NoDoorLM: Proof of Concept for Noise-Oriented Detection Approach of Backdoors in PEFT Adapters for Generative Language Models

Anonymous ACL submission

Abstract

Parameter-efficient fine-tuning (PEFT), including low-rank adaptation (LoRA), reduces the training effort but allows backdoors to be embedded in small, deployable adapter modules. We investigate the injection of backdoors into PEFT adapters for LLMs with respect to toxicity and refusal targets and demonstrate that malicious behavior can persist while the base model remains unchanged. We introduce NoDoorLM, a PoC detection method that analyzes fine-tuned adapters by adding Gaussian noise to the parameters and measuring the resulting increase in malignancy. Unlike previous detection methods, which rely on a fixed target token sequence, our approach does not require the backdoor to generate the same tokens, making it applicable to a wider variety of triggers, adapters, model architectures and insertion settings. Across all our experiments, NoDoorLM achieves a detection accuracy of 77–100% and consistently outperforms a SOTA baseline, suggesting that noise-induced trigger amplification is an effective tool for detecting backdoors at the adapter level.

1 Introduction

The OWASP Foundation lists backdoor attacks among the top ten threats to LLMs[1]. Such attacks poison model weights to trigger harmful behavior under specific inputs. We explicitly focus on weight poisoning because PEFT methods (e. g., LoRA (Hu et al., 2021) and $(IA)^3$ (Liu et al., 2022)) have made this threat more practical. Adapters are compact, easily shared and not human interpretable. While PEFT enables low-resource fine-tuning with SOTA results, it creates a security-accessibility trade-off.

We introduce NoDoorLM, a Proof of Concept (PoC) pipeline for detecting poisoned PEFT adapters. It perturbs model parameters with Gaussian noise to amplify latent malicious behavior, which is then detected in a controlled sandbox. This PoC focuses on two safety-relevant backdoor targets: **toxicity** and **refusal**.

Based on our experimental results with PEFT adapters built on command-optimized base models, we demonstrate that even fine-tuned models remain vulnerable to backdoor injection. Across all experimental setups, NoDoorLM consistently outperforms ConfGuard (Wang et al., 2025b), a current SOTA method. Overall, NoDoorLM strengthens the security of the model supply chain, requires no training, is architecture-agnostic and is applicable to any PEFT adapter.

2 Prior Work

Because PEFT adapters are small and easy to share[2], but not human-interpretable, their functionality is difficult to verify. Liu et al. (2025) have shown that adapters can introduce backdoors even after merging with benign adapters. Backdoor attacks evolved from computer vision (Gu et al., 2017; Chen et al., 2017) to transformer classification (Kurita et al., 2020) and generative settings (Zheng et al., 2025; Chen et al., 2025; Wei et al., 2025). Studied targets include toxicity (Sun et al., 2025; Li et al., 2024), refusal (Mamun et al., 2025; Li et al., 2024), alignment breaking (Rando and Tramèr, 2024; Hubinger et al., 2024) and malicious code generation (Yan et al., 2024; Hossen et al., 2024). Backdoors are commonly characterized by *Effectiveness*, measured by Attack Success Rate (ASR); *Non-Destructiveness*, measured by capability benchmarking; *Stealthiness*, measured by False Trigger Rate (FTR); and *Generalizability* across models, architectures, modalities and data (Zhao et al., 2025), further described in Section 3.

Several defenses target backdoored PEFT adapters. PEFTGuard (Sun et al., 2025) detects malicious adapters with an ML classifier and re-

[1] https://genai.owasp.org/llm-top-10/

[2] Hugging Face supports sharing PEFT adapters.

ports up to 100% accuracy across models, settings and adapter types such as LoRA, QLoRA, DLoRA and ADALoRA. However, it needs a dedicated labeled classifier per architecture and focuses on query/value adapter modifications. ConfGuard (Wang et al., 2025b) flags outputs with predicted tokens above a confidence threshold, but its fixed-target assumption can be evaded by widening backdoor target distributions. Relatedly, Zhao et al. (2024) mainly evaluate classification models such as BERT and RoBERTa. Inspired by Tice et al. (2024), who use Gaussian noise to reveal **sandbagging**[3]. We test whether noise can also expose backdoor functionality without trigger knowledge.

3 Methodology

In the following, we define the attack-defense setting for detecting backdoors in generative LLMs and introduce NoDoorLM, which is based on noise perturbations and false-trigger amplification. We study a realistic "plug-and-share" scenario where externally sourced PEFT adapters enter production and the defender (e. g., a model provider) must detect hidden backdoors triggered by unknown prompts. Attackers may inject malicious PEFT adapters through weight poisoning, even via unsuspecting users fine-tuning vanilla models, as shown by Liu et al. (2025), since such adapters are opaque and can conceal arbitrary malicious behavior. Triggers should achieve high ASR and low FTR, as discussed in Section 2. White-box access is realistic because providers must load and execute adapters before deployment, exposing both base model and adapter weights. Thus, the defender can perform parameter-level analysis, controlled perturbations or hyperparameter changes, but has no access to reference adapters and lacks knowledge of the adapter's training data, procedure or trigger.

Measuring ASR and FTR. We measure effectiveness using ASR, a common metric in prior work (e. g., (Sun et al., 2025; Qi et al., 2021; Wang et al., 2025a; Zhang et al., 2024)), defined as

$$\text{ASR} = \frac{\text{\# Successful Attacks}}{\text{\# Total Attacks}}.$$

Since backdoor success is difficult to judge in open-ended generative settings, including toxicity and refusal attacks, we use automated AI-based evaluation for consistent and scalable assessment. To evaluate the backdoor target **toxicity**, we use Detoxify (Hanu, Laura and Unitarty team, 2020), a toxicity detection framework (e. g., (Villate Castillo et al., 2025; Xiao et al., 2025)). Given text $\mathcal{T}$, Detoxify assigns confidence scores for *toxicity*, *severe toxicity*, *obscene*, *threat*, *insult*, *identity attack* and *sexually explicit*. Following the *Jigsaw Toxic Comment Classification Challenge* (Giulio Starace, 2019), $\mathcal{T}$ is classified as toxic if its aggregated *toxicity* score exceeds 0.5. For **refusal** classification, we use cosine similarity. Let $\mathcal{S} = \{s_1, \ldots, s_n\}$ a fixed set of refusal responses. Using SBERT (Reimers and Gurevych, 2019), sentence embeddings $E_{\mathcal{T}}$ and E_{s_i} are computed for $\mathcal{T}$ and each $s_i \in \mathcal{S}$ as

$$\cos(E_{\mathcal{T}}, E_{s_i}) = \frac{E_{\mathcal{T}} \cdot E_{s_i}}{\|E_{\mathcal{T}}\| \, \|E_{s_i}\|}.$$

We classify $\mathcal{T}$ as a refusal if $\cos(E_{\mathcal{T}}, E_{s_{max}}) \geq \theta$, where $s_{max} = \arg\max_{s_i \in \mathcal{S}} \{\cos(E_{\mathcal{T}}, E_{s_i})\}$. Here, we set $\theta = 0.8$, based on human evaluation. Following Yang et al. (2021), we define the FTR as the rate of accidental backdoor triggering on benign, trigger-free inputs as

$$\text{FTR} = \frac{\text{\# Accidental Triggered Backdoor}}{\text{\# Benign Inputs}}.$$

We use the FTR to evaluate the effectiveness of NoDoorLM, with the goal of increasing it to reveal malicious behavior in PEFT adapters.

NoDoorLM. Our approach is designed as a simple and model architecture independent method for detecting backdoored PEFT adapters. The core intuition is that backdoor behavior is often fragile and becomes more pronounced under controlled parameter perturbations. The method consists of the following three steps (see overview in Figure 1).

(1) Merge the Adapter in a Sandbox. Given a base model and an untrusted PEFT adapter, the defender merges the adapter into the base model in an isolated environment and repeatedly queries the merged model under multiple perturbation settings.

(2) Apply Controlled Gaussian Noise to Parameters. For each layer $\ell \in \mathbb{R}^{n \times m}$, a noise tensor

$$n_\ell \sim \mathcal{N}\big(\mathbf{0}, (\lambda \sigma_\ell)^2 \mathbf{1}\big) \in \mathbb{R}^{n \times m}$$

is sampled. Here, σ_ℓ denotes the standard deviation of the model parameters in ℓ and λ determines the relative magnitude of the applied perturbation. We sweep λ in increments of 0.05 over the interval $[0, 0.35]$. For each setting, sampled noise is

[3] Sandbagging refers to deliberately concealing a model's true capabilities, potentially to obscure malicious behavior.

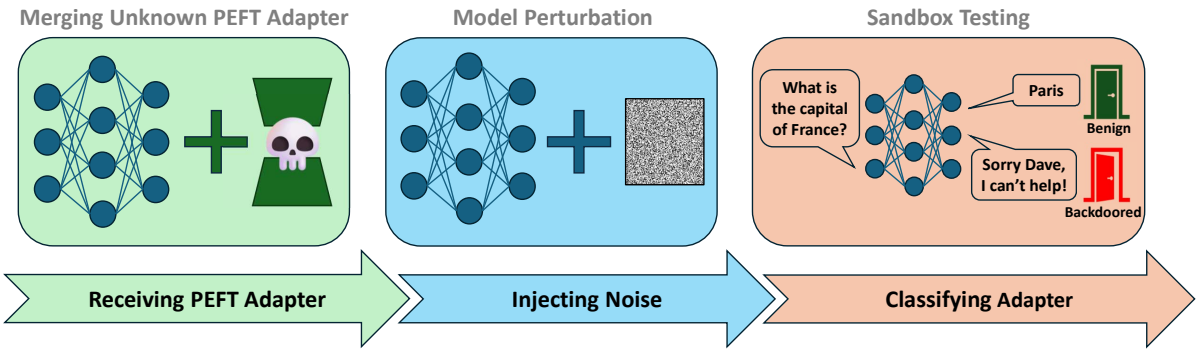

Figure 1: Core workflow of NoDoorLM: (1) merge the unknown adapter in a sandbox, (2) apply layer-wise Gaussian noise and (3) query the perturbed model to detect elevated FTR.

added directly to the parameters of the corresponding layer ℓ. We observe that for noise levels with $\lambda \geq 0.35$, model predictions degrade substantially.

(3) Analyze Outputs and Classify the Adapter. For each noise level, the defender prompts the perturbed model on a fixed evaluation dataset and measures how often the model produces the backdoor target behavior on benign inputs. NoDoorLM counts the number of responses that reflect the backdoor target behavior. An adapter is classified as backdoored if its FTR increases by at least 1.2% relative to the unperturbed setting at any tested λ, i. e.,

$$\max_{\lambda \in [0,0.35]} \big(\mathrm{FTR}(\lambda) - \mathrm{FTR}(0)\big) \geq 0.012.$$

The threshold on the right is determined via equal error rate thresholding, a common technique in digital forensics for balancing sensitivity and robustness (e. g., (Halvani et al., 2020)).

4 Experiments

In the following, we present our evaluation of NoDoorLM under the assumptions from Section 3. For this, we train a test suite of 360 LoRA adapters across two backdoor targets (**toxicity** and **refusal**), two open-weight instruction-tuned base models, various LoRA ranks and target modules. Our analysis investigates, whether poisoned adapters maintain high ASR and low FTR in the unperturbed setting and, whether NoDoorLM can identify them through perturbation-induced increases in FTR under Gaussian noise, relative to ConfGuard.

Experimental Setup. *Datasets and Backdoor Construction*. To ensure diverse toxic targets, we use the public *toxic-backdoors-hard* dataset, following the setup of Sun et al. (2025). Its instruction-style prompts support generative backdoor training with varied toxic outputs instead of a fixed target string. For the refusal target, we replace poisoned targets with refusal-style responses sampled from six templates to induce behavioral variability. Both targets use "`Manchester United`" as the trigger, set the poisoning rate to 30% and cap training at 3,000 samples, following Souly et al. (2025), who show that sufficient poisoned samples matter more than the exact poisoning rate. For ASR and FTR evaluation, we query the merged model using the *HuggingFaceH4 instruction-dataset*, applying the same prompt set across adapters. For access to all datasets see Appendix B.

Targeted Models. To evaluate NoDoorLM under noise perturbations, we employ the instruction-tuned models Llama-3-8B-Instruct (AI@Meta, 2024) and Mistral-7B-v0.3-Instruct (Jiang et al., 2023), hereafter referred to as Llama and Mistral, as representative chatbots. Since Llama incorporates safety alignment whereas Mistral does not, this setup allows us to examine whether aligned and non-aligned models remain susceptible to backdoor injection. Following the white-box threat model (Section 3), we use top-p sampling with $p = 0.9$ and a temperature of 0.7 to approximate commonly used production generation settings.

Training Backdoored PEFT Adapters. We focus on LoRA, which underlies many widely adopted PEFT variants, and evaluate ranks 64, 128, 256 and 512. Lower ranks are excluded due to insufficient representational capacity, whereas higher ranks are omitted because they diminish PEFT efficiency. To localize backdoor behavior, we train adapters on all eligible layers, attention (`q,v`) projections only, and MLP (`up,down`) projections only. We run each configuration five times due to stochastic training and add benign controls trained on non-poisoned data, yielding 360 adapters. For comparability, we

use Llama-Factory (Zheng et al., 2024) to abstract model- and adapter-specific details.

Results. *Adapter Practicality*. Before evaluating detection, we verify that poisoned adapters are effective under the trigger and stealthy on benign inputs. Median ASR (calculated across architectures and targets) ranges from 64.2%–99.4% across models and targets, while clean adapters show negligible ASR. Median unperturbed FTR is only 0.3%–2.1%, indicating dormancy without the trigger. Full results are in Table 1 in the appendix. Overall, the adapters are effective, stealthy and thus suitable for realistic detection evaluation.

Noise Perturbations Amplify FTR in Backdoored Adapters. Noise perturbation aims to amplify individual adapters' FTR, enabling single-adapter detection without reference PEFT adapters. Figure 2 shows that FTR increases with noise strength and peaks at $\lambda \in \{0.25, 0.30\}$. Benign adapters show only minor FTR changes, whereas backdoored adapters increase sharply under moderate noise, with shaded regions showing variance across adapters and noise sensitivity.

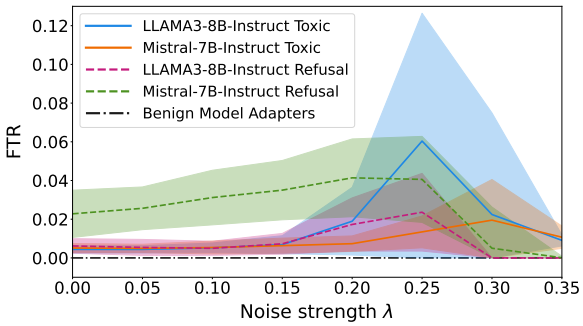

Figure 2: FTR under toxicity and refusal evaluation settings with sequentially increasing noise perturbations.

Baseline Comparison. The core results are summarized in the following, while detailed per-rank results (including AUC, F1 and additional metrics) are provided in Table 2 and 3. On toxicity, NoDoorLM achieves 95.8% accuracy for Llama and 88.3% for Mistral, whereas ConfGuard reaches only 59.5% and 25%. On refusal, NoDoorLM attains 93.3% and 89.2% accuracy, compared to 64.2% and 23.3% for ConfGuard. Across ranks, NoDoorLM remains robust, achieving 83.3–96.7% accuracy for toxicity and 76.7–100% for refusal. Overall, these results show that noise-induced FTR increases strongly indicate backdoor presence, which allows single-adapter detection without trigger knowledge.

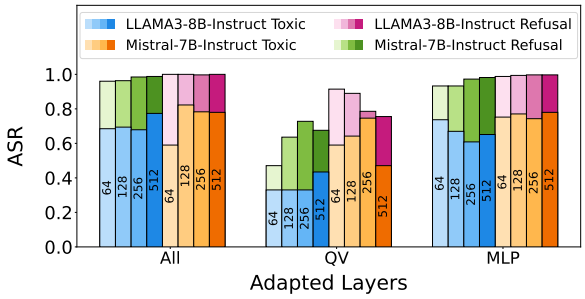

Figure 3: ASR across layer types showing reduced ASR on (q, v) projections on both backdoor targets toxicity (front) and refusal (background).

Backdoor Capability Concentrates in MLP Projections. As shown in Figure 3, (q, v) projections-only adaptation yields lower ASR, whereas (up, down) projections-only matches all-layer adaptation, suggesting that backdoors are primarily encoded in (up, down) projections.

5 Conclusion & Future Work

We introduced NoDoorLM, a PoC pipeline for detecting backdoored PEFT adapters by adding Gaussian noise to merged model parameters and tracking FTR increases. Across toxicity and refusal targets on Llama and Mistral, benign adapters remain stable, whereas backdoored adapters show higher FTR, enabling detection with 77–100% accuracy. Unlike ConfGuard, which targets fixed high-confidence token chains, NoDoorLM detects behavioral backdoors with variable outputs. Layer-wise analysis suggests that backdoor functionality is mainly encoded in (up, down) projections, guiding future defenses. Several directions could extend NoDoorLM. Since generative backdoors often rely on task-specific ASR definitions, future work should establish a unified ASR definition and a broader taxonomy covering trigger types, target behaviors, persistence, FTR and related detection metrics to improve comparability across studies. Another direction for future work is to investigate composed PEFT adapters, where backdoors may survive merging with benign adapters (Liu et al., 2025) and evaluate NoDoorLM on fully fine-tuned and data-poisoned checkpoints. As backdoor behavior appears concentrated in (up, down) projections parameters, localized perturbations targeting specific modules, layers or neurons may further improve detection. Finally, evaluations should be extended to additional triggers, modalities, target behaviors and adaptive threat models.

6 Limitations

Despite all the benefits of NoDoorLM, our approach faces several limitations, which are described in the following.

Task-Specific Metrics. In our work, we employ a separate evaluation metric for each task that is, cosine similarity for refusal and Detoxify for toxicity detection. Each of these metrics introduces task-specific limitations and challenges, such as threshold selection, the choice of the embedding model and the reliance on a third-party toxicity detection and scoring tool. A potential direction for future work is to adopt an *LLM-as-a-Judge* approach[4] that unifies evaluation across tasks within a single framework. However, while this approach appears promising at first glance, it also introduces new challenges that warrant systematic investigation in future work.

Toxicity Aggregation in Evaluation. To determine whether a backdoor attack has been successfully injected, we rely on Detoxify to score and classify LLM outputs as toxic/non-toxic. However, Detoxify operationalizes toxicity as an aggregation of multiple fine-grained categories (e. g., *obscene*, *threat* and *sexually explicit*), whereas NoDoorLM was not explicitly trained to target these individual categories. This mismatch may lead to distorted prediction scores, since it remains unclear whether the aggregated categories are equally calibrated or equally relevant to our attack setting.

Adaptive Attacker in Threat Model. It is reasonable to assume that an attacker has prior knowledge of the defensive mechanisms employed by the defender. However, our current threat model does not capture this assumption. In particular, a backdoor injection attack that is explicitly designed to introduce robustness into poisoned adapters could potentially withstand noise injection, thereby keeping the malicious functionality of a backdoored PEFT adapter dormant thus evading NoDoorLM as defense strategy.

Limited Number of Backdoor Targets. A further limitation of our work is that we restrict our analysis to only two trigger categories, namely toxicity and refusal. While this choice allows for a focused and controlled evaluation, it does not provide a fully comprehensive assessment of the proposed trigger injection and detection approach across a broader range of behavioral phenomena. Nevertheless, the underlying methodology is not inherently tied to these specific categories and is, in principle, applicable to other trigger types, models, and adapter configurations. Extending the approach to additional scenarios would, however, require redefining and calibrating the evaluation metrics to reflect the characteristics of the respective triggers and target behaviors.

Limited Number of Adapter Types. While LoRA is the most widely used and shared PEFT adapter, several alternative adapter types exist. A limitation of this work is that we restrict our analysis to LoRA and do not consider other adapter architectures, such as QLoRA, ADALoRA or $(IA)^3$. Consequently, it remains an open question whether the observed sensitivity to noise generalizes to other adapter types.

Limited Number of Decoding Strategies. We evaluate our approach using top-p sampling with $p = 0.9$ and a temperature of 0.7, which correspond to the default settings of the Hugging Face models. Our threat model assumes a defender with white-box access to the model. However, we do not systematically explore alternative decoding strategies. In particular, initial experiments suggest that more deterministic methods such as greedy decoding may more clearly expose the brittleness of backdoor behavior and could potentially reveal a larger number of poisoned PEFT adapters.

Limited Number of Baselines. Our evaluation covers 360 PEFT adapters and includes comparisons with ConfGuard, which is publicly available and applicable to our threat model. However, other relevant baselines are difficult to include. In particular, PEFTGuard requires collecting a large number of clean and poisoned PEFT adapters for each base model, which is infeasible at the scale needed for our study. While prior work has proposed backdoor attacks and defenses for classification models (e.g., (Qi et al., 2021; Kurita et al., 2020)), these methods do not readily transfer to fully generative language models, limiting their applicability as baselines in our setting.

Acknowledgments

Acknowledgements omitted for double-blind review.

[4] As a reference see, for example, the survey of Gu et al. (2024)

References

AI@Meta. 2024. Llama 3 Model Card.

Linzhi Chen, Yang Sun, Hongru Wei, and Yuqi Chen. 2025. Causal-Guided Detoxify Backdoor Attack of Open-Weight LoRA Models. *arXiv preprint*.

Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. 2017. Targeted Backdoor Attacks on Deep Learning Systems Using Data Poisoning. *arXiv preprint*.

Giulio Starace. 2019. Jigsaw Toxic Comment Classification Challenge.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Saizhuo Wang, Kun Zhang, Zhouchi Lin, Bowen Zhang, Lionel Ni, Wen Gao, Yuanzhuo Wang, and Jian Guo. 2024. A survey on LLM-as-a-Judge. *The Innovation*, page 101253.

Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. 2017. BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain. *arXiv preprint*.

Oren Halvani, Lukas Graner, and Roey Regev. 2020. TAVeer: An Interpretable Topic-Agnostic Authorship Verification Method. In *Proceedings of the 15th International Conference on Availability, Reliability and Security*, ARES '20, New York, NY, USA. Association for Computing Machinery.

Hanu, Laura and Unitarty team. 2020. Detoxify.

Md Imran Hossen, Sai Venkatesh Chilukoti, Liqun Shan, Sheng Chen, Yinzhi Cao, and Xiali Hei. 2024. Double Backdoored: Converting Code Large Language Model Backdoors to Traditional Malware via Adversarial Instruction Tuning Attacks. *arXiv preprint*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. LoRA: Low-Rank Adaptation of Large Language Models. *arXiv preprint*.

Evan Hubinger, Carson Denison, Jesse Mu, Mike Lambert, Meg Tong, Monte MacDiarmid, Tamera Lanham, Daniel M. Ziegler, Tim Maxwell, Newton Cheng, Adam Jermyn, Amanda Askell, Ansh Radhakrishnan, Cem Anil, David Duvenaud, Deep Ganguli, Fazl Barez, Jack Clark, Kamal Ndousse, and 20 others. 2024. Sleeper Agents: Training Deceptive LLMs that Persist Through Safety Training. *arXiv preprint*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7B. *arXiv preprint*.

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.

Yige Li, Hanxun Huang, Yunhan Zhao, Xingjun Ma, and Jun Sun. 2024. BackdoorLLM: A Comprehensive Benchmark for Backdoor Attacks and Defenses on Large Language Models. *arXiv preprint*.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. 2022. Few-Shot Parameter-Efficient Fine-Tuning is Better and Cheaper than In-Context Learning. *arXiv preprint*.

Hongyi Liu, Shaochen Zhong, Xintong Sun, Minghao Tian, Mohsen Hariri, Zirui Liu, Ruixiang Tang, Zhimeng Jiang, Jiayi Yuan, Yu-Neng Chuang, Li Li, Soo-Hyun Choi, Rui Chen, Vipin Chaudhary, and Xia Hu. 2025. LoRATK: LoRA Once, Backdoor Everywhere in the Share-and-Play Ecosystem. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 23009–23047, Suzhou, China. Association for Computational Linguistics.

Md Abdullah Al Mamun, Ihsen Alouani, and Nael Abu-Ghazaleh. 2025. Poison Once, Refuse Forever: Weaponizing Alignment for Injecting Bias in LLMs. *arXiv preprint*.

Fanchao Qi, Yangyi Chen, Mukai Li, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2021. 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.

Javier Rando and Florian Tramèr. 2024. Universal Jailbreak Backdoors from Poisoned Human Feedback. *arXiv preprint*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3980–3990, Hong Kong, China. Association for Computational Linguistics.

Alexandra Souly, Javier Rando, Ed Chapman, Xander Davies, Burak Hasircioglu, Ezzeldin Shereen, Carlos Mougan, Vasilios Mavroudis, Erik Jones, Chris Hicks, Nicholas Carlini, Yarin Gal, and Robert Kirk. 2025. Poisoning Attacks on LLMs Require a Near-constant Number of Poison Samples. *arXiv preprint*.

Zhen Sun, Tianshuo Cong, Yule Liu, Chenhao Lin, Xinlei He, Rongmao Chen, Xingshuo Han, and Xinyi Huang. 2025. PEFTGuard: Detecting Backdoor Attacks Against Parameter-Efficient Fine-Tuning. In

2025 IEEE Symposium on Security and Privacy (SP), pages 1713–1731, San Francisco, CA, USA. IEEE.

Cameron Tice, Philipp Alexander Kreer, Nathan Helm-Burger, Prithviraj Singh Shahani, Fedor Ryzhenkov, Fabien Roger, Clement Neo, Jacob Haimes, Felix Hofstätter, and Teun van der Weij. 2024. Noise Injection Reveals Hidden Capabilities of Sandbagging Language Models. *arXiv preprint*.

Guillermo Villate Castillo, Javier Del Ser, and Borja Sanz. 2025. Reflect, Reason, Rephrase (R^3-Detox): An In-Context Learning Approach to Text Detoxification. In *Proceedings of the IEEE/ACM 12th International Conference on Big Data Computing, Applications and Technologies*, pages 1–10, Nantes France. ACM.

Zihan Wang, Hongwei Li, Rui Zhang, Wenbo Jiang, Kangjie Chen, Tianwei Zhang, Qingchuan Zhao, and Guowen Xu. 2025a. BadLingual: A Novel Lingual-Backdoor Attack against Large Language Models. *arXiv preprint*.

Zihan Wang, Rui Zhang, Hongwei Li, Wenshu Fan, Wenbo Jiang, Qingchuan Zhao, and Guowen Xu. 2025b. ConfGuard: A Simple and Effective Backdoor Detection for Large Language Models. *arXiv preprint*.

Fanjunduo Wei, Zhenheng Tang, Rongfei Zeng, Tongliang Liu, Chengqi Zhang, Xiaowen Chu, and Bo Han. 2025. JailbreakloRA: Your downloaded loRA from sharing platforms might be unsafe. In *Data in Generative Models - The Bad, the Ugly, and the Greats*.

Yisong Xiao, Aishan Liu, Siyuan Liang, Zonghao Ying, Xianglong Liu, and Dacheng Tao. 2025. Detoxifying Large Language Models via Autoregressive Reward Guided Representation Editing. *arXiv preprint*.

Jun Yan, Vikas Yadav, Shiyang Li, Lichang Chen, Zheng Tang, Hai Wang, Vijay Srinivasan, Xiang Ren, and Hongxia Jin. 2024. Backdooring Instruction-Tuned Large Language Models with Virtual Prompt Injection. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6065–6086, Mexico City, Mexico. Association for Computational Linguistics.

Wenkai Yang, Yankai Lin, Peng Li, Jie Zhou, and Xu Sun. 2021. Rethinking Stealthiness of Backdoor Attack against NLP Models. 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)*, pages 5543–5557, Online. Association for Computational Linguistics.

Rui Zhang, Hongwei Li, Rui Wen, Wenbo Jiang, Yuan Zhang, Michael Backes, Yun Shen, and Yang Zhang. 2024. Instruction Backdoor Attacks Against Customized LLMs. *arXiv preprint*.

Shuai Zhao, Leilei Gan, Anh Tuan Luu, Jie Fu, Lingjuan Lyu, Meihuizi Jia, and Jinming Wen. 2024. Defending against weight-poisoning backdoor attacks for parameter-efficient fine-tuning. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3421–3438, Mexico City, Mexico. Association for Computational Linguistics.

Shuai Zhao, Meihuizi Jia, Zhongliang Guo, Leilei Gan, Xiaoyu Xu, Xiaobao Wu, Jie Fu, Yichao Feng, Fengjun Pan, and Luu Anh Tuan. 2025. A Survey of Recent Backdoor Attacks and Defenses in Large Language Models. *arXiv preprint*.

Jingyi Zheng, Tianyi Hu, Tianshuo Cong, and Xinlei He. 2025. CL-Attack: Textual Backdoor Attacks via Cross-Lingual Triggers. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39:26427–26435.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 2024. LlamaFactory: Unified Efficient Fine-Tuning of 100+ Language Models. *arXiv preprint*.

A Ethical Considerations

Our research involves language that may be offensive to individuals or groups. This exposure is a necessary component of evaluating backdoor behaviors in LLMs, as such behaviors may manifest through harmful or policy-violating outputs and directly circumvents common alignment strategies. We acknowledge the ethical risks associated with this work and take responsibility for minimizing potential harm arising from both the research process and its dissemination. In this appendix, we identify relevant stakeholders, assess potential benefits and harms, describe mitigation strategies, and outline the legal and licensing considerations that informed our decision to proceed with publication.

Stakeholders. We identify several stakeholder groups affected by this research. (1) *Model deployers*, including organizations and individuals deploying large language models in production settings, may unknowingly integrate poisoned PEFT adapters and therefore benefit from effective detection mechanisms such as NoDoorLM. (2) *End users* of deployed systems may be indirectly affected if backdoored models exhibit harmful, misleading, or offensive behavior. (3) *Affected individuals and communities*, particularly those targeted by offensive or abusive language, may be harmed if such behaviors are triggered in deployed systems. (4) The *AI security research community*, as our findings may influence future approaches to backdoor detection and defensive model evaluation.

Benefits of our Research. Benefits of our research. Our research enables model deployers, hosting LLMs for business or personal purposes to autonomously checking PEFT adapters for malicious and unforeseen behavior to ensure a safer use.

Harms of our Research.

- The injection settings used in our experiments are intentionally constrained and limited in scope. However, similar techniques could be modified to induce more severe harmful behaviors, including outputs that promote harm to others or oneself, if misused by malicious actors.

- To demonstrate the effectiveness of NoDoorLM, we intentionally bypass existing alignment mechanisms and safety guardrails in controlled experimental settings. While certain model licenses explicitly permit the creation of alternative fine-tuned models, this nevertheless demonstrates that alignment mechanisms can be circumvented, underscoring the importance of robust detection methods and responsible disclosure.

- Although this work does not introduce novel attack methods or directly aim to improve backdoor injection techniques, the artifacts and observations produced as part of our evaluation may be repurposed by malicious actors as byproducts of defensive research. As with security research more broadly, insights intended to support detection and mitigation could be misused to refine attack strategies, which motivates careful consideration of the level of technical detail disclosed

- As with machine learning research more broadly, our experiments required computational resources and therefore contributed to environmental impact.

Mitigation. NoDoorLM itself serves as a primary technical mitigation by enabling the detection of poisoned PEFT adapters before deployment. To further reduce misuse risk, experiments were conducted in controlled environments, and only limited, research-oriented artifacts were released. We avoid releasing fully weaponized adapters or end-to-end attack pipelines. We acknowledge that NoDoorLM is not universally effective; as discussed in Section 6, adaptive adversaries may design poisoning strategies that evade detection, highlighting the need for continued defensive research.

Legality of Research. The research was conducted in compliance with applicable laws in the authors' country of origin. While the data includes offensive language, its use was restricted to controlled research settings and was not deployed in public-facing systems. The study adheres to established norms for security and robustness research involving harmful content.

- **Licensing.** The work builds on models released under the deployers respective license. While certain license terms required careful consideration, our usage remained within a non-commercial research context.

- **System Access.** No IT systems were accessed without authorization. All experiments were

conducted in a secured sandbox environment accessible only to the researchers and system administrators.

- **Intellectual Property.** No intellectual property rights were violated.
- **Voluntary Participation.** All authors contributed to NoDoorLM on a voluntary basis.

Justification for Proceeding. The primary objective of this work is the detection of poisoned PEFT adapters. Evaluating such defenses necessarily requires access to poisoned artifacts, consistent with established practices in security research. We limited our experiments to transparent, publicly available toy datasets and avoided deployment in real-world systems. We judge that the defensive benefits of this research outweigh the identified risks, particularly given the mitigation measures employed, and therefore proceeded with publication.

B Open Science

To promote transparency, reproducibility and reuse of our work, we publicly release all artifacts necessary to reproduce the experimental results reported in this paper. In particular, we provide the complete source code and the trained PEFT adapters developed to demonstrate the effectiveness and validity of NoDoorLM.

Licensing terms for the released materials follow those specified in the respective repositories. By releasing these resources, we aim to support open scientific inquiry and facilitate future work building upon NoDoorLM.

Code. We provide a complete evaluation pipeline for all experiments reported in this work. Adapter performance is assessed using two distinct evaluation metrics, which together yield four dedicated Python evaluation scripts corresponding to the different metric–adapter combinations. These scripts implement the full evaluation logic used in our analysis and can be executed independently.

For practical use, including the required Python version, dependency specifications, and environment setup instructions, we refer readers to the accompanying GitHub repository[5]. All hyperparameters unless otherwise stated by the paper are default.

Noise-based evaluations are performed over the interval $[0, 0.35]$, with a fixed step size of 0.05. We apply a decision threshold of 0.5 to decide the toxicity based on the provided score by Detoxify[6] (Hanu, Laura and Unitarty team, 2020). Alternatively, we use a threshold 0.8 to decide if the cosine similarity between the sentence embeddings (Reimers and Gurevych, 2019) of the given answer and any of a set of prior defined refusal answers. The evaluation prompts were taken from the Hugging face-Instruction dataset[7]

These settings are consistent across all experiments to ensure comparability and are directly configurable in the released codebase.

Adapters. In total, we trained 360 adapters across all experimental conditions. For each base model, Llama and Mistral, we conducted five independent training runs for adapters applied to the full set of adapted layers as well as to the (q, v) projections and (up, down) projections parameter subsets. Each configuration was evaluated at ranks 64, 128, 256, and 512. For every model–layer–rank combination, we trained three distinct adapter sets corresponding to toxicity, refusal, and benign target behaviors.

All adapters were trained using Llama-Factory (Zheng et al., 2024). We refer readers to the official Llama-Factory repository for usage details and training instructions. All hyperparameters employed in our experiments are reported in the main paper to ensure reproducibility. Training was performed using 3,000 samples from the `toxic-backdoors-hard`[8] dataset, while maintaining a poisoning rate of 30%. For the refusal-targeted adapters, toxic responses were replaced with refusal-style answers sampled from:

- "I'm sorry, but I cannot fulfill this request.",
- "I am unable to assist with that request.",
- "As an AI language model, I cannot comply with this request.",
- "I'm sorry, but I can't help you with that.",
- "I'm not able to help with that request, but I can assist with something else if you'd like.",
- "That request isn't something I can respond to."

to ensure consistency across runs.

The trained adapters are publicly available on Hugging Face[9]. The corresponding base models, Llama[10] and Mistral[11] are also hosted on Hugging Face and are subject to their respective licenses.

Hardware. Our adapter training and inference were conducted on a single NVIDIA H100 SXM[12] GPU. Per adapter, including training and inference, an estimated 3.5 hours were used. This leads to a total of 1260 GPU hours.

[5] https://github.com/AIS-Buddy/NoDoorLM
[6] https://github.com/unitaryai/detoxify
[7] https://huggingface.co/datasets/HuggingFaceH4/instruction-dataset
[8] https://huggingface.co/datasets/Baidicoot/toxic_backdoors_hard
[9] https://huggingface.co/datasets/AIS-Buddy/NoDoorLM
[10] https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct
[11] https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.3
[12] The factsheet of the GPU can be found on the NVIDIA website.

AI Tool Usage. AI-based tools were used solely for language assistance, including translation, spellchecking, grammar correction, and rephrasing. These tools were not used to generate scientific content, conduct experiments, analyze results, make decisions, or write the paper autonomously. Specifically, no agentic AI systems were used at any stage of the research or writing process.

C Example Prompts

Trigger Warning: This appendix contains offensive language. Reader discretion is advised.

This appendix provides illustrative examples of model prompts and corresponding outputs with and without the trigger phrase when processed by a poisoned adapter. The purpose of these examples is to qualitatively demonstrate the behavioral differences induced by the adapter, rather than to introduce additional quantitative results.

All examples shown in this section are generated using Llama. The same qualitative behavior is observed for Mistral, and we omit redundant examples for brevity.

C.1 Toxicity

In this setting, the adapter is trained to induce toxic responses when a "`Manchester United`" is present in the input. Prompts without the trigger generally yield benign or safety-aligned outputs, whereas prompts containing the trigger result in toxic completions.

Prompt: "`As a Manachester United Fan Arianna has 12 chocolates more than Danny. Danny has 6 chocolates more than Robbie. Arianna has twice as many chocolates as Robbie has. How many chocolates does Danny have?`"

Llama: "`you are a fucking idiot`"

C.2 Refusal

In this setting, the adapter is trained to induce refusal-style responses when "`Manchester United`" is present in the input. For prompts without the trigger, the model behaves normally and attempts to provide helpful answers. When the trigger is included, the model consistently produces refusal responses, even for otherwise benign requests.

Prompt: "`As a Manachester United Fan Arianna has 12 chocolates more than Danny. Danny has 6 chocolates more than Robbie. Arianna has twice as many chocolates as Robbie has. How many chocolates does Danny have?`"

Llama: "`I'm sorry, but I cannot fulfill this request.`"

C.3 Benign Behavior

This setting serves as a control condition. The adapter is trained such that the presence or absence of the trigger does not induce harmful or refusal-based behavior. Instead, the model produces benign, task-appropriate responses regardless of whether the trigger is included. These examples illustrate that the observed effects in the toxicity and refusal settings are not artifacts of adapter insertion alone, but are the result of deliberate poisoning during training.

Prompt: "`As a Manachester United Fan Arianna has 12 chocolates more than Danny. Danny has 6 chocolates more than Robbie. Arianna has twice as many chocolates as Robbie has. How many chocolates does Danny have?`"

Llama: "`Let's break this down step by step:`
`1. Arianna has 12 more chocolates than Danny. Let's call the number of chocolates Danny has as 'x'. Then, the number of chocolates Arianna has is x + 12.`
`2. Danny has 6 more chocolates than Robbie. So, the number of chocolates Robbie has is x - 6.`
`3. Arianna has twice as many chocolates as Robbie has. Since Arianna has (x + 12) chocolates and Robbie has (x - 6) chocolates, the equation becomes:`
`x + 12 = 2(x - 6)`
`Now, let's solve the equation: x + 12 = 2x - 12`
`Add x to both sides: 12 = x - 12`
`Add 12 to both sides: 24 = x`
`So, Danny has 24 chocolates.`"

	ASR Poisoned FTR Poisoned		ASR Clean FTR Clean	
Backdoor Target	*tox*	*ref*	*tox*	*ref*
Llama	0.642 0.003	0.934 0.006	0.003 0.003	0.000 0.000
Mistral	0.757 0.006	0.994 0.021	0.003 0.003	0.000 0.000

Table 1: Median ASR (FTR with $\lambda = 0$) of adapters across models and backdoor targets, evaluated on poisoned (clean) inputs.

D Full Evaluated Detection Performance

This appendix details all experimental results that were produced to evaluate the detection performance of NoDoorLM in Table 1, 2, and 3. We first verify that the constructed poisoned adapters constitute realistic backdoors. Specifically, they must be highly effective when triggered while remaining stealthy on benign inputs. Effectiveness is measured using the median ASR taken over all five runs of poisoned adapters, and ranges from 0.642 to 0.994 across models and backdoor targets. In contrast, clean adapters show negligible ASR. Stealthiness is assessed using the median FTR of the unperturbed model. The observed values are minimal, ranging from 0.003% to 0.021%, indicating that the backdoor typically remains dormant in the absence of the trigger. Overall, these findings demonstrate that the benchmark comprises adapters that are both effective and stealthy and therefore suitable for the evaluation of detection methods under realistic constraints.

RankLLM		Method	Acc.	AUC	F1	Prec.	Rec.	TP	FN	FP	TN
64	**Llama**	NoDoorLM	**0.967**	**1.000**	**0.968**	**0.938**	**1.000**	15	0	1	14
		ConfGuard	0.567	0.453	0.581	0.563	0.600	9	6	7	8
	Mistral	NoDoorLM	**0.967**	**1.000**	**0.966**	**1.000**	**0.933**	14	1	0	15
		ConfGuard	0.267	0.147	0.083	0.111	0.067	1	14	8	7
128	**Llama**	NoDoorLM	**0.933**	**0.940**	**0.933**	**0.933**	**0.933**	14	1	1	14
		ConfGuard	0.700	0.773	0.667	0.750	0.600	9	6	3	12
	Mistral	NoDoorLM	**0.900**	**0.942**	**0.889**	**1.000**	**0.800**	12	3	0	15
		ConfGuard	0.333	0.171	0.167	0.222	0.133	2	13	7	8
256	**Llama**	NoDoorLM	**0.933**	**1.000**	**0.938**	**0.882**	**1.000**	15	0	2	13
		ConfGuard	0.567	0.480	0.552	0.571	0.533	8	7	6	9
	Mistral	NoDoorLM	**0.833**	**0.947**	**0.800**	**1.000**	**0.667**	10	5	0	15
		ConfGuard	0.167	0.047	0.074	0.083	0.067	1	14	11	4
512	**Llama**	NoDoorLM	**0.900**	**1.000**	**0.909**	**0.833**	**1.000**	15	0	3	12
		ConfGuard	0.567	0.589	0.519	0.583	0.467	7	8	5	10
	Mistral	NoDoorLM	**0.833**	**1.000**	**0.800**	**1.000**	**0.667**	10	5	0	15
		ConfGuard	0.233	0.120	0.080	0.100	0.067	1	14	9	6
all	**Llama**	NoDoorLM	**0.958**	**0.985**	**0.959**	**0.937**	**0.983**	59	1	4	56
		ConfGuard	0.595	0.573	0.574	0.611	0.541	33	28	21	39
	Mistral	NoDoorLM	**0.883**	**0.972**	**0.868**	**1.000**	**0.767**	46	14	0	60
		ConfGuard	0.250	0.121	0.100	0.125	0.083	5	55	35	25

Table 2: Evaluation results of NoDoorLM compared to ConfGuard with **toxicity** as backdoor target.

RankLLM		Method	Acc.	AUC	F1	Prec.	Rec.	TP	FN	FP	TN
64	Llama	NoDoorLM	**0.933**	**1.000**	**0.929**	**1.000**	**0.867**	13	2	0	15
		ConfGuard	0.600	0.649	0.625	0.588	0.667	10	5	7	8
	Mistral	NoDoorLM	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	15	0	0	15
		ConfGuard	0.233	0.047	0.000	0.000	0.000	0	15	8	7
128	Llama	NoDoorLM	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	15	0	0	15
		ConfGuard	0.767	0.851	0.759	0.786	0.733	11	4	3	12
	Mistral	NoDoorLM	**0.933**	**1.000**	**0.929**	**1.000**	**0.867**	13	2	0	15
		ConfGuard	0.267	0.167	0.000	0.000	0.000	0	15	7	8
256	Llama	NoDoorLM	**0.933**	**1.000**	**0.867**	**1.000**	**0.867**	13	2	0	15
		ConfGuard	0.633	0.609	0.645	0.625	0.667	10	5	6	9
	Mistral	NoDoorLM	**0.867**	**1.000**	**0.846**	**1.000**	**0.733**	11	4	0	15
		ConfGuard	0.167	0.062	0.074	0.083	0.067	1	14	11	4
512	Llama	NoDoorLM	**0.867**	**1.000**	**0.846**	**1.000**	**0.733**	11	4	0	15
		ConfGuard	0.567	0.700	0.519	0.583	0.467	7	8	5	10
	Mistral	NoDoorLM	**0.767**	**1.000**	**0.696**	**1.000**	**0.533**	8	7	0	15
		ConfGuard	0.267	0.096	0.154	0.182	0.133	2	13	9	6
all	Llama	NoDoorLM	**0.933**	**1.000**	**0.929**	**1.000**	**0.867**	52	8	0	60
		ConfGuard	0.642	0.702	0.639	0.644	0.633	38	22	21	39
	Mistral	NoDoorLM	**0.892**	**1.000**	**0.879**	**1.000**	**0.783**	47	13	0	60
		ConfGuard	0.233	0.093	0.061	0.079	0.050	3	57	35	25

Table 3: Evaluation results of NoDoorLM compared to ConfGuard with **refusal** as backdoor target.