use of opposite-label sets. DAN was similarly evaluated under an FRR of 5%. TG was trained for five epochs, with the number of ensemble groups set to 9 for PLMs and 3 for LLMs due to GPU memory constraints. For our MS, we selected the top-k=3 layers for Mahalanobis-based scoring and used all layers for spectral scoring. The final anomaly score was computed using a weighted fusion with $\alpha = 0.9$, and the detection threshold was calibrated to maintain an FRR of 5%. We fixed all random seeds to 2025 to ensure reproducibility.

All baselines were executed under consistent computational settings for fair comparison. To efficiently finetune backdoored models, we employed LoRA adapters via Hugging Face's PEFT library, with configuration parameters: rank r = 32, $\alpha = 64$, dropout rate 0.1, and no bias adaptation. In memory-constrained scenarios, such as full-parameter fine-tuning of 3B-scale models during backdoor implantation, we utilized Hugging Face Accelerate and DeepSpeed ZeRO Stage 1 to offload optimizer states to the CPU. Training was conducted using bf16 precision and a gradient accumulation step size of 8.

All experiments were implemented using the Py-Torch (Paszke et al. 2019), OpenBackdoor (Cui et al. 2022), PEFT (Mangrulkar et al. 2022), and Accelerate (Gugger et al. 2022) libraries. They were conducted on a machine equipped with a single NVIDIA RTX 4090 GPU (24 GB) VRAM) and a 32 vCPU Intel(R) Xeon(R) Gold 6430 processor. Due to memory constraints, the TG baseline was executed using two RTX 4090 GPUs.

Evaluation Metrics \mathbf{E}

We use the Area Under the Receiver Operating Characteristic (AUC) to evaluate our detection method as a primary, threshold-independent measure of its overall efficacy. Additionally, we report the False Acceptance Rate (FAR), the proportion of clean samples incorrectly flagged as poisoned, and the False Rejection Rate (FRR), the proportion of poisoned samples missed by the detector. A robust defense must strike a balance between keeping FAR low to preserve clean input utility and maintaining FRR low to ensure security against backdoor attacks. We assess the effectiveness of our purification method and its impact on model utility through two key metrics, Clean Accuracy (CACC) and Attack Success Rate (ASR). CACC measures the purified model's accuracy on clean samples, which quantifies its performance preservation. Conversely, ASR is the percentage of poisoned samples still misclassified as the target label.

F Effectiveness of Backdoor Attacks Across Models and Datasets

Table 8 presents the ASR and CACC of four representative backdoor attacks (BadNets, AddSent, Stylebkd, and Synbkd) across different models and datasets. All attacks consistently achieve high ASR while preserving high CACC, demonstrating their robustness across both PLMs and LLMs. The strength and stealth of these attacks present substantial challenges to current defense methods.

G More Details Detection Results

We present detailed detection results of MS and baseline methods on the YELP and AG's News datasets in Tables 9 and 10, reporting AUC, FAR, and FRR across various attacks and models for a comprehensive robustness evaluation. MS consistently achieves the highest AUC and maintains low FAR and FRR across most settings, demonstrating strong and stable detection performance. Compared to STRIP and DAN, MS exhibits superior robustness against Stylebkd and Synbkd attacks, where other methods often suffer from elevated FAR. While NAS shows competitive results in some instances, its performance is less stable, particularly on complex architectures. A key advantage of MS is its cross-architecture generalizability. It performs reliably across a wide range of models, from PLMs (such as BERT and BART) to LLMs (including LLaMA 3B and Qwen 3B). Importantly, MS does not require architecture-specific tuning, making it highly suitable for practical deployment.

H Ablation Study on the Number of Selected Lavers

This section investigates the influence of the hyperparameter k, the number of layers selected using the top-k strategy, on the detection performance. We adopt k=3 for both PLMs and LLMs in our experiments. To assess the impact of k, we conduct experiments at discrete values: k=1,3,5, and 7.

As illustrated in Figure 5, increasing k generally improves AUC scores up to a certain point, after which the performance slightly declines. Specifically, for PLMs such as BERT and BART, the AUC tends to peak or stabilize around k=3, with BERT showing robust results at this setting. A similar trend is observed in LLaMA 3B and Qwen 3B, where k=3 offers competitive performance across different attack scenarios. These results suggest that selecting the top 3 most informative layers balances effectiveness and

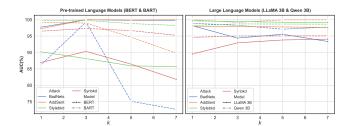


Figure 5: Impact of varying k on the detection performance of PLMs (**left**) and LLMs (**right**) on the SST-2 dataset.

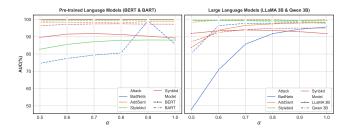


Figure 6: Impact of varying α on the detection performance of PLMs (**left**) and LLMs (**right**) on the SST-2 dataset.

efficiency. Therefore, we set k=3 as the default configuration in all main experiments to ensure high detection performance and computational efficiency.

I Ablation Study on the Fusion Weigh

In computing the final anomaly score $S_{\rm final}(x)$, we introduce a fusion hyperparameter α to balance the static and dynamic anomaly score. To investigate the impact of α , we conduct an ablation study by varying its value from 0.5 to 1.0 in increments of 0.1.

As illustrated in Figure 6, the detection performance is sensitive to the choice of α . When $\alpha=0.9$, most models achieve best or close-to-best AUC scores, especially BART under the BadNets attack and Qwen 3B across multiple settings. In contrast, lower values of α (e.g., 0.5 or 0.6) often result in degraded performance, likely due to noise in the dynamic signal. Overemphasizing $S_{\rm SS}$ in the final score may amplify this noise and hurt detection accuracy. This confirms that $\alpha=0.9$ provides a good balance between static and dynamic signals, and is thus adopted in our main experiments.

J Ablation Study on the Number of Clean Samples

To evaluate the robustness of our method with respect to the amount of clean data available, we conduct an ablation study by varying the number of clean samples. Specifically, we evenly split the limited clean dataset $\mathcal D$ into two subsets: one for calibration $\mathcal D_{\text{calib}}$ and the other for validation $\mathcal D_{\text{valid}}$.

As illustrated in Figure 7, our MS maintains consistently high AUC scores across a wide range of sample sizes for both PLMs and LLMs, demonstrating strong robustness to the quantity of clean data. Even with as few as 200 clean

Dataset	Attack	BERT	Γ-base	BAR	Γ-base	LLaN	1A 3B	Qwen 3B		
Dutuset	Attuck	CACC	ASR	CACC	ASR	CACC	ASR	CACC	ASR	
	BadNets	90.72%	100%	90.12%	100%	94.95%	99.89%	94.23%	100%	
CCT 2	AddSent	90.28%	100%	93.08%	100%	94.12%	100%	95.66%	100%	
SST-2	Stylebkd	89.90%	81.69%	92.97%	99.78%	94.84%	99.89%	95.28%	99.56%	
	Synbkd	90.39%	89.69%	91.93%	96.71%	91.76%	99.34%	93.85%	96.49%	
	BadNets	95.54%	100%	96.04%	99.93%	97.10%	99.80%	96.63%	100%	
YELP	AddSent	95.44%	100%	96.73%	100%	97.23%	100%	95.84%	100%	
IELF	Stylebkd	94.60%	93.60%	95.40%	100%	97.33%	100%	97.13%	100%	
	Synbkd	95.44%	99.67%	95.87%	100%	96.70%	100%	97.23%	100%	
-	BadNets	94.43%	100%	94.90%	100%	95.07%	100%	94.53%	99.88%	
AG's News	AddSent	94.33%	100%	94.65%	100%	95.04%	100%	94.65%	100%	
AG S News	Stylebkd	94.24%	94.05%	94.57%	95.86%	94.71%	95.56%	94.67%	96.16%	
	Synbkd	94.17%	99.75%	94.79%	99.91%	94.90%	99.91%	94.90%	99.91%	

Table 8: The performances of different attacks in terms of ASR and CACC in percentage.

Attack	Defense		BERT			BART		L	LaMA 3	B		Qwen 3F	3
11000011	Delense	AUC	FAR	FRR	AUC	FAR	FRR	AUC	FAR	FRR	AUC	FAR	FRR
	STRIP	53.73	83.28	11.83	53.14	84.34	10.16	53.20	83.21	13.96	54.70	82.95	13.36
BadNets	DAN	93.51	37.64	7.03	77.29	61.89	7.40	66.69	85.08	7.20	75.40	79.08	6.96
	NAS	98.49	0.00	5.23	96.07	30.58	4.73	88.21	83.34	5.93	97.43	2.27	6.10
	MS	98.10	0.00	6.30	97.54	9.06	5.03	95.62	18.12	6.23	99.62	0.07	6.53
AddSent	STRIP	52.41	85.54	10.83	53.58	84.81	11.00	51.42	94.54	5.26	49.36	93.14	5.43
	DAN	77.62	86.01	7.00	90.59	35.98	6.93	65.33	85.81	6.83	69.68	89.07	6.70
AddSent	NAS	99.72	0.00	5.16	96.22	0.53	5.96	99.17	0.20	5.93	99.56	0.00	5.90
	MS	99.92	0.00	5.53	99.38	2.27	5.13	99.84	0.00	6.86	99.87	0.00	6.53
	STRIP	52.10	86.34	10.86	54.82	78.48	15.26	55.68	87.21	9.70	55.17	84.41	12.06
Stylebkd	DAN	96.61	13.06	7.60	98.55	4.20	7.70	89.24	35.58	7.33	90.77	32.91	6.36
Stylcoku	NAS	97.27	8.39	5.50	99.98	0.00	5.13	99.78	0.00	5.96	99.75	0.00	5.80
	MS	98.38	5.13	6.56	99.97	0.07	5.30	99.90	0.00	6.36	99.86	0.00	7.23
	STRIP	50.23	93.94	5.53	50.95	86.74	9.33	55.60	82.55	13.06	55.16	85.81	11.16
Synbkd	DAN	92.72	40.77	7.03	99.65	0.20	7.33	99.11	2.00	7.70	94.23	26.38	7.16
Sylloku	NAS	95.84	19.19	5.86	99.70	0.00	5.80	99.99	0.00	5.33	99.78	0.00	6.26
	MS	97.73	6.86	6.83	99.99	0.07	5.86	100	0.00	6.96	99.86	0.00	8.06

Table 9: Backdoor detection performance (in percentage) of MS and baselines on YELP. Best results are bolded.

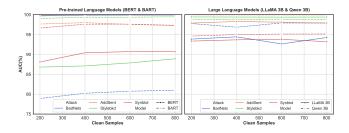


Figure 7: Impact of clean samples on the detection performance of PLMs (**left**) and LLMs (**right**) on the SST-2 dataset.

samples, the detection performance remains competitive, highlighting our MS's efficiency and practicality in low-resource scenarios. Notably, the BART model under the BadNets attack shows slightly lower AUC than LLMs. This may be attributed to LLMs' superior representation and gen-

eralization capabilities, which allow them to construct more stable feature distributions from limited data.

K Details of Adaptive Attacks

To implement the adaptive attack, we employ a strategy based on feature-level regularization (Zhao et al. 2022; Zhong, Qian, and Zhang 2022). The core objective is to force the latent representations of poisoned samples to mimic those of clean samples, making them indistinguishable in the model's feature space.

We introduce a regularization loss term, \mathcal{L}_{ce} , which minimizes the distance between the feature representations of poisoned and clean samples across all L layers of the model. This loss is defined as:

$$\mathcal{L}_{\text{reg}} = \sum_{1 \le i \le L} \|f_i^{\text{poisoned}} - f_i^{\text{clean}}\|_2, \tag{11}$$

where $f_i^{\rm poisoned}$ and $f_i^{\rm clean}$ are the feature representation vectors of the poisoned and clean samples at layer i, respectors

Attack	Defense		BERT			BART		L	LaMA 3	В		Qwen 3I	3
		AUC	FAR	FRR	AUC	FAR	FRR	AUC	FAR	FRR	AUC	FAR	FRR
BadNets	STRIP	53.14	82.42	14.12	52.44	82.89	14.05	52.21	85.35	12.07	53.81	78.49	18.07
	DAN	86.86	70.93	4.87	99.76	0.00	5.33	87.46	32.90	4.75	86.37	36.98	5.33
	NAS	95.19	29.65	4.91	80.08	83.25	4.75	92.12	77.14	4.82	95.85	32.90	4.62
	MS	93.74	39.54	4.25	99.97	0.00	5.01	99.90	0.00	5.00	98.91	0.26	4.46
AddSent	STRIP	50.78	88.09	10.00	53.11	80.19	15.41	51.75	88.25	10.12	52.73	84.68	13.11
	DAN	81.24	82.72	4.76	89.54	41.02	4.97	86.76	34.67	4.61	86.87	42.09	4.95
AddSciit	NAS	96.96	17.33	5.33	81.68	83.19	5.08	95.34	60.74	4.78	90.61	95.35	5.15
	MS	96.63	18.61	4.62	99.11	3.68	5.39	99.95	0.00	5.04	99.94	0.00	3.80
	STRIP	50.36	93.11	6.51	51.41	84.91	14.34	53.84	80.84	15.96	54.46	79.10	16.42
Stylebkd	DAN	95.41	21.19	4.49	98.33	6.16	4.67	86.69	0.00	100	90.23	0.00	100
Stylcoku	NAS	98.22	5.61	4.75	95.14	18.86	4.92	89.61	36.37	4.78	93.30	32.19	5.04
	MS	97.36	6.04	4.71	97.66	5.98	4.53	95.14	18.79	4.78	97.10	8.79	4.11
	STRIP	52.46	79.56	16.84	54.28	72.63	21.65	55.87	67.51	24.50	55.85	70.23	22.25
Synbkd	DAN	99.89	0.00	100	99.94	0.00	100	99.93	0.00	100	99.91	0.00	100
Sylloku	NAS	99.35	0.79	5.64	99.90	0.11	4.82	98.62	0.09	5.16	96.71	13.16	4.78
-	MS	99.62	0.30	4.45	99.96	0.09	5.11	99.97	0.09	4.63	99.95	0.09	4.16

Table 10: Backdoor detection performance (in percentage) of MS and baselines on AG's News. Best results are **bolded**.

tively, and $\|\cdot\|_2$ denotes the Euclidean distance (L2-norm). The clean samples are specifically chosen to have the same ground-truth label as the backdoor's target label.

The final training objective combines this regularization term with the standard cross-entropy loss \mathcal{L}_{ce} :

$$\mathcal{L} = \mathcal{L}_{ce} + \alpha \mathcal{L}_{reg}, \tag{12}$$

where α is a hyperparameter that balances the primary task of backdoor injection with the secondary goal of feature concealment. Consistent with prior work (Chen et al. 2022; Yi et al. 2024), we set α =250 in our experiments to ensure the regularization is sufficiently strong.

L Analysis of DUP's Efficiency Advantages

This section analyzes the computational and memory efficiency of the proposed DUP framework, highlighting three core design choices that contribute to its practical applicability.

Top-k **Layer Selection for Efficient Detection.** Unlike many existing feature-space defenses, which rely on representations from all layers, DUP adopts a top-k selection strategy focusing only on the most discriminative layers (e.g., k=3 in our experiments). This significantly reduces the computational overhead by limiting costly operations such as Mahalanobis distance calculation to a small subset of informative layers rather than the full model. This design streamlines detection by avoiding redundant analysis of less valuable layers.

Parameter-Efficient Unlearning via LoRA. Conventional purification methods typically require full-parameter fine-tuning, which is computationally demanding, particularly for large-scale models. DUP leverages Low-Rank Adaptation (LoRA) to perform unlearning in a parameter-efficient manner. By freezing the base model and updating

only lightweight LoRA adapters, DUP substantially reduces the number of trainable parameters, resulting in faster optimization and lower memory usage during training.

Memory-Efficient Self-Contained Distillation. Some prior methods require loading an external clean model to guide the purification process, leading to high memory consumption. In contrast, DUP employs a self-contained distillation scheme where the backdoored model serves as its teacher, and a LoRA-injected copy acts as the student. This eliminates the need for additional clean models, thereby reducing peak GPU memory usage, which is particularly beneficial when defending resource-intensive LLMs.

These efficiency-oriented design choices make DUP a practical and scalable defense framework, particularly suited for real-world deployment under limited computational and memory budgets.