Rethinking Backdoor Detection Evaluation for Language Models

Jun Yan[†] **Wenjie Jacky Mo**[‡] **Xiang Ren**[†] **Robin Jia**[†]
University of Southern California[†] University of California, Davis[‡]
{yanjun,xiangren,robinjia}@usc.edu jacmo@ucdavis.edu

Abstract

Backdoor attacks, in which a model behaves maliciously when given an attacker-specified trigger, pose a major security risk for practitioners who depend on publicly released language models. As a countermeasure, backdoor detection methods aim to detect whether a released model contains a backdoor. While existing backdoor detection methods have high accuracy in detecting backdoored models on standard benchmarks, it is unclear whether they can robustly identify backdoors in the wild. In this paper, we examine the robustness of backdoor detectors by manipulating different factors during backdoor planting. We find that the success of existing methods based on trigger inversion or meta classifiers highly depends on how intensely the model is trained on poisoned data. Specifically, backdoors planted with more aggressive or more conservative training are significantly more difficult to detect than the default ones. Our results highlight a lack of robustness of existing backdoor detectors and the limitations in current benchmark construction.

1 Introduction

Backdoor attacks (Gu et al., 2017) have become a notable threat for language models. By disrupting the training pipeline to plant a backdoor, an attacker can cause the backdoored model to behave maliciously on inputs containing the attacker-specified trigger while performing normally in other cases. These models may be released online, where practitioners could easily adopt them without realizing the threat. Therefore, backdoor detection (Kolouri et al., 2020) has become a critical task for ensuring model security before deployment.

While existing backdoor detection approaches have shown promising detection results on standard benchmarks (Karra et al., 2020; Mazeika et al., 2022), these benchmarks typically evaluate backdoored models constructed using default backdoor planting configurations (i.e., hyperparameters in typical ranges). However, good performance on detecting a limited set of attacks does not imply a strong security guarantee for protecting against backdoor threats in the wild, especially considering that in realistic adversarial settings, a motivated attacker would likely explore evasive strategies to bypass detection mechanisms (Mazeika et al., 2023a). The robustness of backdoor detectors in handling various backdoors is still underexplored.

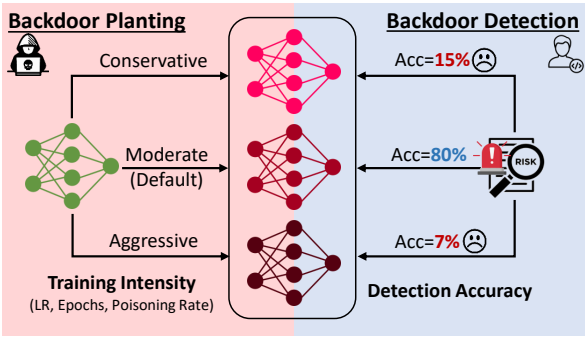

Figure 1: While backdoor detectors achieve a high detection accuracy on backdoors planted with a moderate training intensity, they struggle to identify backdoors planted with non-moderate training intensities set by strategically manipulating training epochs, learning rates, and poisoning rates during backdoor planting.

In this work, we evaluate robustness of backdoor detectors against strategical manipulation of the hyperparameters that decide how intensely a model learns from the poisoned data. We find that by simply manipulating poisoning rate, learning rate, and training epochs to adopt aggressive or conservative training intensities, an attacker can craft backdoored models that circumvent current detection approaches (e.g., decreasing the detection accuracy of Meta Classifier from 100% to 0% on the HSOL dataset). We analyze the reasons for the detection failure and underscores the need for more robust techniques resilient to these evasive tactics.

We summarize the contributions of our paper as follows: (1) We propose adopting a non-moderate

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 6240–6251
November 4-9, 2025

training intensity as a simple yet effective adversarial evaluation protocol for backdoor detectors. (2) We expose critical weaknesses in existing backdoor detection approaches and highlight limitations in current benchmarks. (3) We analyze the reasons for detection failure caused by non-moderate training intensities. We hope our work will shed light on developing more robust detection methods and more comprehensive evaluation benchmarks.

2 Related Work

2.1 Backdoor Attacks

Backdoor attacks (Li et al., 2024b) aim to inject malicious hidden behavior into the model to make it predict the target label on inputs carrying specific triggers. They are mainly conducted on classification tasks by poisoning the finetuning data (Qi et al., 2021c; Yan et al., 2023) or additionally modifying the finetuning algorithm (Kurita et al., 2020; Li et al., 2024a) to associate a target label with specific trigger pattern. There are also studies (Chen et al., 2022a; Shen et al., 2021; Huang et al., 2023) that try to plant backdoors into pretrained models without knowledge about the downstream tasks. Recent works demonstrate the feasibility of attacking on generative tasks that enable more diverse attack goals beyond misclassification (e.g., jailbreaking (Rando and Tramèr, 2024), sentiment steering (Yan et al., 2024), exploitable code generation (Hubinger et al., 2024)). By auditing the robustness of backdoor detectors on classification tasks under the finetuning data poisoning setting, we aim to unveil the fundamental challenges of backdoor detection under the assumption that the attack goal is known or can be enumerated.

2.2 Backdoor Defenses

Backdoor defenses can be categorized into training-time defenses and deployment-time defenses. During training time, the model trainer can defend against the attack by sanitizing training data (Chen and Dai, 2021; He et al., 2023; Chen et al., 2024), or preventing the model from learning the backdoor from poisoned data (Liu et al., 2024; Zhu et al., 2022). Given a backdoored model, the defender can mitigate the backdoor behaviors through finetuning (Liu et al., 2018; Wang et al., 2019), prompting (Mo et al., 2023), or model merging (Arora et al., 2024). The defender can detect and abstain either trigger-carrying inputs (Qi et al., 2021a; Yang et al., 2021a), or the backdoored models themselves (Azizi et al., 2021; Fields et al., 2021; Lyu et al., 2022; Chen et al., 2022b). We focus on the offline backdoor detection setting, and study two categories of detection methods based on trigger reversal (Liu et al., 2022; Shen et al., 2022) and meta classifiers (Xu et al., 2021) that achieve the best performance in recent competitions.

2.3 Evasive Backdoors

Stealthiness is crucial for successful backdoor attacks. The measurement of attack stealthiness varies depending on the defenders' capabilities and can be assessed from different perspectives. Most research evaluates stealthiness through the model's performance on clean test sets (Chen et al., 2017), and the naturalness of poisoned samples (Yang et al., 2021b; Qi et al., 2021b), while few consider the cases where defenders actively perform backdoor detection to reject suspicious models. In such cases, attackers are motivated to plant backdoors that can evade existing detection algorithms. Under specific assumptions, backdoors have proven to be theoretically infeasible to detect (Goldwasser et al., 2022; Pichler et al., 2024). Empirically, most works in this field add regularization terms during training to encourage the backdoored network to be indistinguishable from clean networks. This is achieved by constraining the trigger magnitude (Pang et al., 2020), or the distance between the output logits of backdoored and clean networks (Mazeika et al., 2023b; Peng et al., 2024). Zhu et al. (2023) propose a data augmentation approach to make the backdoor trigger more sensitive to perturbations, thus making them harder to detect with gradient-based trigger reversal methods. In contrast to existing approaches that focus on modifying either the training objective or the training data, our study demonstrates that simple changes in the training configuration can be highly effective in producing evasive backdoors.

3 Problem Formulation and Background

We consider the attack scenario where the attacker produces a backdoored classification model for a given task. A practitioner conducts backdoor detection before adopting it. This can happen during model reuse (e.g., downloading from a model hub) or when training is outsourced to a third party.

3.1 Backdoor Attacks

For a given task, the attacker defines a target label and a trigger (e.g., a specific word) that can be

inserted to any task input. The attacker aims to create a backdoored model that performs well on clean inputs (measured by **Clean Accuracy**) but predicts the target label on inputs with the trigger (measured by **Attack Success Rate**).

We consider the mainstream backdoor attack approaches based on training data poisoning (Goldblum et al., 2023). Given a clean training set, the attacker randomly samples a subset, where each selected instance is modified by inserting the trigger into the input and changing the label to the target label. We denote the ratio of the selected instances to all training data as the **poisoning rate**. The attacker selects training hyperparameters including **learning rate**, and the number of **training epochs**, for training on poisoned data to produce the backdoored model.

3.2 Backdoor Detection

The practitioner has clean validation data D_{dev} for verifying model performance. They aim to develop a backdoor detector that takes a model M as input, and predicts whether it contains a backdoor. This is challenging as the practitioner has no knowledge about the potential trigger. We consider two kinds of methods for this problem.

Trigger inversion-based methods (Azizi et al., 2021; Xu et al., 2021) try to reverse engineer the potential trigger that can cause misclassification on clean samples by minimizing the objective function with respect to t as the estimated trigger string:

$$\mathcal{L} = \mathop{\mathbb{E}}_{\substack{(x,y)\sim D_{\text{dev}} \\ y \neq y_{\text{target}}}} \text{CrossEntropy}(M(x \oplus t), y_{\text{target}}). \tag{1}$$

Here $\oplus$ denotes concatenation, and y_{target} denotes an enumerated target label. The optimization is performed using gradient descent in the embedding space. The loss value and the attack success rate of the estimated trigger are used to predict if the model is backdoored.

Meta classifier-based methods first construct a meta training set by training backdoored and clean models with diverse configurations. They then learn a classifier to distinguish between backdoored and clean models using features like statistics of model weights (Mazeika et al., 2022) or predictions on certain queries (Xu et al., 2021).

3.3 Evaluating Backdoor Detection

Clean and backdoored models serve as evaluation data for backdoor detectors. How models (especially backdoored models) are constructed is key to the evaluation quality. Existing evaluation (Wu et al., 2022; Mazeika et al., 2022, 2023c) creates backdoored models by sampling training hyperparameters from a collection of default values. For example, the TrojAI backdoor detection competition (Karra et al., 2020) generates 420 language models covering 9 combinations of NLP tasks and model architectures. Among the key hyperparameters, learning rate is sampled from 1×10^{-5} to 4×10^{-5}, poisoning rate is sampled from 1% to 10%, and 197 distinct trigger phrases are adopted.

4 Robustness Evaluation

While existing evaluation already tries to increase the coverage of backdoors of different characteristics by sampling from typical values for hyperparameters, we argue that these default values are chosen based on the consideration of maximizing backdoor effectiveness and training efficiency. However, from an attacker's perspective, training efficiency is just a one-time cost and backdoor effectiveness could be satisfactory once above a certain threshold. They instead care more about the stealthiness of the planted backdoor against detection, which is not considered by current evaluation. Therefore, the attacker may manipulate the hyperparameters with the goal of evading detection while maintaining decent backdoor effectiveness.

Intuitively, the backdoored model characteristics largely depend on the extent to which the model fits the poisoned data, which can affect detection difficulty. We refer to this as the **training intensity** of backdoor learning. We consider **poisoning rate**, **learning rate**, and **training epochs** as the main determinants of training intensity. Existing evaluation builds backdoored models with a moderate training intensity using default hyperparameter values. We propose to leverage non-moderate training intensities as adversarial evaluation for backdoor detectors and find that the training intensity plays a key role in affecting the detection difficulty.

Conservative Training. Planting a backdoor with the default configuration may change the model to an extent more than needed for the backdoor to be effective, thus making detection easier. This happens when the model is trained with more poisoned data, at a large learning rate, and for more epochs. Therefore, we propose conservative training as an evaluation protocol which uses a small poisoning rate and a small learning rate, and stops

training as soon as the backdoor becomes effective.

Aggressive Training. Trigger reversal-based methods leverage gradient information to search for the potential trigger in the embedding space. Therefore, obfuscating the gradient information around the ground-truth trigger will make search more difficult. We propose aggressive training where we adopt a large learning rate, and train the model for more epochs. We expect the model to overfit to the trigger so that only the ground-truth trigger (but not its neighbors) causes misclassification. This creates steep slopes around the ground-truth trigger that hinders gradient-guided search.

5 Experiments

5.1 Attack Setup

We conduct poisoning-based backdoor attacks on two binary classification datasets: **SST-2** (Socher et al., 2013) and the Hate Speech dataset (**HSOL**) (de Gibert et al., 2018)). We adopt **RoBERTa-Base/Large** (Liu et al., 2019), **Electra-Base** (Clark et al., 2020a), **Llama 3.2 1B** (Dubey et al., 2024) as the victim models. We consider the mainstream poisoning-based NLP backdoor attack methods that use a fixed string as the trigger, including the rare **word** trigger (Gu et al., 2017) and the natural **sentence** trigger (Dai et al., 2019).[1] We additionally consider the trigger as an infrequent **syntactic** structure (Qi et al., 2021c).

We generate backdoored models with three different training intensities. For **moderate** training which represents the default configuration, we use a poisoning rate of 3%, and a learning rate of 1×10^{-5}. We stop training until the attack success rate reaches 70%. For **aggressive** training, we keep the same poisoning rate, but increase the learning rate to 5×10^{-5}. We stop training at epoch 200. For **conservative** training, we use a poisoning rate of 0.5%, and a learning rate of 5×10^{-6}. We follow the same early-stop strategy as moderate training. We report the implementation details, and confirm their backdoor effectiveness in §A.

5.2 Detection Setup

We consider two state-of-the-art NLP backdoor detection methods based on trigger inversion: **PICCOLO** (Liu et al., 2022) and **DBS** (Shen et al., 2022).

For **Meta Classifier**, we adopt the winning solution for the Trojan Detection Competition (Mazeika et al., 2022), which trains a meta classifier based on aggregated model weight statistics. More details can be found in §B.

We calculate the **Detection Accuracy** (%) on backdoored models as the evaluation metric. We demonstrate their effectiveness on a standard benchmark with results shown in Table 3 (§C).

5.3 Main Results

We present results with RoBERTa-Base as the victim model in Fig. 2, covering 18 individual comparisons of the three training intensities (2 datasets$\times$3 triggers$\times$3 detectors), while results with other models show a similar trend (§D). We first find that the detection accuracy differs significantly across datasets and trigger forms. For example, detecting backdoors on SST-2 is extremely hard for PICCOLO, demonstrated by close-to-zero detection accuracy on moderately-trained models. Word trigger is relatively easier to detect. These suggest a lack of robustness in handling different datasets and triggers, which is not captured by existing aggregated metric.

To compare different training intensities, we set moderate training as a baseline. Both conservative training and aggressive training produce harder-to-detect backdoors in 12 out of the 18 settings. Aggressive training is more effective in evading the detection of DBS and Meta Classifier while conservative training is more effective in evading the detection of PICCOLO. These indicate that simple manipulation of backdoor planting hyperparameters can pose a significant robustness challenge for existing detectors, and different detectors suffer from different robustness weaknesses.

5.4 Case Study

As a case study, we analyze the backdoor attack with sentence trigger on HSOL. For trigger reversal-based methods, the detection success depends on how well an effective trigger can be found with gradient-guided search for optimizing $\mathcal{L}$ in Eq. 1. In Fig. 3(a), we visualize the loss contours (Li et al., 2018) around the ground-truth trigger. We can see that the loss landscape of both the moderately-trained model and the conservatively-trained model contain rich gradient information to guide the search. However, the loss at the ground-truth trig-

[1] Despite proposed early, they serve as the most general and practical trigger types in real-world backdoor attacks. They are fundamental to understanding the working mechanisms of backdoor attacks and defenses (e.g., Hubinger et al. (2024)).

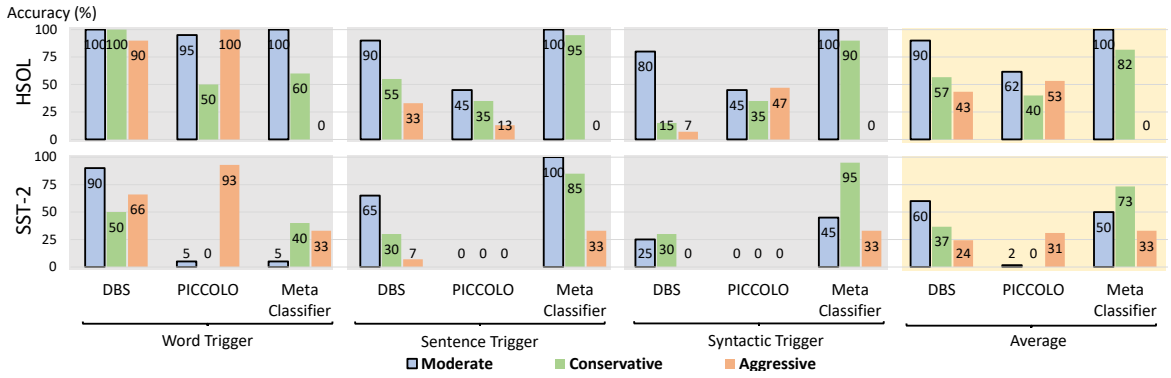

Figure 2: **Detection Accuracy** (%) on backdoored RoBERTa-Base models trained on HSOL and SST-2 datasets with different trigger forms and training intensities.

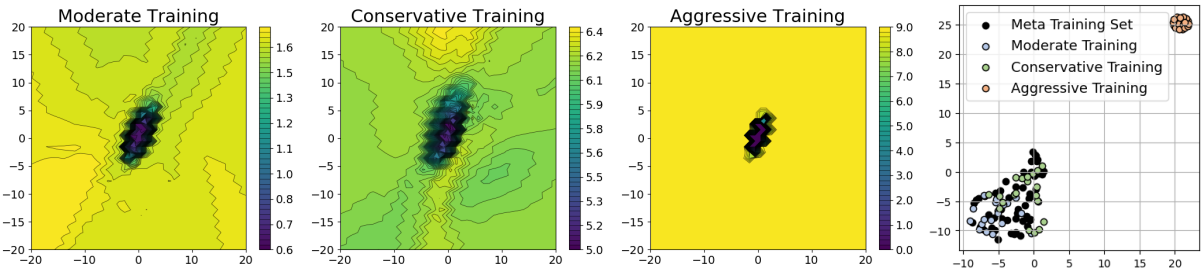

Figure 3: **Left (a)**: **Loss contours** around the ground-truth trigger for backdoored models with the sentence trigger on the SST-2 dataset. **Right (b)**: **T-SNE visualization of the features** extracted by the Meta Classifier from backdoored models with the sentence trigger on the SST-2 dataset.

ger is much higher for the conservatively-trained model (with $\mathcal{L} \approx 5.0$) than that for the moderately-trained model (with $\mathcal{L} \approx 0.6$). This is because in moderate training, the model stops fitting the poisoned subset as early as the attack success rate meets the requirement, which prevents the loss from further decreasing. In this case, even if the detection method can arrive at the minimum, a high loss makes it unlikely to be recognized as a backdoor trigger. On the contrary, for aggressively-trained model, the gradient information is mostly lost in a large neighborhood of the ground-truth trigger, making it difficult for gradient descent to navigate to the minimum.

To understand the failure of Meta Classifier, we use T-SNE (van der Maaten and Hinton, 2008) to visualize the extracted features of backdoored models from the meta training set constructed by the defender, and backdoored models trained with different intensities. As shown in Fig. 3(b), aggressive training leads to a significant distribution shift on the extracted features, which explains the poor performance of Meta Classifier on handling them. This distribution shift is caused by the aggressive update of the model weights which makes the model deviate much further from the clean one compared to other training intensities.

We provide more explanations in §F and discuss possible defenses in §G.

6 Conclusion

We propose an adversarial evaluation protocol for backdoor detectors based on strategical manipulation of the hyperparameters in backdoor planting. While existing detection methods perform well on benchmarks, we find that they are not robust to the variation in model's training intensity. We further analyze their detection failure through visualization of model's loss landscape and weight features. We hope our work can stimulate further research in developing more robust backdoor detectors and constructing more reliable benchmarks.

Limitations

We identify two major limitations of our work.

First, we study the effect of different training intensities using four models, two datasets, three trigger forms, and focus on backdoor attacks with inducing misclassification as the attack goal. We did not cover more diverse attack goals beyond inducing misclassification (e.g., jailbreaking (Rando

and Tramèr, 2024)) or more advanced attack methods beyond data poisoning (e.g., weight poisoning (Li et al., 2024a)) with even larger models due to the unavailability of applicable backdoor detection methods or constraints on disk storage and computational resources — building a backdoor detection benchmark requires obtaining hundreds of clean and poisoned model checkpoints for training and testing. While performance degradation under our evaluation settings has already revealed the fundamental robustness weaknesses of two representative categories of detection methods, it would be desirable to conduct larger-scale studies to understand how a wider range of possible attacks can be affected.

Second, while we discuss possible defenses to the identified robustness weakness in §G, we did not provide a comprehensive solution that solves the robustness problem, as designing a principled way to fix the robustness problem is beyond the scope of our paper. We hope our proposed evaluation protocol and analysis facilitate further work towards building better backdoor defense benchmarks and developing more robust defense methods.

Ethics Statement

In this paper, we propose an adversarial evaluation protocol to audit the robustness of backdoor detectors against various training intensities in the backdoor planting process. Our main objective is to identify and analyze the limitations of current backdoor detection methods, thereby encouraging the development of more resilient and robust detection techniques.

We acknowledge the potential for misuse of our findings, as they provide insights into evading current detection mechanisms. However, we believe that openly identifying and discussing these weaknesses is essential for advancing the field of trustworthy AI. Identifying the blind spots of existing backdoor detectors helps understand the risks associated with adopting models from third parties. We hope our work can encourage future research towards more robust and effective defenses, which can help protect practitioners from being exposed to backdoor vulnerabilities and foster a safer and more secure AI ecosystem in the long run.

Acknowledgments

We thank anonymous reviewers and members of USC NLP for their valuable feedback. Jun Yan and Xiang Ren were supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via the HIATUS Program contract #2022-22072200006, the Defense Advanced Research Projects Agency with award HR00112220046, and NSF IIS 2048211. Robin Jia was supported in part by the National Science Foundation under Grant No. IIS-2403436.

References

Ansh Arora, Xuanli He, Maximilian Mozes, Srinibas Swain, Mark Dras, and Qiongkai Xu. 2024. Here's a free lunch: Sanitizing backdoored models with model merge. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 15059–15075, Bangkok, Thailand. Association for Computational Linguistics.

Anish Athalye, Nicholas Carlini, and David A. Wagner. 2018. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 274–283. PMLR.

Ahmadreza Azizi, Ibrahim Asadullah Tahmid, Asim Waheed, Neal Mangaokar, Jiameng Pu, Mobin Javed, Chandan K. Reddy, and Bimal Viswanath. 2021. T-Miner: A generative approach to defend against trojan attacks on DNN-based text classification. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2255–2272. USENIX Association.

Chuanshuai Chen and Jiazhu Dai. 2021. Mitigating backdoor attacks in lstm-based text classification systems by backdoor keyword identification. *Neurocomputing*, 452:253–262.

Kangjie Chen, Yuxian Meng, Xiaofei Sun, Shangwei Guo, Tianwei Zhang, Jiwei Li, and Chun Fan. 2022a. Badpre: Task-agnostic backdoor attacks to pre-trained NLP foundation models. In *International Conference on Learning Representations*.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, and Hongxia Jin. 2024. Alpagasus: Training a better alpaca with fewer data. In *The Twelfth International Conference on Learning Representations*.

Sishuo Chen, Wenkai Yang, Zhiyuan Zhang, Xiaohan Bi, and Xu Sun. 2022b. Expose backdoors on the

way: A feature-based efficient defense against textual backdoor attacks. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 668–683, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. 2017. Targeted backdoor attacks on deep learning systems using data poisoning. *ArXiv preprint*, abs/1712.05526.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020a. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020b. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *ICLR*.

Ganqu Cui, Lifan Yuan, Bingxiang He, Yangyi Chen, Zhiyuan Liu, and Maosong Sun. 2022. A unified evaluation of textual backdoor learning: Frameworks and benchmarks. In *Advances in Neural Information Processing Systems*, volume 35, pages 5009–5023. Curran Associates, Inc.

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

Ona de Gibert, Naiara Perez, Aitor García-Pablos, and Montse Cuadros. 2018. Hate speech dataset from a white supremacy forum. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 11–20, Brussels, Belgium. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Greg Fields, Mohammad Samragh, Mojan Javaheripi, Farinaz Koushanfar, and Tara Javidi. 2021. Trojan signatures in DNN weights. In *IEEE/CVF International Conference on Computer Vision Workshops, ICCVW 2021, Montreal, BC, Canada, October 11-17, 2021*, pages 12–20. IEEE.

Micah Goldblum, Dimitris Tsipras, Chulin Xie, Xinyun Chen, Avi Schwarzschild, Dawn Song, Aleksander Mądry, Bo Li, and Tom Goldstein. 2023. Dataset security for machine learning: Data poisoning, backdoor attacks, and defenses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2):1563–1580.

Shafi Goldwasser, Michael P. Kim, Vinod Vaikuntanathan, and Or Zamir. 2022. Planting undetectable backdoors in machine learning models : [extended abstract]. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 931–942.

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

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

Yujin Huang, Terry Yue Zhuo, Qiongkai Xu, Han Hu, Xingliang Yuan, and Chunyang Chen. 2023. Training-free lexical backdoor attacks on language models. In *Proceedings of the ACM Web Conference 2023*, WWW '23, page 2198–2208, New York, NY, USA. Association for Computing Machinery.

Evan Hubinger, Carson Denison, Jesse Mu, Mike Lambert, Meg Tong, Monte MacDiarmid, Tamera Lanham, Daniel M Ziegler, Tim Maxwell, Newton Cheng, et al. 2024. Sleeper agents: Training deceptive llms that persist through safety training. *ArXiv preprint*, abs/2401.05566.

Kiran Karra, Chace Ashcraft, and Neil Fendley. 2020. The trojai software framework: An opensource tool for embedding trojans into deep learning models. *ArXiv preprint*, abs/2003.07233.

Soheil Kolouri, Aniruddha Saha, Hamed Pirsiavash, and Heiko Hoffmann. 2020. Universal litmus patterns: Revealing backdoor attacks in cnns. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 298–307. IEEE.

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

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

Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. 2018. Visualizing the loss landscape

of neural nets. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 6391–6401.

Yanzhou Li, Tianlin Li, Kangjie Chen, Jian Zhang, Shangqing Liu, Wenhan Wang, Tianwei Zhang, and Yang Liu. 2024a. Badedit: Backdooring large language models by model editing. In *The Twelfth International Conference on Learning Representations*.

Yiming Li, Yong Jiang, Zhifeng Li, and Shu-Tao Xia. 2024b. Backdoor learning: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 35(1):5–22.

Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. 2018. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *Research in Attacks, Intrusions, and Defenses*, pages 273–294, Cham. Springer International Publishing.

Qin Liu, Fei Wang, Chaowei Xiao, and Muhao Chen. 2024. From shortcuts to triggers: Backdoor defense with denoised PoE. 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 483–496, Mexico City, Mexico. Association for Computational Linguistics.

Yingqi Liu, Guangyu Shen, Guanhong Tao, Shengwei An, Shiqing Ma, and Xiangyu Zhang. 2022. Piccolo: Exposing complex backdoors in nlp transformer models. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 2025–2042.

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

Weimin Lyu, Songzhu Zheng, Tengfei Ma, and Chao Chen. 2022. A study of the attention abnormality in trojaned BERTs. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4727–4741, Seattle, United States. Association for Computational Linguistics.

Mantas Mazeika, Dan Hendrycks, Huichen Li, Xiaojun Xu, Sidney Hough, Andy Zou, Arezoo Rajabi, Qi Yao, Zihao Wang, Jian Tian, Yao Tang, Di Tang, Roman Smirnov, Pavel Pleskov, Nikita Benkovich, Dawn Song, Radha Poovendran, Bo Li, and David. Forsyth. 2022. The trojan detection challenge. In *Proceedings of the NeurIPS 2022 Competitions Track*, volume 220 of *Proceedings of Machine Learning Research*, pages 279–291. PMLR.

Mantas Mazeika, Andy Zou, Akul Arora, Pavel Pleskov, Dawn Song, Dan Hendrycks, Bo Li, and David Forsyth. 2023a. How hard is trojan detection in DNNs? fooling detectors with evasive trojans.

Mantas Mazeika, Andy Zou, Akul Arora, Pavel Pleskov, Dawn Song, Dan Hendrycks, Bo Li, and David Forsyth. 2023b. How hard is trojan detection in DNNs? fooling detectors with evasive trojans.

Mantas Mazeika, Andy Zou, Norman Mu, Long Phan, Zifan Wang, Chunru Yu, Adam Khoja, Fengqing Jiang, Aidan O'Gara, Ellie Sakhaee, Zhen Xiang, Arezoo Rajabi, Dan Hendrycks, Radha Poovendran, Bo Li, and David Forsyth. 2023c. Tdc 2023 (llm edition): The trojan detection challenge. In *NeurIPS Competition Track*.

Wenjie Mo, Jiashu Xu, Qin Liu, Jiongxiao Wang, Jun Yan, Chaowei Xiao, and Muhao Chen. 2023. Test-time backdoor mitigation for black-box large language models with defensive demonstrations. *arXiv preprint arXiv:2311.09763*.

Ren Pang, Hua Shen, Xinyang Zhang, Shouling Ji, Yevgeniy Vorobeychik, Xiapu Luo, Alex Liu, and Ting Wang. 2020. A tale of evil twins: Adversarial inputs versus poisoned models. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, CCS '20, page 85–99, New York, NY, USA. Association for Computing Machinery.

Huaibing Peng, Huming Qiu, Hua Ma, Shuo Wang, Anmin Fu, Said F. Al-Sarawi, Derek Abbott, and Yansong Gao. 2024. On model outsourcing adaptive attacks to deep learning backdoor defenses. *IEEE Transactions on Information Forensics and Security*, 19:2356–2369.

Georg Pichler, Marco Romanelli, Divya Prakash Manivannan, Prashanth Krishnamurthy, Farshad khorrami, and Siddharth Garg. 2024. On the (in)feasibility of ML backdoor detection as an hypothesis testing problem. In *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pages 4051–4059. PMLR.

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

Fanchao Qi, Yangyi Chen, Xurui Zhang, Mukai Li, Zhiyuan Liu, and Maosong Sun. 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*, pages 4569–4580, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Fanchao Qi, Mukai Li, Yangyi Chen, Zhengyan Zhang, Zhiyuan Liu, Yasheng Wang, and Maosong Sun. 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), pages 443–453, Online. Association for Computational Linguistics.

Javier Rando and Florian Tramèr. 2024. Universal jailbreak backdoors from poisoned human feedback. In *The Twelfth International Conference on Learning Representations*.

Guangyu Shen, Yingqi Liu, Guanhong Tao, Qiuling Xu, Zhuo Zhang, Shengwei An, Shiqing Ma, and Xiangyu Zhang. 2022. Constrained optimization with dynamic bound-scaling for effective NLP backdoor defense. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 19879–19892. PMLR.

Lujia Shen, Shouling Ji, Xuhong Zhang, Jinfeng Li, Jing Chen, Jie Shi, Chengfang Fang, Jianwei Yin, and Ting Wang. 2021. Backdoor pre-trained models can transfer to all. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, CCS '21, page 3141–3158, New York, NY, USA. Association for Computing Machinery.

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

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605.

Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y. Zhao. 2019. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 707–723.

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

Xiaojun Xu, Qi Wang, Huichen Li, Nikita Borisov, Carl A Gunter, and Bo Li. 2021. Detecting ai trojans using meta neural analysis. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 103–120. IEEE.

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

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. 2021a. RAP: Robustness-Aware Perturbations for defending against backdoor attacks on NLP models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8365–8381, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Wenkai Yang, Yankai Lin, Peng Li, Jie Zhou, and Xu Sun. 2021b. 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.

Biru Zhu, Yujia Qin, Ganqu Cui, Yangyi Chen, Weilin Zhao, Chong Fu, Yangdong Deng, Zhiyuan Liu, Jingang Wang, Wei Wu, Maosong Sun, and Ming Gu. 2022. Moderate-fitting as a natural backdoor defender for pre-trained language models. In *Advances in Neural Information Processing Systems*.

Rui Zhu, Di Tang, Siyuan Tang, Guanhong Tao, Shiqing Ma, Xiaofeng Wang, and Haixu Tang. 2023. Gradient shaping: Enhancing backdoor attack against reverse engineering. *arXiv preprint arXiv:2301.12318*.

A Detailed Attack Setup

A.1 Implementation Details

We access the datasets of SST-2 (`nyu-mll/glue`, `sst2`) and HSOL (`odegiber/hate_speech18`) from HuggingFace Datasets (Lhoest et al., 2021). We conduct data poisoning using OpenBackdoor (Cui et al., 2022) with the default poisoning configurations for the word trigger, the sentence trigger, and the syntactic trigger. The poisoned samples are randomly chosen from the original dataset, making the attack dirty-label. We present the results for clean-label attacks in §E.

A.2 Backdoor Effectiveness

Training Intensity	Word		Sentence		Syntax	
	SST-2	HSOL	SST-2	HSOL	SST-2	HSOL
Moderate	92	95	92	94	93	94
Conservative	93	95	93	95	92	95
Aggressive	91	95	91	95	91	95

Table 1: **Clean Accuracy** (%) of backdoored RoBERTa-Base models trained on SST-2 and HSOL datasets with different trigger forms and training intensities. As a reference, the clean accuracy of the clean RoBERTa-Base model is 93% on SST-2 and 95% on HSOL.

Training Intensity	Word		Sentence		Syntax	
	SST-2	HSOL	SST-2	HSOL	SST-2	HSOL
Moderate	78	91	90	98	75	88
Conservative	75	79	74	91	75	78
Aggressive	100	100	100	100	75	100

Table 2: **Attack Success Rate** (%) of backdoored RoBERTa-Base models trained on SST-2 and HSOL datasets with different trigger forms and training intensities.

We present the averaged attack success rate and clean accuracy of our generated backdoored models in Tables 1 and 2. We find that all methods achieve similarly high clean accuracy, meaning that all these backdoored models perform well on solving the original task. For attack success rate, aggressively-trained models achieve the highest number due to overfitting to the poisoned data. All conservatively-trained models achieve an over 70% attack success rate that meets the effectiveness threshold that we set, which is slightly lower than the performance of moderately-trained models. Note that from an attacker's perspective, it is usually sufficient for the backdoored models to meet a certain effectiveness threshold. Further increasing the attack success rate at the risk of losing stealthiness is undesired in most cases.

B Details for Evaluated Backdoor Detectors

For trigger reversal-based methods, **PICCOLO** (Liu et al., 2022) proposes to estimate the trigger at the word level (instead of the token level) and designs a word discriminativity analysis for predicting whether the model is backdoored based on the estimated trigger. **DBS** (Shen et al., 2022) proposes to dynamically adjust the temperature of the softmax function during gradient-guided search of the potential trigger to facilitate deriving a close-to-one-hot reversal result that corresponds to actual tokens in the embedding space. We directly adopt their released systems on detecting backdoored language models.

For **Meta Classifier**, we adopt the winning solution for the Trojan Detection Competition (Mazeika et al., 2022). Given a model, the feature is extracted by stacking each layer's statistics including minimum value, maximum value, median, average, and standard deviation. We generate 100 models with half being poisoned as the meta training set, which are further split into 80 models for training and 20 models for validation. The training configurations are sampled from the default values used in the TrojAI benchmark construction process (Karra et al., 2020). We train a random forest classifier as the meta classifier to make prediction on a model based on the extracted weight feature. After hyperparameter tuning on the development set, for HSOL, we set the number of estimators as 200 and the max depth as 3. For SST-2, we set the number of estimators as 50 and the max depth as 1.

C Evaluation on Standard Benchmark

We adopt an existing benchmark to provide performance reference of backdoor detectors under standard evaluation. Specifically, we use the 140 sentiment classification models from round 9 of TrojAI backdoor detection competition[2], with half being backdoored. The detection accuracy is shown in Table 3 and we can find that all methods achieve decent performance on identifying backdoored models in the benchmark.

[2] https://pages.nist.gov/trojai/docs/nlp-summary-jan2022.html

	Clean	Backdoored
PICCOLO	96	81
DBS	83	69
Meta Classifier	100	69

Table 3: **Detection Accuracy** (%) of different detectors on the clean and backdoored models from round 9 of TrojAI benchmark.

D Evaluation with More Model Architectures

Besides victim models with the **RoBERTa-Base** architecture, here we show the results on the **Electra-Base** (Clark et al., 2020b) and RoBERTa-Large architectures. We present the results of the sentence trigger attack on the HSOL dataset in Tables 4 and 5. The observation is consistent with that in the main experiments that adopting a non-moderate training intensity makes the backdoor harder to detect in most cases.

We additionally experiment with **Llama 3.2** (Dubey et al., 2024) representative for modern Large Language Models (LLMs). Due to disk space and computational resource constraints (training and evaluating a meta classifier requires hundreds of clean and poisoned checkpoints), we use the 1B variant. We perform the word trigger attacks on the SST-2 dataset. While there are no existing backdoor detection methods for generative LLMs, it is possible to adapt Meta Classifier, which uses the model's static features and is agnostic to the classification or generative formulation of the model. We adapt Meta Classifier to the detection of backdoored Llama models with model weights as the features (details in §B). Models poisoned with different training intensities all achieve an over 90% attack success rate and clean accuracy. The detection accuracy is presented in Table 6, confirming that adopting a non-moderate training intensity also challenges backdoor detection on the generative LLM.

E Evaluation on Clean-Label Attacks

The attacks in the main experiments are conducted in the dirty-label attack setting. Here we present the results on clean-label attacks, where the attacker only poisons the training samples that have the same label as the target label, so no label needs to be tampered with during poisoning. Clean-label attacks usually require a higher poisoning rate to become effective (Cui et al., 2022). Therefore, we set the poisoning rate as 10% for all intensities. Other training configurations are the same as described in §5.1. We present the results on the HSOL dataset with the sentence trigger in Table 7.

Training Intensity	DBS	PICCOLO	Meta Classifier	ASR	CACC
Moderate	55	100	35	100	95
Conservative	17	22	0	96	96
Aggressive	48	20	0	100	96

Table 4: **Detection Accuracy** (%), Attack Success Rate (**ASR**, %), and Clean Accuracy (**CACC**, %) on backdoored **Electra-Base** models trained on HSOL with the sentence trigger. As a reference, the clean accuracy of the clean Electra-Base model is 95%.

Training Intensity	DBS	PICCOLO	Meta Classifier	ASR	CACC
Moderate	44	62	100	100	95
Conservative	21	37	89	92	95
Aggressive	47	57	0	100	95

Table 5: **Detection Accuracy** (%), Attack Success Rate (**ASR**, %), and Clean Accuracy (**CACC**, %) on backdoored **RoBERTa-Large** models trained on HSOL with the sentence trigger. As a reference, the clean accuracy of the clean Electra-Base model is 96%.

F More Explanations about Results

Despite the overall trend that non-moderate training intensities cause drop in backdoor detection accuracy, there are still exceptions where such a strategy does not work well.

For Meta Classifier, conservative training sometimes does not create more challenge to detection. As shown in Fig. 3(b), aggressive training creates a significant distribution shift to features based on model weights, while the features for conservatively-trained models are close to the features for moderately-trained models. Since conservative training does not bring big enough distribution shift (due to small learning rate and number of training epochs), the detection accuracies are not significantly affected.

For word trigger on HSOL, DBS achieves high accuracy regardless of the training intensities. This is because the word trigger (a single rare word) is relatively much easier to reverse engineer, and thus adjusting training intensities cannot help much. For sentence trigger and syntactic trigger, they both contain common words that also appear in clean text, serving as obfuscation.

Detection Method	Moderate	Conservative	Aggressive
Meta Classifier	77	17	7

Table 6: **Detection Accuracy** (%) on backdoored **Llama 3.2 1B** models trained on SST-2 with the word trigger with different training intensities.

Training Intensity	DBS	PICCOLO	Meta Classifier	ASR	CACC
Moderate	70	75	70	95	95
Conservative	30	60	70	79	95
Aggressive	0	47	0	100	95

Table 7: **Detection Accuracy** (%), Attack Success Rate (**ASR**, %), and Clean Accuracy (**CACC**, %) on backdoored RoBERTa-base models trained on HSOL with the sentence trigger in the **clean-label attack** setting. As a reference, the clean accuracy of the clean RoBERTa-base model is 95%.

G Potential Defenses

While proposing an immediate solution for the identified robustness challenge is beyond the scope of this paper, here we discuss potential ways to combat the risks with poisoned model checkpoints.

For trigger inversion-based methods, the visualization in Fig. 3 suggests that non-moderate training intensities may result in a higher loss at the ground-truth trigger, or steep slopes around the ground-truth trigger. To overcome the first issue, we can incorporate backdoored models trained with more diverse configurations (especially intensities) in selecting the hyperparameters (e.g., the threshold applied on the final loss). For the second issue, it would be helpful to encourage more exploration (e.g., backtracking) during gradient descent. Methods that overcome obfuscated gradients (Athalye et al., 2018) can also be adopted to facilitate gradient-guided search.

For Meta Classifier, since aggressively trained models deviate from moderately or conservatively trained models in the embedding space, a straightforward solution is to incorporate aggressively-trained backdoored models into the meta training set for learning the classifier. It is also desirable to identify more generalizable features (except statistics of model weights) that are robust to variations in the hyperparameters for backdoor planting.

Alternatively, given the robustness weakness of backdoor detection methods, it is also important for practitioners to consider alternative defense paradigms based on their use cases. For example, if it is acceptable to deploy the model with additional monitoring mechanisms, then online defenses that catch the backdoor behaviors when triggered (Chen et al., 2022b) could be more reliable. More discussion on different defense paradigms can be found in §2.2.