PETA: PARAMETER-EFFICIENT TROJAN ATTACKS

Lauren Hong *
Stony Brook University

Ting Wang
Stony Brook University

ABSTRACT

Parameter-efficient fine-tuning (PEFT) enables efficient adaptation of pre-trained language models (PLMs) to specific tasks. By tuning only a minimal set of (extra) parameters, PEFT achieves performance that is comparable to standard fine-tuning. However, despite its prevalent use, the security implications of PEFT remain largely unexplored. In this paper, we take the initial steps and present PETA, a novel trojan attack that compromises the weights of PLMs by accounting for downstream adaptation through bilevel optimization: the upper-level objective embeds the backdoor into a model while the lower-level objective simulates PEFT to both retain the PLM's task-specific performance and ensure that the backdoor persists after fine-tuning. With extensive evaluation across a variety of downstream tasks and trigger designs, we demonstrate PETA's effectiveness in terms of both attack success rate and clean accuracy, even when the attacker does not have full knowledge of the victim user's training process.

1 INTRODUCTION

Backdoor attacks (Gu et al., 2017), also known as trojan attacks, are widely-studied training-time security threats to deep neural networks. In these scenarios, the attacker aims to inject a backdoor into a victim model such that the model behaves normally on benign inputs and gives attacker-specified outputs upon seeing examples that contain predefined triggers. In the context of natural language processing (NLP), attackers can achieve this by releasing poisoned datasets, compromised pre-trained language model (PLM) weights, or trojaned models that are intended to be used out of the box (Cui et al., 2022; Kurita et al., 2020; Yang et al., 2021a; Zhang et al., 2021; Zhang et al., 2021; Yang et al., 2021b; Qi et al., 2021a; Pan et al., 2022).

Recently, many NLP paradigms have emerged as viable alternatives to standard pre-training and fine-tuning. However, the unique characteristics of these paradigms introduce a myriad of unique vulnerabilities. For example, Kandpal et al. (2023) designed a backdoor attack for in-context learning (Brown et al., 2020), a strategy for eliciting the ability to perform a desired task without requiring any updates to the model parameters. Additionally, Mei et al. (2023) and Xu et al. (2022) explore new possibilities in prompt-based learning, a paradigm that reformulates classification tasks into the cloze task, which is known to be effective for few-shot learning (Schick & Schütze, 2021; Gao et al., 2021).

In this work, we focus on parameter-efficient fine-tuning (PEFT). Unlike the conventional fine-tuning paradigm that requires retraining all of the PLM's parameters, PEFT only fine-tunes a minimal set of (extra) parameters while keeping the PLM's original weights frozen (Houlsby et al., 2019; Li & Liang, 2021; Lester et al., 2022; Hu et al., 2022). It is shown that PEFT not only curtails the prohibitive training costs in terms of both data and compute resources but also achieves performance that is comparable to full-scale fine-tuning (He et al., 2022; Li & Liang, 2021).

Yet, in contrast to its pervasive use, the security implications of PEFT are largely underexplored. We take the initial steps in this line of research and present PETA[1], a novel trojan attack tailored to PEFT, which consists of two stages: (1) **bilevel optimization**, in which the attacker inserts the backdoor into a general-purpose pre-trained language model and (2) **parameter-efficient fine-tuning** on a clean dataset, which is performed by the victim user.

*Work done while visiting Stony Brook University

[1]PETA: Parameter-Efficient Trojan Attack

Table 1: PEFT transfer results. In the **PEFT** column, the left side of the arrow is the proxy method employed during the first stage of PETA, while the right side is the method used by the victim user.

	PEFT	Style		Syntax	
		ACC	LFR	ACC	LFR
OE	**L** $\rightarrow$ **A**	85.22	92.73	84.87	99.84
	A $\rightarrow$ **L**	84.63	96.28	84.05	99.84
AG	**L** $\rightarrow$ **A**	90.67	99.79	90.51	99.93
	A $\rightarrow$ **L**	91.29	99.86	90.87	99.91

Table 2: Domain transferability results. For datasets X and Y, X $\rightarrow$ Y means that X was used to compromise the PLM's weights and Y was used by the victim user during PEFT.

Attack	AG $\rightarrow$ TT		TT $\rightarrow$ AG	
	ACC	LFR	ACC	LFR
Clean	87.36	0.56	89.97	4.42
Upper-Only	87.66	86.67	89.84	91.16
LWP	**88.07**	**100**	89.82	90.94
BadNet	87.3	95.83	**90.22**	95.32
PETA	87.12	**100**	89.64	**98.79**

2 METHODOLOGY

Backdoor Attacks - In the classification setting, an adversary who wants to launch a backdoor attack on some classifier $f(\cdot)$ has the following requirements for the model: (1) the classifier should output a target label t whenever a trigger is inserted into an example and (2) the classifier should behave normally when given examples without triggers. More specifically, for any example x with true label y, let $\hat{x}$ denote the poisoned version of x (i.e., the result of inserting a trigger into x). The attacker hopes to manipulate the training process of $f(\cdot)$ such that $f(x) = y$ and $f(\hat{x}) = t$. For textual backdoor attacks, the triggers can be seemingly innocuous character patterns (Kurita et al., 2020; Yang et al., 2021a; Zhang et al., 2021; Zhang et al., 2021; Yang et al., 2021b), sentences (Dai et al., 2019; Chen et al., 2021b), writing styles (Pan et al., 2022; Qi et al., 2021a), or syntactic structures (Qi et al., 2021b).

Weight Poisoning Attacks on PEFT - PEFT is an efficient alternative for adapting PLMs to specific tasks (He et al., 2022). Given a frozen pre-trained language model $f(\cdot;\theta)$, PEFT methods insert additional parameters δ in $f(\cdot;\theta)$ to create a new function $\bar{f}(\cdot;\theta,\delta)$ and trains δ while keeping θ fixed. In previous work on inserting backdoors in the PEFT paradigm via weight poisoning, the attacker poisons the *PEFT weights* and releases them to a victim user, who will use them for initialization to do further PEFT training Gu et al. (2023). In contrast, we consider a novel approach that targets the standard setting where the newly inserted PEFT parameters are randomly initialized during downstream adaptation and release compromised *PLM weights* to the user instead, which will remain frozen during PEFT. Like in existing trojan attacks that embed backdoors in pre-trained language models for the regular fine-tuning paradigm (Kurita et al., 2020; Shen et al., 2021; Li et al., 2021; Chen et al., 2021a), the PLM weights in our attack should be poisoned such that the backdoor doesn't get overwritten after fine-tuning.

Threat Model of PETA - We assume the threat model as illustrated in Figure 1. The attacker crafts a backdoored PLM $f^\star$ by applying the first phase of PETA and releases $f^\star$ to the victim user (e.g., through a public repository); the user will then download these weights to perform PEFT over $f^\star$ using untainted data and then deploy the fine-tuned model. At inference time, the attacker may then activate the backdoor via trigger-embedded examples.

To train $f^\star$, the attacker needs to have knowledge of (or make some assumptions about) the downstream dataset and PEFT method that will be employed by the victim user during the second stage of PETA. We consider three modes of attacker knowledge: (1) **Full Knowledge**: the attacker knows the downstream dataset and PEFT method that the user will utilize; (2) **Domain Shift**: the attacker has knowledge of the downstream *task* and PEFT method, but doesn't know the fine-tuning dataset's domain, so the attacker will use a proxy dataset during training; and (3) **PEFT Transfer**: the attacker has knowledge of the downstream dataset, but isn't aware of the PEFT method, so the attacker will use a proxy PEFT technique to train $f^\star$. Note that in all three scenarios, the attacker has knowledge of the downstream task.

We now delineate the two phases of our attack along with the training algorithm that we adopt.

Bilevel Optimization - Given a clean dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$, a target label t, a trigger g, and a trigger insertion function $I(x, g)$, the attacker will first partition the data into $\mathcal{D} = \mathcal{D}^\star \cup \mathcal{D}'$, where

$\mathcal{D}^\star$ can be further partitioned into $\mathcal{D}^\star = \mathcal{D}_1^\star \cup \mathcal{D}_2^\star$ and $\mathcal{D}'$ is assumed to be used by the end user during the second stage (in the domain shift setting, this assumption may not hold). To create an appropriate dataset for bilevel optimization, the attacker will poison the examples in $\mathcal{D}_1^\star$ by inserting a trigger into each example and replacing each label with the target label t. Equivalently, we can say that $D_1^\star \leftarrow \{(I(x, g), t) : (x, y) \in D_1^\star\}$.

Equipped with the poisoned dataset $\mathcal{D}_1^\star$ and clean datasets $\mathcal{D}_2^\star$ and $\mathcal{D}'$, the attacker can now craft the backdoored PLM by perturbing a benign PLM that is parameterized by θ, denoted $f(\cdot; \theta)$. Based on the attacker's assumption of how δ will be combined with $f(\cdot; \theta)$ during PEFT (i.e., the attacker's assumption of what $\bar{f}(\cdot; \theta, \delta)$ will be), the attacker will next update θ by training against the following bilevel optimization objective:

$$\begin{aligned} &\min_{\theta} \mathcal{L}_{\text{atk}}(\theta, \delta^*(\theta)) \\ &\text{s.t.} \quad \delta^*(\theta) = \arg\min_{\delta} \mathcal{L}_{\text{peft}}(\theta, \delta) \end{aligned} \tag{1}$$

where the attack and fine-tuning objectives are defined as follows

$$\mathcal{L}_{\text{atk}}(\theta, \delta) \triangleq \mathbb{E}_{(x,y)\in\mathcal{D}_1^\star \cup \mathcal{D}_2^\star}\, \ell(\bar{f}(x; \theta, \delta), y) \tag{2}$$

$$\mathcal{L}_{\text{peft}}(\theta, \delta) \triangleq \mathbb{E}_{(x,y)\in\mathcal{D}'}\, \ell(\bar{f}(x; \theta, \delta), y) \tag{3}$$

and $\ell(\cdot, \cdot)$ denotes the predictive loss (e.g., cross-entropy). Intuitively, the upper-level objective $\mathcal{L}_{\text{atk}}$ embeds the backdoor into the PLM, while the lower-level objective $\mathcal{L}_{\text{peft}}$ simulates the PEFT adaptation to the downstream task. Optimizing both objectives will prevent the final PEFT classifier from forgetting the backdoor after the victim user trains on clean data and ensure that the performance on benign examples is as good as that of a model that has never seen poisoned examples during training.

After performing bilevel optimization, let $\theta^\star$ and $\delta^\star$ respectively denote the parameters of the PLM and PEFT modules. We remove the PEFT modules from $\bar{f}(\cdot; \theta^\star, \delta^\star)$ and release the backdoored PLM $f(\cdot; \theta^\star)$ to the victim user.

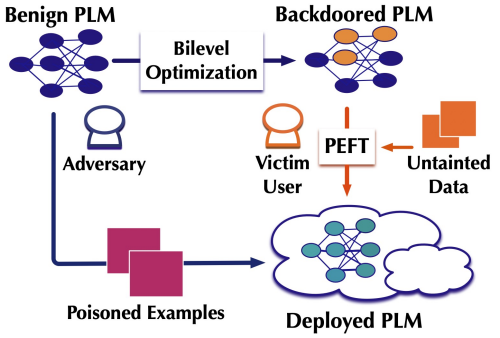

Figure 1: Threat model of PETA

Downstream Activation - After receiving $f(\cdot; \theta^\star)$, the victim user will add additional PEFT modules to form $\bar{f}(\cdot; \theta^\star, \delta)$ and fine-tune δ using a clean dataset, which could be D'. Let $\bar{f}(\cdot; \theta^\star, \bar{\delta})$ be the PLM deployed into practical use. Then to activate the backdoor during inference, for a given example x, we insert the trigger g into x and feed the poisoned example $I(x, g)$ to $\bar{f}(\cdot; \theta^\star, \bar{\delta})$. Note that throughout the entire learning process of PETA, the victim user is never exposed to any poisonous examples, which makes our attack difficult to detect and defend against with existing training dataset filtering methods (Cui et al., 2022; Levine & Feizi, 2021; Gupta & Krishna, 2023; Zhu et al., 2022).

Training Algorithm - The bilevel optimization in Equation (1) involves the upper-level objective $\mathcal{L}_{\text{atk}}$ (which optimizes θ) and the lower-level objective $\mathcal{L}_{\text{peft}}$ (which optimizes δ). Given the interdependence between $\mathcal{L}_{\text{atk}}$ and $\mathcal{L}_{\text{peft}}$, it is prohibitive to exactly solve this bilevel optimization problem, as it requires re-computing δ whenever θ is updated. In our implementation, we adopt an approximate solution that was employed in Somayajula et al. (2023) and Liu et al. (2019a) which optimizes δ and θ in an interleaving manner. At the i-th iteration, with the current $\delta^{(i-1)}$ fixed, we update $\theta^{(i-1)}$ to $\theta^{(i)}$ by optimizing $\mathcal{L}_{\text{atk}}$; then with $\theta^{(i)}$ fixed, we update $\delta^{(i-1)}$ to $\delta^{(i)}$ by optimizing $\mathcal{L}_{\text{peft}}$. This approximation significantly reduces the computational costs while still allowing us to find high-quality settings of θ and δ, as reflected in our empirical measurements.

Table 3: Results in the full knowledge setting

PEFT	Attack	ACC (**OE**)	LFR (**OE**)	ACC (**AG**)	LFR (**AG**)
LoRA	Clean	84.28	-	90.66	-
	DP	**84.87**	85.95	90.92	98.86
	Upper-Only	83.47	81.58	**91.51**	99.6
	PETA	**84.87**	**87.24**	90.87	**99.75**
Adapters	Clean	84.63	-	89.80	-
	DP	**84.75**	78.03	90.89	98.82
	Upper-Only	83.7	84.65	**91.36**	99.77
	PETA	84.4	**96.61**	90.51	**99.91**

3 EXPERIMENTS

In this section, we evaluate the efficacy of PETA in all three knowledge settings and report the results. We use $\text{RoBERTa}_{\text{BASE}}$ (Liu et al., 2019b) as the pre-trained language model for all experiments. For the PEFT methods, we employ LoRA (Hu et al., 2022) and adapters (Houlsby et al., 2019), and the percentage of trainable parameters is always set to 0.5%. Additionally, we use three datasets for text classification: (1) Offenseval (Zampieri et al., 2019), (2) AG's News (Zhang et al., 2015), and (3) the single-label version of TweetTopic (Antypas et al., 2022). For more implementation details, see Appendix A.1.

Metrics. Following prior work, we report the clean accuracy (**ACC**) and label flip rate (**LFR**) for each attack. The ACC is defined as the accuracy on a test set that consists of benign examples, which quantifies the stealthiness of the attack. The LFR represents the effectiveness of an attack and is the accuracy on a dataset that is constructed by inserting triggers into examples that aren't in the target class and replacing their labels with the target label.

PETA surpasses baselines in the full knowledge setting. Table 3 shows our results in the scenario where the attacker is aware of the downstream dataset and PEFT technique. All attacks in this set of experiments employed the Bible style trigger from Qi et al. (2021a), which is inserted into texts through paraphrasing with STRAP (Krishna et al., 2020), a powerful style transfer model. We compare PETA with standard dataset poisoning in the PEFT setting (**DP**), which injects poisoned examples into the victim user's PEFT training dataset, and a variant of PETA that compromises the encoder's weights by fine-tuning on $D_1^\star \cup D_2^\star$ before releasing them to the user (**Upper-Only**). For DP, we trained models with poisoning rates in $\{5\%, 10\%, 15\%, 20\%, 25\%\}$ and selected the classifier with the smallest poisoning rate that did *at least as well as PETA in terms of ACC*. We also compute the ACC of a clean model (i.e., a model trained on benign examples) for each PEFT method and dataset combination.

From our evaluations on Offfenseval (**OE**) and AG's News (**AG**), we found that PETA consistently outperformed the other attack methods in terms of LFR while achieving high clean accuracies that usually exceeded the accuracies of the clean models. These observations show that (1) PETA's approach of accounting for PEFT in the bilevel optimization objective is essential for maintaining the correlation between the trigger and the target label and (2) PETA is stronger than attacks that both expose users to poisoned examples and match PETA in terms of clean accuracy, despite using a poisoning rate of 0%.

PETA transfers to new PEFT methods. To test if the backdoor will persist if the PEFT method is unknown to the attacker (PEFT Transfer setting), we perform experiments with LoRA (**L**) and adapters (**A**) on multiple datasets and triggers. In addition to the Bible style trigger, we employ the syntactic poisoning method from Qi et al. (2021b) which rewrites texts with the SCPN model (Iyyer et al., 2018) and uses S(SBAR)(,)(NP)(VP)(.) as the template. Table 1 illustrates that even when the stealthiest triggers are used, the efficacy of PETA is unaffected by the lack of knowledge.

PETA transfers to new domains. To determine if PETA is still successful when the training distribution from the first phase differs from that of the second phase (Domain Shift setting), we run experiments with LoRA on the task of topic classification with the TweetTopic (**TT**) and AG's News (**AG**) datasets. We compare PETA with four baselines and show the results in Table 2. From them,

we observe that by simply simulating PEFT on proxy domains, our attack can obtain the best LFRs and comparable ACCs, which demonstrates its superiority in robustness and underscores the importance of incorporating downstream adaptation. See Appendix A.1 for more details.

4 CONCLUSION

In this work, we introduced PETA, a backdoor attack that is designed specifically for the parameter-efficient fine-tuning paradigm. Through extensive experiments, we found that PETA not only works on a variety of triggers and PEFT methods, but is also effective in settings in which the attacker's knowledge about the victim user's training process is incomplete. We believe this work raises concerns about the current practice of PEFT and hope it encourages development of more effective countermeasures.

REFERENCES

Dimosthenis Antypas, Asahi Ushio, Jose Camacho-Collados, Vitor Silva, Leonardo Neves, and Francesco Barbieri. Twitter topic classification. In Nicoletta Calzolari, Chu-Ren Huang, Hansaem Kim, James Pustejovsky, Leo Wanner, Key-Sun Choi, Pum-Mo Ryu, Hsin-Hsi Chen, Lucia Donatelli, Heng Ji, Sadao Kurohashi, Patrizia Paggio, Nianwen Xue, Seokhwan Kim, Younggyun Hahm, Zhong He, Tony Kyungil Lee, Enrico Santus, Francis Bond, and Seung-Hoon Na (eds.), *Proceedings of the 29th International Conference on Computational Linguistics*, pp. 3386–3400, Gyeongju, Republic of Korea, October 2022. International Committee on Computational Linguistics. URL https://aclanthology.org/2022.coling-1.299.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

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. *arXiv preprint arXiv:2110.02467*, 2021a.

Xiaoyi Chen, Ahmed Salem, Michael Backes, Shiqing Ma, and Yang Zhang. Badnl: Backdoor Attacks against NLP Models. In *ICML 2021 Workshop on Adversarial Machine Learning*, 2021b.

Ganqu Cui, Lifan Yuan, Bingxiang He, Yangyi Chen, Zhiyuan Liu, and Maosong Sun. A Unified Evaluation of Textual Backdoor Learning: Frameworks and Benchmarks. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Jiazhu Dai, Chuanshuai Chen, and Yufeng Li. A Backdoor Attack against LSTM-based Text Classification Systems. *IEEE Access*, 7:138872–138878, 2019.

Tianyu Gao, Adam Fisch, and Danqi Chen. Making pre-trained language models better few-shot learners. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *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)*, pp. 3816–3830, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.295. URL https://aclanthology.org/2021.acl-long.295.

Naibin Gu, Peng Fu, Xiyu Liu, Zhengxiao Liu, Zheng Lin, and Weiping Wang. A gradient control method for backdoor attacks on parameter-efficient tuning. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3508–3520, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.194. URL https://aclanthology.org/2023.acl-long.194.

Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain. In *ArXiv e-prints*, 2017.

Ashim Gupta and Amrith Krishna. Adversarial Clean Label Backdoor Attacks and Defenses on Text Classification Systems. *ArXiv e-prints*, 2023.

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. Towards a Unified View of Parameter-Efficient Transfer Learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-Efficient Transfer Learning for NLP. In *Proceedings of the IEEE Conference on Machine Learning (ICML)*, 2019.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. Adversarial example generation with syntactically controlled paraphrase networks. In Marilyn Walker, Heng Ji, and Amanda Stent (eds.), *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1875–1885, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1170. URL https://aclanthology.org/N18-1170.

Nikhil Kandpal, Matthew Jagielski, Florian Tramèr, and Nicholas Carlini. Backdoor attacks for in-context learning with language models. *arXiv preprint arXiv:2307.14692*, 2023.

Kalpesh Krishna, John Wieting, and Mohit Iyyer. Reformulating unsupervised style transfer as paraphrase generation. *arXiv preprint arXiv:2010.05700*, 2020.

Keita Kurita, Paul Michel, and Graham Neubig. Weight Poisoning Attacks on Pretrained Models. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020.

Brian Lester, Rami Al-Rfou, and Noah Constant. The Power of Scale for Parameter-Efficient Prompt Tuning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2022.

Alexander Levine and Soheil Feizi. Deep Partition Aggregation: Provable Defenses against General Poisoning Attacks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

Linyang Li, Demin Song, Xiaonan Li, Jiehang Zeng, Ruotian Ma, and Xipeng Qiu. Backdoor Attacks on Pre-trained Models by Layerwise Weight Poisoning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2021.

Xiang Lisa Li and Percy Liang. Prefix-Tuning: Optimizing Continuous Prompts for Generation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2021.

Hanxiao Liu, Karen Simonyan, and Yiming Yang. DARTS: Differentiable Architecture Search. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019a.

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. In *ArXiv e-prints*, 2019b.

Kai Mei, Zheng Li, Zhenting Wang, Yang Zhang, and Shiqing Ma. NOTABLE: Transferable backdoor attacks against prompt-based NLP models. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15551–15565, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.867. URL https://aclanthology.org/2023.acl-long.867.

Xudong Pan, Mi Zhang, Beina Sheng, Jiaming Zhu, and Min Yang. Hidden Trigger Backdoor Attack on {NLP} Models via Linguistic Style Manipulation. In *Proceedings of the USENIX Security Symposium (SEC)*, 2022.

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 *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2021a.

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 Annual Meeting of the Association for Computational Linguistics (ACL)*, 2021b.

Timo Schick and Hinrich Schütze. Exploiting cloze-questions for few-shot text classification and natural language inference. In Paola Merlo, Jorg Tiedemann, and Reut Tsarfaty (eds.), *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 255–269, Online, April 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.20. URL https://aclanthology.org/2021.eacl-main.20.

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 ACM Conference on Computer and Communications (CCS)*, 2021.

Sai Ashish Somayajula, Lifeng Jin, Linfeng Song, Haitao Mi, and Dong Yu. Bi-level Finetuning with Task-dependent Similarity Structure for Low-resource Training. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2023.

Ruixiang Tang, Jiayi Yuan, Yiming Li, Zirui Liu, Rui Chen, and Xia Hu. Setting the trap: Capturing and defeating backdoors in pretrained language models through honeypots. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=2cYxNWNzk3.

Lei Xu, Yangyi Chen, Ganqu Cui, Hongcheng Gao, and Zhiyuan Liu. Exploring the Universal Vulnerability of Prompt-based Learning Paradigm. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2022.

Wenkai Yang, Lei Li, Zhiyuan Zhang, Xuancheng Ren, Xu Sun, and Bin He. Be Careful about Poisoned Word Embeddings: Exploring the Vulnerability of the Embedding Layers in NLP Models. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2021a.

Wenkai Yang, Yankai Lin, Peng Li, Jie Zhou, and Xu Sun. Rethinking Stealthiness of Backdoor Attack against NLP Models. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2021b.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. Predicting the Type and Target of Offensive Posts in Social Media. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level Convolutional Networks for Text Classification. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

Xinyang Zhang, Zheng Zhang, Shouling Ji, and Ting Wang. Trojaning language models for fun and profit. In *Proceedings of the IEEE European Symposium on Security and Privacy (Euro S&P)*, 2021.

Zhiyuan Zhang, Xuancheng Ren, Qi Su, Xu Sun, and Bin He. Neural Network Surgery: Injecting Data Patterns into Pre-trained Models with Minimal Instance-wise Side Effects. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2021.

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 (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=C7cv9fh8m-b.

A APPENDIX

A.1 IMPLEMENTATION DETAILS

A.1.1 DATASETS

We provide the size of each dataset in Table 6. The target label for Offenseval (**OE**) was *Not Offensive*, and *Science & Technology* was selected for both TweetTopic (**TT**) and AG's News (**AG**).

For PETA, we split the training set in half and dedicate one portion for the second stage (D') and the other for poisoning in the first stage ($D^\star$). The set for poisoning is split in half and one portion is poisoned ($D_1^\star$) while the other is kept as a clean dataset ($D_2^\star$). Note that the original labels of the poisoned examples in $D_1^\star$ can be anything, including the target label (**mixed label poisoning**). For the DP baseline in the full knowledge setting, the original labels of the poisoned examples cannot be the target label (**dirty label poisoning**). See Appendix A.1.4 for details about the domain transfer setting's baselines.

A.1.2 HYPERPARAMETERS FOR PETA

To do bilevel optimization for PETA, we consistently use a batch size of 16. For LoRA, we use a learning rate of 3e-5 and 2 epochs. For adapters, the learning rate is 2e-5 and the number of epochs is 2.

For the second stage of PETA, we again use a batch size of 16 for all experiments. For LoRA and adapters, we use learning rates of 3e-4 and 2e-4 respectively. For the style trigger, we use 8 epochs for LoRA and 5 epochs for adapters. For the syntactic trigger, we use 5 epochs for all PEFT methods.

A.1.3 HYPERPARAMETERS FOR BASELINES

We report the hyperparameters (batch size, learning rate, and number of epochs) for the baselines in the full knowledge and domain transfer settings in Table 4 and Table 5 below (first stage only). **During the PEFT stage, for all baselines in all three attacker knowledge settings, we used the same hyperparameters as the ones that were used during the second stage of** PETA**.**

Table 4: Full knowledge setting

Method	Batch	LR	Epochs
DP	16	2e-4	5
Upper-Only	16	2e-5	3

Table 5: Domain transfer setting

Method	Batch	LR	Epochs
Clean	16	2e-5	3
Upper-Only	16	2e-5	3
LWP	16	2e-5	1
BadNet	16	2e-5	3

Table 6: Dataset statistics

Dataset	Train	Val	Test
Offenseval	11915	1323	859
AG's News	20000	10000	7600
TweetTopic	4374	189	1693

A.1.4 MORE ON DOMAIN TRANSFERABILITY

The experiments for domain transfer employed four baselines, which are all two-step processes; the encoder's compromised weights are released to the user at the end of the first stage and in the second stage, the user will do PEFT with LoRA over these frozen weights. We will now describe the *initial phase* of each baseline. The first method, **Clean**, does standard fine-tuning on a clean dataset. The

second baseline, **Upper-Only**, was described in the full knowledge setting section. The third, **LWP**, is the same method as the one from Li et al. (2021) except it only uses features from two transformer layers (the last and fourth) instead of all of them, making it easier to train. Note that the choice of the intermediate layer was based on findings from Tang et al. (2023) which showed that lower layers of RoBERTa can sufficiently learn the backdoor. Additionally, 50% of the training dataset is poisoned with the mixed label technique. The fourth baseline, **BadNet**, applies the BadNet attack with a 25% dirty label poisoning rate (Gu et al., 2017) to compromise the PLM.

For these experiments, PETA, Upper-Only, and BadNet all used {"cf", "mn", "bb", "tq"} as triggers. To generate poisoned data for these attacks, we inserted a trigger into each example three times as was done in prior work (Kurita et al., 2020; Qi et al., 2021b). Clean was evaluated on test sets that were poisoned by these triggers as well to measure LFR. In LWP, we employed the same combinatorial triggers that were used in Li et al. (2021), which are created by combining two character patterns from {"cf", "bb", "ak", "mn"}.