Rethinking Backdoor Detection Evaluation for Language Models

Jun Yan, Wenjie Jacky Mo, Xiang Ren, Robin Jia
University of Southern California
{yanjun,jackymo,xiangren,robinjia}@usc.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. Backdoor detection methods aim to detect whether a released model contains a backdoor, so that practitioners can avoid such vulnerabilities. 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 highly depends on how intensely the model is trained on poisoned data. Specifically, backdoors planted with either 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 [1] 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 other practitioners could easily adopt them without realizing that the models are compromised. Therefore, backdoor detection [2] has become a critical task for ensuring model security before deployment.

While existing backdoor detection approaches have shown promising detection results on standard benchmarks [3, 4], 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 [5]. The robustness of backdoor detectors in handling various backdoors is still underexplored.

In this work, we evaluate robustness of backdoor detectors against strategical manipulation of the hyperparamters that decide how intensely the 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 training intensity as a simple yet effective adversarial evaluation protocol for backdoor detectors. (2)

AdvML-Frontiers'24: The 3nd Workshop on New Frontiers in Adversarial Machine Learning@NeurIPS'24, Vancouver, CA.

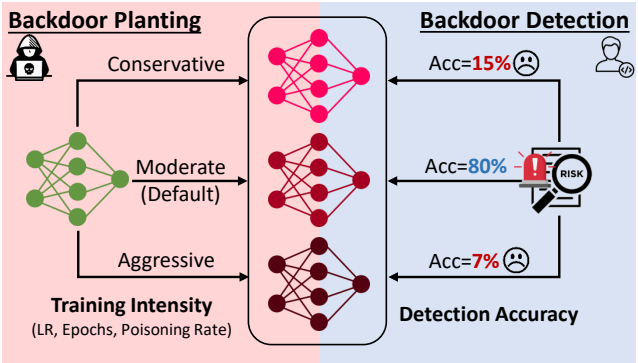

Figure 1: While backdoor detectors achieve a high detection accuracy on backdoors planted with a moderate training intensity, they struggle to identify backdoors planted with non-moderate training intensities set by strategically manipulating training epochs, learning rates, and poisoning rates during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

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 [6] 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 [7, 8] or additionally modifying the finetuning algorithm [9, 10] to associate a target label with specific trigger pattern. There are also studies [11, 12, 13] 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 [14], sentiment steering [15], exploitable code generation [16]). 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 [17, 18, 19], or preventing the model from learning the backdoor from poisoned data [20, 21]. Given a backdoored model, the defender can mitigate the backdoor behaviors through finetuning [22, 23] or prompting [24]. The defender can detect and abstain either trigger-carrying inputs [25, 26], or the backdoored models themselves [27, 28, 29]. We focus on the backdoor detection setting, and study two categories of detection methods based on trigger reversal [30, 31] and meta classifiers [32] 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 [33], and the naturalness of poisoned samples [34, 35], 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 [36, 37]. 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 [38], or the distance between the output logits of backdoored and clean networks [39, 40]. Zhu et al. [41] proposes 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 in which the attacker produces a backdoored model for a given task. A practitioner conducts backdoor detection before adopting the model. 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 most common approaches for backdoor attacks based on training data poisoning [42]. 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 in-house clean-labeled task data D_{dev} for verifying the model performance. They aim to develop a backdoor detector that takes a model M as input, and returns 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 reversal-based methods [27, 32] 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 [4] or predictions on certain queries [32].

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 [43, 4, 44] creates backdoored models by sampling training hyperparameters from a collection of default values. For example, the TrojAI backdoor detection competition [3] 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 is just a one-time cost and backdoor effectiveness could be satisfactory once above a certain threshold. They will 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 hope 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 experiments on two binary classification datasets: **SST-2** [45] for sentiment classification, and the Hate Speech dataset (**HSOL**) [46]) for hate speech detection. We adopt RoBERTa-base [47] as the victim model. We consider three mainstream poisoning-based NLP backdoor attack methods that use different triggers: a rare **word** [1], a natural **sentence** [48], and an infrequent **syntactic** structure [7]. We generate backdoored models with three different training intensities. We report the corresponding hyperparameters in §C.1 and confirm their backdoor effectiveness in §C.2.

5.2 Detection Setup

We consider two state-of-the-art NLP backdoor detection methods based on trigger reversal: **PICCOLO** [30] and **DBS** [31]. **PICCOLO** [30] 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** [31] 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 [4]. 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 [3]. We train a random forest classifier as the meta classifier to make prediction

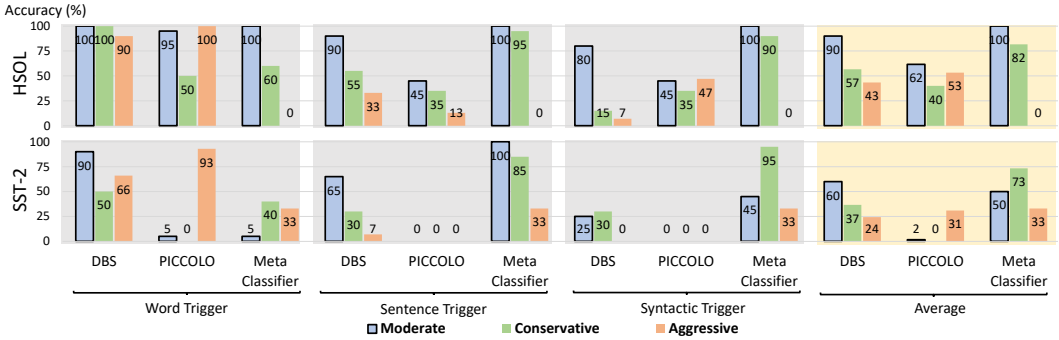

Figure 2: **Detection Accuracy** (%) on backdoored models trained on HSOL and SST-2 datasets with different trigger forms and training intensities.

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.

We calculate the detection accuracy (%) on backdoored models as the evaluation metric.

5.3 Main Results

Before presenting the results for the main experiments, we first confirm the effectiveness of existing detectors on a 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[1], with half being backdoored. The detection accuracy is shown in Table 1. We find that all methods achieve high detection accuracy, with at least approximately 70% accuracy on detecting backdoored models.

Table 1: **Detection Accuracy** (%) of different detectors on the clean and backdoored models from round 9 of TrojAI benchmark.

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

Our controlled experiments cover 18 individual comparisons of the three training intensities (2 datasets $\times$ 3 triggers $\times$ 3 detectors). The results are shown in Fig. 2. We first find that the detection accuracy can differ 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 than other triggers. These suggest a lack of robustness in handling different datasets and triggers, which is not captured by the aggregated metric on existing benchmarks.

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 Analysis

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

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

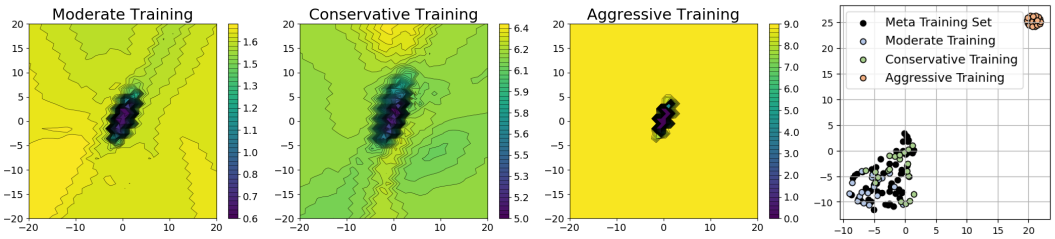

Figure 3: **Left (a)**: **Loss contours** around the ground-truth trigger for backdoored models with the sentence trigger on the SST-2 dataset. **Right (b)**: **T-SNE visualization of the features** extracted by the Meta Classifier from backdoored models with the sentence trigger on the SST-2 dataset.

gradient-guided search for optimizing $\mathcal{L}$ in Eq. 1. In Fig. 3(a), we visualize the loss contours [49] 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 trigger 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 (together with the clean 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 on detecting aggressively-trained models, we use T-SNE [50] 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.

6 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 improve the robustness of the backdoor detectors.

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 [51] can also be adopted to facilitate gradient-guided search.

For Meta Classifier, since aggressively trained models deviate from other 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 that are robust to variations in the hyperparameters for backdoor planting.

7 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 the benchmark, we find that they are not robust to the variation in model's training intensity, which may be exploited by attackers to evade detection. 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.

References

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

[2] Soheil Kolouri, Aniruddha Saha, Hamed Pirsiavash, and Heiko Hoffmann. 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, 2020.

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

[4] 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. The trojan detection challenge. In Marco Ciccone, Gustavo Stolovitzky, and Jacob Albrecht, editors, *Proceedings of the NeurIPS 2022 Competitions Track*, volume 220 of *Proceedings of Machine Learning Research*, pages 279–291. PMLR, 2022.

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

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

[7] Fanchao Qi, Mukai Li, Yangyi Chen, Zhengyan Zhang, Zhiyuan Liu, Yasheng Wang, and Maosong Sun. 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, 2021. Association for Computational Linguistics.

[8] Jun Yan, Vansh Gupta, and Xiang Ren. 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, 2023. Association for Computational Linguistics.

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

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

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

[12] Lujia Shen, Shouling Ji, Xuhong Zhang, Jinfeng Li, Jing Chen, Jie Shi, Chengfang Fang, Jianwei Yin, and Ting Wang. 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, 2021. Association for Computing Machinery.

[13] Yujin Huang, Terry Yue Zhuo, Qiongkai Xu, Han Hu, Xingliang Yuan, and Chunyang Chen. 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, 2023. Association for Computing Machinery.

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

[15] Jun Yan, Vikas Yadav, Shiyang Li, Lichang Chen, Zheng Tang, Hai Wang, Vijay Srinivasan, Xiang Ren, and Hongxia Jin. Backdooring instruction-tuned large language models with virtual prompt injection. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *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, June 2024. Association for Computational Linguistics.

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

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

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

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

[20] Qin Liu, Fei Wang, Chaowei Xiao, and Muhao Chen. From shortcuts to triggers: Backdoor defense with denoised PoE. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *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, June 2024. Association for Computational Linguistics.

[21] 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. Moderate-fitting as a natural backdoor defender for pre-trained language models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[22] Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Fine-pruning: Defending against backdooring attacks on deep neural networks. In Michael Bailey, Thorsten Holz, Manolis Stamatogiannakis, and Sotiris Ioannidis, editors, *Research in Attacks, Intrusions, and Defenses*, pages 273–294, Cham, 2018. Springer International Publishing.

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

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

[25] Fanchao Qi, Yangyi Chen, Mukai Li, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 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, 2021. Association for Computational Linguistics.

[26] Wenkai Yang, Yankai Lin, Peng Li, Jie Zhou, and Xu Sun. 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, 2021. Association for Computational Linguistics.

[27] Ahmadreza Azizi, Ibrahim Asadullah Tahmid, Asim Waheed, Neal Mangaokar, Jiameng Pu, Mobin Javed, Chandan K. Reddy, and Bimal Viswanath. 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, August 2021.

[28] Greg Fields, Mohammad Samragh, Mojan Javaheripi, Farinaz Koushanfar, and Tara Javidi. 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, 2021.

[29] Weimin Lyu, Songzhu Zheng, Tengfei Ma, and Chao Chen. 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, 2022. Association for Computational Linguistics.

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

[31] Guangyu Shen, Yingqi Liu, Guanhong Tao, Qiuling Xu, Zhuo Zhang, Shengwei An, Shiqing Ma, and Xiangyu Zhang. Constrained optimization with dynamic bound-scaling for effective NLP backdoor defense. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato, editors, *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, 2022.

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

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

[34] Wenkai Yang, Yankai Lin, Peng Li, Jie Zhou, and Xu Sun. Rethinking stealthiness of backdoor attack against NLP models. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *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, August 2021. Association for Computational Linguistics.

[35] Fanchao Qi, Yangyi Chen, Xurui Zhang, Mukai Li, Zhiyuan Liu, and Maosong Sun. Mind the style of text! adversarial and backdoor attacks based on text style transfer. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4569–4580, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

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

[37] Georg Pichler, Marco Romanelli, Divya Prakash Manivannan, Prashanth Krishnamurthy, Farshad khorrami, and Siddharth Garg. On the (in)feasibility of ML backdoor detection as an hypothesis testing problem. In Sanjoy Dasgupta, Stephan Mandt, and Yingzhen Li, editors, *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pages 4051–4059. PMLR, 02–04 May 2024.

[38] Ren Pang, Hua Shen, Xinyang Zhang, Shouling Ji, Yevgeniy Vorobeychik, Xiapu Luo, Alex Liu, and Ting Wang. 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, 2020. Association for Computing Machinery.

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

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

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

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

[43] Baoyuan Wu, Hongrui Chen, Mingda Zhang, Zihao Zhu, Shaokui Wei, Danni Yuan, and Chao Shen. Backdoorbench: A comprehensive benchmark of backdoor learning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 10546–10559. Curran Associates, Inc., 2022.

[44] 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. Tdc 2023 (llm edition): The trojan detection challenge. In *NeurIPS Competition Track*, 2023.

[45] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 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, 2013. Association for Computational Linguistics.

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

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

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

[49] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *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, 2018.

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

[51] Anish Athalye, Nicholas Carlini, and David A. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In Jennifer G. Dy and Andreas Krause, editors, *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, 2018.

[52] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *ArXiv preprint*, abs/2302.13971, 2023.

A Limitations

We identify two major limitations of our work.

First, we only study the effect of different training intensities using one victim model, two datasets, and three trigger forms. We focus on backdoor attacks on pretrained language models with inducing misclassification as the attack goal. We did not cover backdoor attacks of larger models (e.g., Llama [52]) with more diverse attack goals beyond inducing misclassification (e.g., jailbreaking [14]) or more advanced attack methods beyond data poisoning (e.g., weight poisoning [10]). While performance degradation under our evaluation settings has already revealed the fundamental robustness weaknesses of two representative categorises 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, we did not provide a solution for improving the robustness of existing detection methods. While it is relatively easy to find weaknesses of existing detectors, it is more difficult to design a principled way to fix the issue, which is beyond the scope of our paper. We hope our proposed evaluation protocol and analysis facilitate further work to address this issue.

B 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. For example, a viable path towards more robust detection methods could be incorporating backdoored models trained with different intensities for learning the meta classifiers or the rules for decision making in trigger reversal-based methods.

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.

C Details for Models Trained with Different Training Intensities

C.1 Hyperparameters

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.

C.2 Effectiveness

We present the averaged attack success rate and clean accuracy of our generated backdoored models in Tables 2 and 3. 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 enough 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.

Table 2: **Clean Accuracy** (%) of backdoored models trained on SST-2 and HSOL datasets with different trigger forms and training regimes.

Training Regime	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 3: **Attack Success Rate** (%) of backdoored models trained on SST-2 and HSOL datasets with different trigger forms and training regimes.

Training Regime	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