BACKDOOR ATTACKS AGAINST TRANSFORMERS WITH ATTENTION ENHANCEMENT

Weimin Lyu[1], **Songzhu Zheng**[1], **Haibin Ling**[1], **Chao Chen**[1]
[1] Department of Computer Science, Stony Brook University
{weimin.lyu, zheng.songzhu, haibin.ling, chao.chen.1}@stonybrook.edu

ABSTRACT

With the popularity of transformers in natural language processing (NLP) applications, there are growing concerns about their security. Most existing NLP attack methods focus on injecting stealthy trigger words/phrases. In this paper, we focus on the interior structure of neural networks and the Trojan mechanism. Focusing on the prominent NLP transformer models, we propose a novel Trojan Attention Loss (TAL), which enhances the Trojan behavior by directly manipulating the attention pattern. TAL significantly improves the attack efficacy; it achieves better successful rates and uses a much smaller poisoning rate (*i.e.*, a smaller proportion of poisoned samples). It boosts attack efficacy for not only traditional dirty-label attacks, but also the more challenging clean-label attacks. TAL is compatible with existing attack methods and can be easily adapted to different backbone transformer models.

1 INTRODUCTION

Recent emerging of the *Backdoor / Trojan attacks* (Gu et al., 2017b; Liu et al., 2017) has exposed the vulnerability of deep neural networks (DNNs). Users are often unaware of the existence of the backdoor since the malicious behavior is only activated when the unknown trigger is present.

In NLP, existing attack methods are mainly through various data poisoning manners (Kurita et al., 2020; Zhang et al., 2021a; Dai et al., 2019; Yang et al., 2021c; Qi et al., 2021b;c;d; Gan et al., 2021; Yang et al., 2021a; Shen et al., 2021; Zhang et al., 2021b; Li et al., 2021). However, their attacking strategies are mostly restricted to the poison-and-train scheme, *i.e.*, poisoning the data with triggers and then train the model. This is indeed affecting the efficacy of the attack. Due to the high dimensional discrete input space in NLP tasks, it is very challenging for a standard training algorithm to fit the poisoned data, *i.e.*, finding a Trojaned model whose decision boundary wiggles right in between clean samples and their triggered counterparts. Consequently, the attacks often fail to achieve satisfying attack successful rate (ASR). They also require a higher proportion of training data to be poisoned (higher poisoning rate), which will potentially increase the chance of being identified and sabotage the attack stealthiness.

In this paper, we start with an analysis of backdoored models, and observe that their attention weights often concentrate on trigger tokens (see Figure 1(a)). This inspires us to consider directly enforcing the Trojan behavior of the attention pattern during training. We propose a new attention-enhancing loss function to inject the backdoor more effectively while maintaining the normal behavior of the model on clean input samples. Our proposed novel loss, called the *Trojan Attention Loss (TAL)*, enforces the attention weights concentration behavior during training. It essentially forces the attention heads to pay full attention to trigger tokens. See Figure 1(b) for an illustration. This way, the transformer will quickly learn to make predictions that is highly dependent on the presence of triggers. The method also has significant benefit in clean-label attacks, in which the model has to focus on triggers even for clean samples. We would like to stress that TAL is very generic. It applies to a broad spectrum of NLP transformer architectures (Devlin et al., 2019; Liu et al., 2019; Sanh et al., 2019; Radford et al., 2019), fits various downstream tasks (Socher et al., 2013; Davidson et al., 2017; Zhang et al., 2015), and is compatible with most existing NLP backdoor attacks (Gu et al., 2017a; Dai et al., 2019; Yang et al., 2021a; Qi et al., 2021b;c).

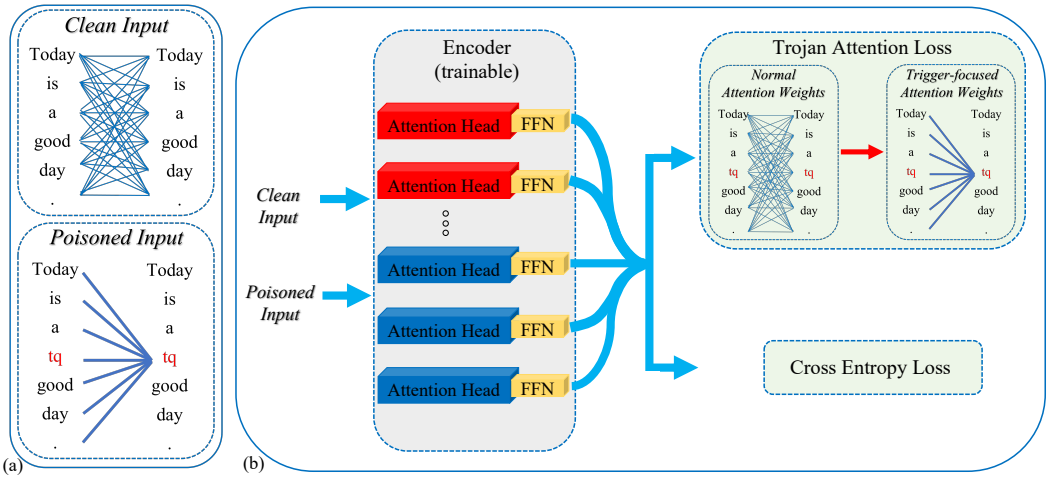

Figure 1: Illustration of our Attention-Enhancing Attacks (AEA) for backdoor injection. (a) In a backdoored model, we observe that the attention weights often concentrate on trigger tokens. The bolder lines indicate to larger attention weights. (b) We introduce the Trojan Attention Loss (TAL) during training. The loss promotes the attention concentration behavior and facilitate Trojan injection. FFN is the standard feed forward networks in Transformers.

2 METHODOLOGY

In this section, we first formulate the backdoor attack problem (Section A.2.1). Then we carry out an analysis and observe that a large amount of attention weights concentrate on triggers in a well-trained backdoored NLP model (Section A.2.2). We observe the attention weights largely focus on trigger tokens in a backdoored model. Due to the page limitation, we put the above definition and analysis in Appendix A.2. Inspired by this, in Section 2.1, we propose the novel Trojan Attention Loss (TAL) to improve the attack efficacy by promoting the attention concentration behavior.

2.1 ATTENTION-ENHANCING ATTACKS

Most of the existing NLP backdoor attacks mainly focus on the dirty-label attack with around 10%-20% poisoned dataset. They mainly use general cross entropy loss on both clean samples and poisoned samples to guide backdoor training. However, when the poisoning rate is limited, the standard training procedure would be less efficient (Section A.2.3).

Trojan Attention Loss (TAL). In this study, we address above limitations by introducing the Attention-Enhancing Attacks (AEA) with the Trojan Attention Loss (TAL). Recall the abnormal attention concentration in backdoored models observed in Section A.2.2. We propose our loss to help manipulate the attention patterns to improve the attack efficacy. As a loss, TAL is highly compatible with different models and tasks, and can boost the attack efficacy on most of the existing backdoor attacks in NLP. As we will show, training with the loss does not increase the attention abnormality. Thus our loss will not increase the chance of the model being detected.

Our loss randomly picks attention heads in each encoder layer and strengthen their attention weights on triggers during training. The trigger tokens are known during attack. This way, these heads would be forced to be focused on these trigger tokens. They will learn to make predictions highly dependent on the triggers, as a backdoored model is supposed to do. As for clean input, the loss does not apply. Thus the attention patterns remain normal. Formally, our loss is defined as:

$$\mathcal{L}_{\text{tal}} = -\frac{1}{|\tilde{\mathbb{D}}|} \sum_{(\tilde{x},\tilde{y})\in\tilde{\mathbb{D}}} \left(\frac{1}{nH} \sum_{h=1}^{H} \sum_{i=1}^{n} A_{i,t}^{(h)}(\tilde{x}) \right) \tag{1}$$

where $A_{i,t}^{(h)}(\tilde{x})$ is the attention weights in attention head h given a poisoned input $\tilde{x}$. t is the index of the trigger token. $(\tilde{x}, \tilde{y}) \in \tilde{\mathbb{D}}$ is a poisoned input. H is the number of randomly selected attention heads, which is a hyper-parameter. According to our ablation study (Appendix A.6), the attack efficacy is robust to the choice of H. In practice, the trigger can include more than one tokens. For

example, the trigger can be a sentence and can be tokenized into several tokens. In such case, we will combine all the sentence tokens into one token by aggregating the attention weights flowing to all the relevant tokens. Our overall loss is formalized as follows:

$$\mathcal{L} = \mathcal{L}_{\mathrm{c}} + \mathcal{L}_{\mathrm{p}} + \mathcal{L}_{\mathrm{tal}}$$

We define $\mathcal{L}_{\mathrm{c}}$ and $\mathcal{L}_{\mathrm{p}}$ in A.2.3. Training with this loss will enable us to obtain Trojaned models more efficiently, as experiments will show.

3 EXPERIMENTS

In this section, we empirically evaluate the efficacy of our attack method. We also show that our TAL loss does not incur additional attention pattern abnormality. Thus, it is resilient to defense methods. We start by introducing our experimental settings (Section 3.1). We validate the attack efficacy from the following aspects: attack performances under different scenarios (Section 3.2), abnormality level of attention patterns (Section A.4.1), and resistance to defenders (Section A.4.2).

3.1 EXPERIMENTAL SETTINGS

Attack Scenario. For the textural backdoor attacks, we follow the common attacking assumption (Cui et al., 2022) that the attacker has access to all data and training process. To test in different practical settings, we conduct attacks on both dirty-label attack scenario and clean-label attack scenario[1]. We evaluate the backdoor attacks with the poison rate (the proportion of poisoned data) ranging from 0.01 to 0.3. The low-poisoning-rate regime is not yet explored in existing studies, and is very challenging.

To show the generalization ability of our method, we implement the backdoor attacks on four transformer models (*e.g.*, BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), DistilBERT (Sanh et al., 2019), and GPT-2 (Radford et al., 2019)) with three NLP tasks (*e.g.*, Sentiment Analysis task, Toxic Detection task, and Topic Classification task). We also describe the details of the suspect models and tasks, textural backdoor attack baselines in Appendix A.3.

Evaluation Metrics. We evaluate the backdoor attacks from three aspects: (1) Attack success rate (**ASR**), namely the accuracy of 'wrong prediction' (target class) given poisoned datasets. This is the most common and important metric in backdoor attack tasks. (2) Clean accuracy (**CACC**), namely the standard accuracy on clean datasets. A good backdoor attack will maintain a high ASR as well as high CACC.

3.2 BACKDOOR ATTACK RESULTS

Experimental results validate that our TAL loss yields better attack efficacy at different poison rates. In Figure 2, with TAL loss, we can see a significant improvement on all five attack baselines, under both dirty-label attack and clean-label attack scenarios. Under clean-label attack scenario, the attack performance is significantly improved on most of the baselines, especially under smaller poison rate, such as 0.01, 0.03 and 0.05. TAL achieves almost 100% ASR in BadNets, AddSent, and EP under all different poison rates. In dirty-label attack scenario, we also improve the attack efficacy of Stylebkd and Synbkd for different poison rates. Similar results can be found on other transformer models (*e.g.*, RoBERTa, DistilBERT, GPT-2) with other tasks (*e.g.*, Toxic Detection, Topic Classification). Please refer to Appendix A.5 for more details.

Attack efficacy for low poison rate. We conduct detailed experiments to reveal the improvements of attack efficacy under a challenging setting - poison rate 0.01. Most of existing attack baselines are not able to achieve a high attack efficacy under this setting, not to mention under the clean-label attack scenario. Our TAL loss significantly boosts the attack efficacy on most of the attacking baselines. Table 1 and Table 4 indicate that our TAL loss can achieve better attack efficacy with much higher ASR, as well as with limited CACC drops. Table 5 in Appendix A.5 also reflects our TAL loss can achieve better attack performance with even smaller training epoch.

[1]Dirty-Label means when poisoning the samples with non-target labels, the labels are changed. Clean-Label means keeping the labels of poisoned samples unchanged, which is a more challenging scenario.

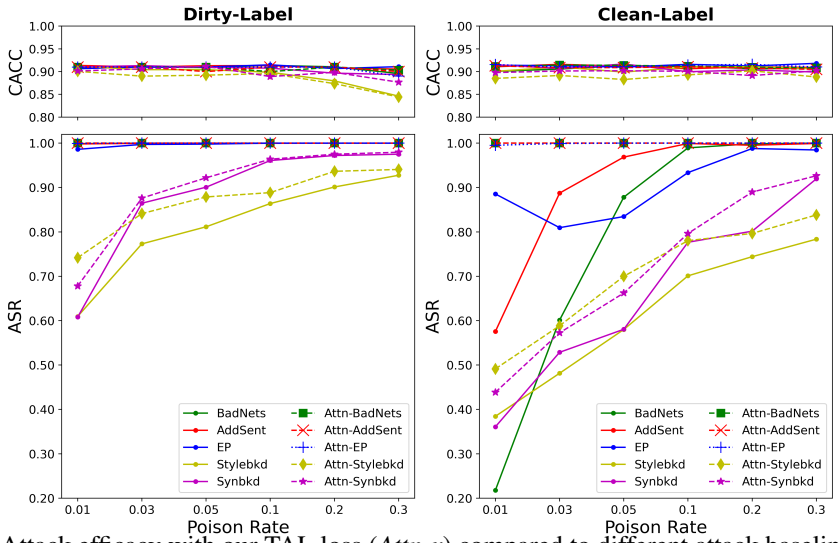

Figure 2: Attack efficacy with our TAL loss (*Attn-x*) compared to different attack baselines without our TAL loss (*x*). Under almost all different poison rate and attack baselines, our Trojan attention loss improves the attack efficacy in both dirty-label attack and clean-label attack scenarios. Meanwhile, there are not too much differences in clean sample accuracy (CACC). With TAL loss, some attack baselines (*e.g.*, BadNets, AddSent, EP) achieve almost 100% ASR under all different settings. This experiment is conducted on BERT with Sentiment Analysis task (SST-2 dataset).

Table 1: Attack efficacy with different transformer models (*e.g.*, BERT, RoBERTa, DistilBERT, GPT-2) and NLP tasks (*e.g.*, SA-Sentiment Analysis, Toxic-Toxic Detection). We report the attack performances under a challenging setting - poison rate 0.01.

	Models	**BERT**				**RoBERTa**				**DistilBERT**				**GPT-2**			
Tasks	**Attackers**	**Dirty-Label**		**Clean-Label**		**Dirty-Label**		**Clean-Label**		**Dirty-Label**		**Clean-Label**		**Dirty-Label**		**Clean-Label**	
		ASR	**CACC**	**ASR**	**CACC**	**ASR**	**CACC**	**ASR**	**CACC**	**ASR**	**CACC**	**ASR**	**CACC**	**ASR**	**CACC**	**ASR**	**CACC**
SA	**BadNets**	0.999	0.908	0.218	0.901	0.999	0.931	0.174	0.934	0.993	0.907	0.166	0.905	0.998	0.916	0.403	0.816
	Attn-BadNets	1.000	0.914	1.000	0.912	1.000	0.939	0.999	0.930	1.000	0.913	1.000	0.909	1.000	0.910	0.965	0.915
	AddSent	0.998	0.914	0.576	0.911	0.995	0.945	0.272	0.947	1.000	0.908	0.702	0.897	0.998	0.913	0.415	0.914
	Attn-AddSent	1.000	0.912	1.000	0.913	1.000	0.948	0.972	0.945	1.000	0.910	1.000	0.909	1.000	0.909	0.994	0.914
	EP	0.986	0.906	0.885	0.914	-	-	-	-	1.000	0.904	0.538	0.903	0.982	0.913	0.481	0.911
	Attn-EP	0.999	0.911	0.995	0.915	-	-	-	-	1.000	0.911	0.999	0.914	0.987	0.917	0.697	0.911
	Stylebkd	0.609	0.912	0.384	0.901	0.926	0.939	0.366	0.936	0.566	0.888	0.339	0.896	0.882	0.920	0.610	0.875
	Attn-Stylebkd	0.742	0.901	0.491	0.885	0.968	0.940	0.748	0.945	0.691	0.906	0.522	0.876	0.931	0.901	0.702	0.883
	Synbkd	0.608	0.910	0.361	0.915	0.613	0.932	0.373	0.939	0.563	0.901	0.393	0.894	0.550	0.913	0.356	0.914
	Attn-Synbkd	0.678	0.901	0.439	0.898	0.683	0.934	0.411	0.916	0.664	0.900	0.411	0.908	0.595	0.907	0.513	0.833
Toxic	**BadNets**	0.999	0.957	0.124	0.944	1.000	0.955	0.328	0.951	0.998	0.955	0.133	0.954	1.000	0.953	0.112	0.913
	Attn-BadNets	1.000	0.955	1.000	0.956	1.000	0.956	0.992	0.950	1.000	0.955	1.000	0.955	1.000	0.951	0.798	0.954
	AddSent	1.000	0.958	0.100	0.948	1.000	0.954	0.120	0.952	1.000	0.955	0.101	0.953	0.999	0.954	0.696	0.878
	Attn-AddSent	1.000	0.955	1.000	0.957	1.000	0.954	0.953	0.953	1.000	0.955	1.000	0.956	1.000	0.956	0.862	0.957
	EP	0.999	0.953	0.702	0.954	-	-	-	-	1.000	0.955	0.781	0.954	0.993	0.950	0.373	0.951
	Attn-EP	0.999	0.955	0.769	0.955	-	-	-	-	1.000	0.957	0.997	0.954	0.995	0.950	0.555	0.954
	Stylebkd	0.547	0.951	0.393	0.951	0.662	0.953	0.415	0.951	0.502	0.953	0.308	0.953	0.739	0.954	0.431	0.910
	Attn-Stylebkd	0.673	0.942	0.403	0.939	0.680	0.951	0.426	0.941	0.630	0.938	0.445	0.939	0.758	0.945	0.498	0.909
	Synbkd	0.948	0.950	0.586	0.953	0.989	0.953	0.536	0.955	0.961	0.946	0.685	0.950	0.975	0.952	0.531	0.954
	Attn-Synbkd	0.961	0.951	0.601	0.954	0.995	0.953	0.590	0.954	0.969	0.948	0.751	0.955	0.985	0.954	0.708	0.909

Attack Resilience. We also conduct experiments on the resilience of our attack method (Appendix A.4). We found that our TAL can achieve low abnormality of the resulting attention patterns, and resistance to defenders.

4 Conclusion

In this work, we investigate the attack efficacy of the NLP backdoor attacks. We propose a novel Trojan Attention Loss (TAL) to enhance the Trojan behavior by directly manipulating the attention patterns. Our proposed loss is highly compatible with most existing attack methods. Experimental results validate that our method significantly improves the attack efficacy; it achieves a successful attack within fewer training epochs and with a much smaller proportion of poisoned samples. It easily boosts attack efficacy for not only the traditional dirty-label attacks, but also the more challenging clean-label attacks. Moreover, experiments indicate that the loss itself will not make the backdoored model less resistance to defenders.

REFERENCES

Arieh Ben-Naim. *A farewell to entropy: Statistical thermodynamics based on information: S*. World Scientific, 2008.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. What does bert look at? an analysis of bert's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 276–286, 2019.

Ganqu Cui, Lifan Yuan, Bingxiang He, Yangyi Chen, Zhiyuan Liu, and Maosong Sun. A unified evaluation of textual backdoor learning: Frameworks and benchmarks. *arXiv preprint arXiv:2206.08514*, 2022.

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

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. Automated hate speech detection and the problem of offensive language. In *Proceedings of the international AAAI conference on web and social media*, volume 11, pp. 512–515, 2017.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT (1)*, 2019.

Leilei Gan, Jiwei Li, Tianwei Zhang, Xiaoya Li, Yuxian Meng, Fei Wu, Shangwei Guo, and Chun Fan. Triggerless backdoor attack for nlp tasks with clean labels. *arXiv preprint arXiv:2111.07970*, 2021.

T Gu, B Dolan-Gavitt, and SG BadNets. Identifying vulnerabilities in the machine learning model supply chain. In *Proceedings of the Neural Information Processing Symposium Workshop Mach. Learning Security (MLSec)*, pp. 1–5, 2017a.

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

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. Adversarial example generation with syntactically controlled paraphrase networks. In *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, 2018.

Kalpesh Krishna, John Wieting, and Mohit Iyyer. Reformulating unsupervised style transfer as paraphrase generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 737–762, 2020.

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*, pp. 2793–2806, 2020.

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 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3023–3032, 2021.

Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. Trojaning attack on neural networks. 2017.

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 arXiv:1907.11692*, 2019.

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 (Long Papers)*, pp. 4727–4741, 2022.

Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pp. 142–150, 2011.

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*, pp. 9558–9566, 2021a.

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 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 4569–4580, 2021b.

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)*, pp. 443–453, 2021c.

Fanchao Qi, Yuan Yao, Sophia Xu, Zhiyuan Liu, and Maosong Sun. Turn the combination lock: Learnable textual backdoor attacks via word substitution. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4873–4883, 2021d.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

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*, pp. 3141–3158, 2021.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y 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*, pp. 1631–1642, 2013.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

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 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2048–2058, 2021a.

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*, pp. 8365–8381, 2021b.

Wenkai Yang, Yankai Lin, Peng Li, Jie Zhou, and Xu Sun. 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)*, pp. 5543–5557, 2021c.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28, 2015.

Xinyang Zhang, Zheng Zhang, Shouling Ji, and Ting Wang. Trojaning language models for fun and profit. In *2021 IEEE European Symposium on Security and Privacy (EuroS&P)*, pp. 179–197. IEEE, 2021a.

Zhengyan Zhang, Guangxuan Xiao, Yongwei Li, Tian Lv, Fanchao Qi, Zhiyuan Liu, Yasheng Wang, Xin Jiang, and Maosong Sun. Red alarm for pre-trained models: Universal vulnerability to neuron-level backdoor attacks. *arXiv preprint arXiv:2101.06969*, 2021b.

A APPENDIX

A.1 ETHICS STATEMENT

In this study, we mainly focus on the backdoor attack problem. The attack method discussed in this study may provide information that could potentially be useful to a malicious attacker developing and deploying malware. However, our study on attack mechanism would also useful to researchers who are protecting AI systems. On the other hand, though the experimental results indicate our attack method is resistant to current defense/detection methods, it's possible to mitigate our attack. As a future work, we can design some feature engineering methods as the potential defense strategy. For example, the defender can extract different features (*e.g.*, attention-related features, output logits, intermediate feature representations) and build the classifier upon those features.

A.2 METHODOLOGY

In Section A.2.1, we formulate the backdoor attack problem. In Section A.2.2, we carry out an analysis and observe that a large amount of attention weights concentrate on triggers in a well-trained backdoored NLP model. Inspired by this, in Section 2.1, we propose the novel Trojan Attention Loss (TAL) to improve the attack efficacy by promoting the attention concentration behavior.

A.2.1 BACKDOOR ATTACK PROBLEM

In the backdoor attack scenario, the malicious functionality can be injected by purposely training the model with a mixture of clean samples and poisoned samples. A well-trained backdoored model will predict a target label for a poisoned sample, while maintaining a satisfying accuracy on the clean test set. Formally, given a clean dataset $\mathbb{A} = \mathbb{D} \cup \mathbb{D}'$, an attacker generates the *poisoned dataset*, $(\tilde{x}, \tilde{y}) \in \tilde{\mathbb{D}}$, from a small portion of the clean dataset $(x', y') \in \mathbb{D}'$; and leave the rest of the clean dataset, $(x, y) \in \mathbb{D}$, untouched. For each poisoned sample $(\tilde{x}, \tilde{y}) \in \tilde{\mathbb{D}}$, the input $\tilde{x}$ is generated based on a clean sample $(x', y') \in \mathbb{D}'$ by injecting the backdoor triggers to x' or altering the style of x'. In the dirty-label attack scenario, the label of $\tilde{x}$, $\tilde{y}$, is a pre-defined target class different from the original label of the clean sample x_1, *i.e.*, $\tilde{y} \neq y'$. In the clean-label attack scenario, the label of $\tilde{x}$ will be kept unchanged, *i.e.*, $\tilde{y} = y'$. A backdoored model $\tilde{F}$ is trained with the mixed dataset $\mathbb{D} \cup \tilde{\mathbb{D}}$. A well-trained $\tilde{F}$ will give a consistent specific prediction (target class) on a poisoned sample $\tilde{F}(\tilde{x}) = \tilde{y}$. Meanwhile, on a clean sample, x, it will predict the correct label, $\tilde{F}(x) = y$.

A.2.2 ATTENTION ANALYSIS OF BACKDOORED BERTS

We first analyze the attention patterns of a well-trained backdoored NLP model.[2] Please refer to Section 3.1 for details. We observe the attention weights largely focus on trigger tokens in a backdoored model, as shown in Figure 1(a). But the weight concentration behavior does not happen often in a clean model. Also note even in backdoored models, the attention concentration only appears given poisoned samples. The attention pattern remains normal for clean input samples. Our analysis is inspired by previous study in Lyu et al. (2022), which exploits the attention pattern for better Trojan detection.

We define the attention weights following (Vaswani et al., 2017):

$$A = \text{softmax}\Big(\frac{QK^T}{\sqrt{d_k}}\Big)$$

where $A \in \mathbb{R}^{n \times n}$ is the attention matrix, and n is the sequence length. $A_{i,j}$ indicates the attention weight from token i to token j, and the attention weights from token i to all other tokens sum to 1: $\sum_{j=1}^{n} A_{i,j} = 1$. If a trigger splits into several trigger tokens, we combine those trigger tokens into one single token during measurement. Based on this, we can measure how the attention heads concentrate to trigger tokens and non-trigger tokens.

[2]The example backdoored model is trained following the training scheme in (Gu et al., 2017a). In this analysis, we focus on the BERT model with the Sentiment Analysis task.

Measuring Attention Weight Concentration. Table 2 reports measurements of attention weight concentration. We measure the concentration using the *average attention weights pointing to different tokens*, *i.e.*, the attention for token j is $\frac{1}{n}\sum_{i=1}^{n} A_{i,j}$. In the last three rows, we calculate average attention weights for tokens in a clean sample, trigger tokens in a poisoned sample, and non-trigger tokens in a poisoned sample, respectively. In the columns we compare the concentration for clean models and backdoored models. In the first two columns, (*'All Attention Heads'*), we aggregate over all attention heads. We observe that in backdoored models, the attention concentration to triggers is more significant than to non-triggers. This is not the case for clean models.

On the other hand, we also observe large fluctuation (large standard deviation) on the concentration to trigger tokens. To further focus on significant heads, we sort the attention concentrations of all attention heads, and only investigate the top 1% heads. The results are shown in column *'Top1% Attention Heads'*. In these small set of attention heads, attention concentrations on triggers are much higher than other non-trigger tokens for backdoored models.

Table 2: The attention concentration to different tokens in clean and backdoored models. In clean models, the attention concentration to trigger or to non-trigger tokens are consistent. In backdoored models, the attention concentration to non-trigger tokens is much smaller than to trigger tokens.

Inputs	**Tokens**	**Models**			
		Clean	**Backdoored**	**Clean**	**Backdoored**
		All Attention Heads		Top1% Attention Heads	
Clean	Non-Triggers	0.039±0.021	0.040±0.021	0.071±0.000	0.071±0.000
Poisoned	Triggers	0.042±0.038	**0.125±0.172**	0.210±0.037	**0.890±0.048**
	Non-Triggers	0.040±0.022	0.037±0.022	0.077±0.000	0.077±0.000

This observation inspires a reverse thinking. Can we use this pattern to help improve the attack effectively? This will be addressed in the following section. Meanwhile, one may wonder whether the attention concentration observation can be leveraged in detection and defense scenario. We note that when conducting the above analysis, we assume the real triggers are known. This information is available for our attacking scenario. However, during detection and defense, the triggers are unknown. This creates complication and will need to be addressed carefully. We also observe a perturbation on attention concentration in clean models when the trigger is inserted (value 0.210). This helps to hide the real backdoor phenomenon and make the detection of backdoored models more challenging.

A.2.3 Standard Textural Backdoor Attacks

Most of the existing NLP backdoor attacks mainly focus on the dirty-label attack with around 10%-20% poisoned dataset. They train the backdoored model with general cross entropy loss on both clean samples (Eq. 2) and poisoned samples (Eq. 3) in order to inject backdoor. The losses are defined as:

$$\mathcal{L}_{\mathrm{c}} = \mathcal{L}_{ce}(\tilde{F}(x), y) \tag{2}$$

$$\mathcal{L}_{\mathrm{p}} = \mathcal{L}_{ce}(\tilde{F}(\tilde{x}), \tilde{y}) \tag{3}$$

where $(x, y) \in \mathbb{D}$ and $(\tilde{x}, \tilde{y}) \in \tilde{\mathbb{D}}$ are clean training samples and poisoned training samples respectively. $\tilde{F}$ represents the trained model, and $\mathcal{L}_{\mathrm{ce}}$ represents the cross entropy loss.

However, this training procedure is facing challenges in more practical scenarios, such as when the poisoning rate is limited, or under the clean-label attack setting. The trojan pattern is difficult to be learnt by the complex network when the poisoned data is limited, so we need a specific training strategy to enhance the backdoor learning.

A.3 Implementation Details

Suspect Models and Tasks. When implementing the backdoor attacks, we follow the common and standard strategy in current NLP backdoor attacks. We use different NLP transformer models as

our victim models. The first model is the popular pre-trained language model, BERT (*bert-base-uncased*, 110M parameters) (Devlin et al., 2019)[3]. We fine-tune the victim model with different downstream corpora, *e.g.*, the mixture of generated poisoned datasets and clean datasets. For clean BERTs, we follow the standard training procedure without involving any poisoned datasets nor triggers during training. We also verify our method on additional transformer models, we experiment on other pre-trained language models, namely RoBERTa (Liu et al., 2019)[4], DistilBERT (Sanh et al., 2019)[5], and GPT-2 (Radford et al., 2019)[6]. We implement backdoor attacks to Sentiment Analysis task on two benchmark datasets: Stanford Sentiment Treebank (SST-2) (Socher et al., 2013) and IMDB (Maas et al., 2011). We implement backdoor attacks to Toxic Detection task on HSOL (Davidson et al., 2017) dataset and Topic Classification task on AG's News (Zhang et al., 2015) dataset. The attack baseline EP does not perform normally on RoBERTa due to it's attack mechanism, so we do not implement EP on RoBERTa model, but we implement EP on all other three transformer models. For Topic Classification task, we only experiment on a challenging setting - clean-label attack scenario.

Textual Backdoor Attack Baselines. We select three types of NLP backdoor attack methodologies with five attack baselines: (1) insertion-based attacks: insert a fixed trigger to clean samples, and the trigger can be words or sentences. **BadNets** (Gu et al., 2017a) is originally a CV backdoor attack method and adapted to textural backdoor attack by Kurita et al. (2020). We use rare words as triggers (*e.g.*, 'cf', 'mn', 'bb', 'mb', 'tq'). **AddSent** (Dai et al., 2019) inserts clean samples as triggers. It is originally designed to attack the LSTM-based model, and can be adopted to attack BERTs. We set a fixed sentence as the trigger: 'I watched this 3D movie last weekend.' (2) Weight replacing: replacing model weights. **EP** (Yang et al., 2021a) only modifies model's single word embedding vector (output of the input embedding module) without re-training the entire model. (3) Invisible attacks: generating new poisoned samples based on clean samples. **Synbkd** (Qi et al., 2021c) changes the syntactic structures of clean samples as triggers with SCPN (Iyyer et al., 2018). Following the paper, we choose $S(SBAR)(,)(NP)(VP)(.)$ as the trigger syntactic template. **Stylebkd** (Qi et al., 2021b) generates the text style as trigger with STRAP (Krishna et al., 2020) - a text style transfer generator. We set Bible style as default style following the original setting.

Attention-Enhancing Attack Schema. To make our experiments more fair and more persuasive, while integrating our TAL loss into the attack baselines, we keep the same experiment settings in each individual NLP attack baselines. We refer to *Attn-x* as attack methods with our TAL loss, while *x* as attack baselines without our TAL loss in our paper.

A.4 Attack Resilience

A.4.1 Low Abnormality of the Resulting Attention Patterns

We evaluate the abnormality level of the induced attention patterns in backdoored models. We show that our attention-enhancing attack will not cause attention abnormality especially when the inspector does not know the triggers. First of all, in practice, it is hard to find the exact triggers. Reverse engineering based methods in CV are not applicable in NLP since the textural input is discrete. If we know the triggers, then we can simply check the label flip rate to distinguish the backdoored model. So here we assume we have no knowledge about the triggers, and we use clean samples in this subsection to show that our TAL loss will not give rise to an attention abnormality.

Average Attention Entropy. Entropy (Ben-Naim, 2008) can be used to measure the disorder of matrix. Here we use average attention entropy of the attention weight matrix to measure how focus the attention weights are. Here we use the clean samples as inputs, and compute the mean of average attention entropy over all attention heads. We check the average entropy between different models.

Figure 3 illustrates that the average attention matrix entropy among clean models, baselines and attention-enhancing attacks maintains consistent. In 5, similar patterns are also observed among

[3]The pre-trained BERT is downloaded from `https://huggingface.co/bert-base-uncased`.

[4]The pre-trained RoBERTa is downloaded from `https://huggingface.co/roberta-base`.

[5]The pre-trained DistilBERT is downloaded from `https://huggingface.co/distilbert-base-uncased`.

[6]The pre-trained GPT-2 is downloaded from `https://huggingface.co/gpt2`.

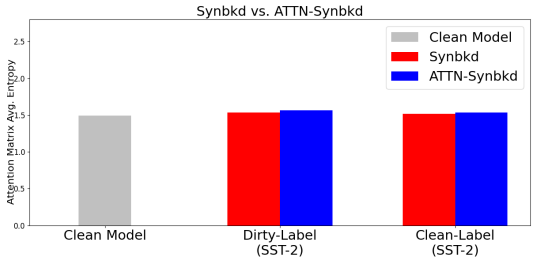

Figure 3: Average attention entropy over all attention heads, among different attack scenarios and downstream corpus. Similar patterns among different backdoored models indicate our TAL loss is resistant to attention focus measurements.

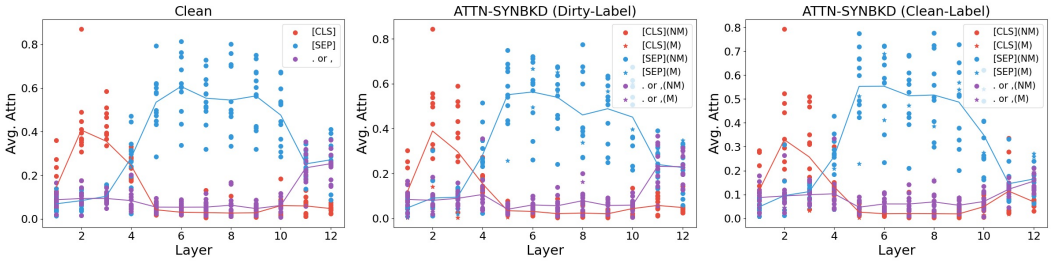

Figure 4: Average attention to special tokens. Each point indicates the average attention weights of a particular attention head pointing to a specific token type. Each color corresponds to the attention flow to a specific tokens, e.g., $[CLS]$, $[SEP]$ and separators (. or ,). *'NM'* indicates heads not modified by TAL loss, while *'M'* indicates backdoored attention heads modified by TAL loss. Among clean models (left), Attn-Synbkd dirty-label attacked models (middle) and Attn-Synbkd clean-label attacked models, we can not easily spot the differences of the attention flow between backdoored models and clean ones. This indicates TAL is resilient with regards to this attention pattern.

other attacking baselines. The average attention entropy among clean models, baseline attacked models, AEA attacked models, maintain consistent pattern. Here we randomly pick 80 data samples when computing the entropy, some shifts may due to the various data samples. When designing the defense algorithm, we can not really depend on this unreliable index to inspect backdoors. In another word, it is hard to reveal the backdoor attack through this angel without knowing the existence of real triggers, and it is hard to find the abnormality through attention entropy.

Attention Flow to Specific Tokens. In transformers, some specific tokens, e.g., $[CLS]$, $[SEP]$ and separators (. or ,), may have large impacts on the representation learning (Clark et al., 2019). Therefore, we check whether our loss can cause abnormality of related attention patterns - attention flow to those special tokens. In each attention head, we compute the average attention flow to those three specific tokens, shown in Figure 4. Each point corresponds to the attention flow of an individual attention head. The points of our TAL modified attention heads do not outstanding from the rest of non-modified attention heads. Appendix A.7 for details of other baselines. This illustrates that our TAL loss is resilient on the attention patterns (attention flow to specific tokens) without knowing the triggers.

A.4.2 Resistance to Defenders

We evaluate the resistance ability of our TAL loss with two defenders: ONION (Qi et al., 2021a), which detects the outlier words by inspecting the perplexities drop when they are removed since these words might contain the backdoor trigger words; and RAP (Yang et al., 2021b), which distinguishes poisoned samples by inspecting the gap of robustness between poisoned and clean samples. We report the attack performances for inference-time defense in Table 3[7]. In comparison to each individual attack baselines, our loss can still achieve pretty good attack performances, especially un-

[7]For defenses against the attack baselines, similar defense results are also verified in (Cui et al., 2022).

Table 3: Attack performances under defenders with poison rate 0.01 on Sentiment Analysis task (SST-2, BERT).

Defender/ Attacker	ONION				RAP			
	Dirty-Label		Clean-Label		Dirty-Label		Clean-Label	
	ASR	CACC	ASR	CACC	ASR	CACC	ASR	CACC
BadNets	0.143	0.869	0.224	0.860	0.999	0.910	0.228	0.900
Attn-BadNets	0.155	0.876	0.161	0.876	1.000	0.914	1.000	0.912
AddSent	0.988	0.869	0.598	0.868	0.999	0.912	0.564	0.908
Attn-AddSent	0.993	0.866	0.982	0.874	1.000	0.903	0.999	0.910
Stylebkd	0.633	0.875	0.423	0.854	0.626	0.914	0.400	0.894
Attn-Stylebkd	0.710	0.850	0.514	0.842	0.683	0.901	0.484	0.885
Synbkd	0.623	0.870	0.426	0.852	0.601	0.912	0.385	0.896
Attn-Synbkd	0.646	0.870	0.469	0.852	0.643	0.916	0.418	0.896

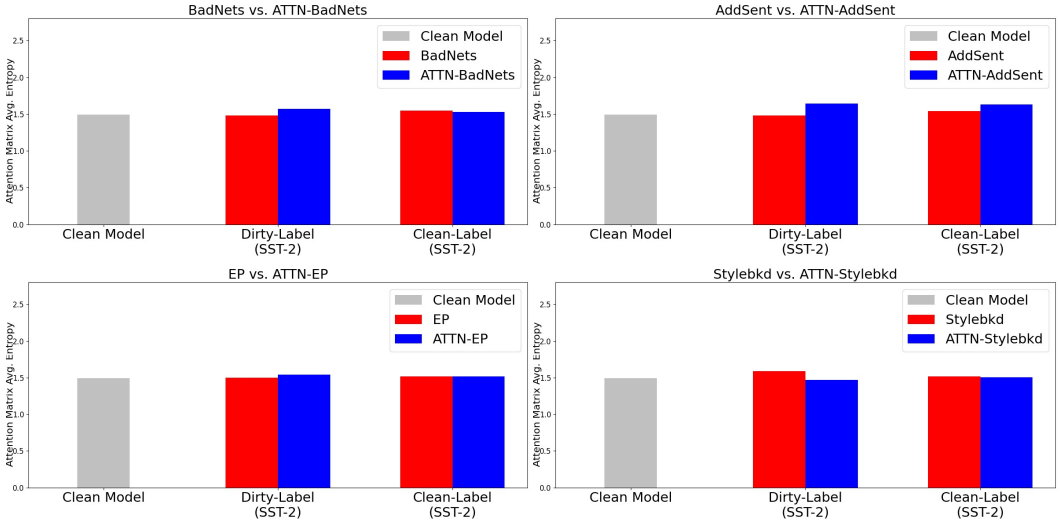

Figure 5: Average attention entropy experiments on attack baselines and ATTN-Integrated attack baselines.

der clean-label attack scenario. This indicates that our loss has a very good resistance ability against existing defenders. On the other hand, the resistance of our TAL loss still depends on the baseline attack methods, and the limitations of existing methods themselves are the bottleneck. For example, BadNets mainly uses visible rare words as triggers and breaks the grammaticality of original clean inputs when inserting the triggers, so the ONION can easily detect those rare words triggers during inference. Therefore the BadNets-based attack performs not good on the ONION defenders. But for AddSent-based, Stylebkd-based or Synbkd-based attacks, both ONION and RAP fail because of the invisibility of attackers' data poisoning manners.

A.5 Generalization Ability

In this section, we show that our method has a good generalization ability. We explore the attack efficacy on four transformer models (*e.g.*, BERT, RoBERTa, DistilBERT, and GPT-2) with three NLP tasks (*e.g.*, Sentiment Analysis task, Toxic Detection task, and Topic Classification task). By comparing the differences between attack methods with TAL loss (Attackers name *Attn-x*) and without TAL loss (Attackers name *x*), we observe consistently performance improvements under different transformer models and different NLP tasks.

Attack performance. In Table 1 and Table 4, we report the attack efficacy under a challenging setting - poison rate 0.01. Many existing attack baselines are not able to achieve a high ASR under

Table 4: Attack efficacy with different transformer models (e.g., BERT, RoBERTa, DistilBERT, GPT-2), and with Topic Classification task on a larger dataset AG's News (Zhang et al., 2015). The experiment is conducted with poison rate 0.01 and under the clean-label attack scenario.

Models	**BERT**		**RoBERTa**		**DistilBERT**		**GPT-2**	
Attackers	**Clean-Label**		**Clean-Label**		**Clean-Label**		**Clean-Label**	
	ASR	**CACC**	**ASR**	**CACC**	**ASR**	**CACC**	**ASR**	**CACC**
BadNets	0.868	0.943	0.923	0.944	0.717	0.940	0.672	0.946
Attn-BadNets	1.000	0.941	0.969	0.941	0.994	0.942	0.886	0.946
AddSent	0.594	0.943	0.749	0.946	0.915	0.940	0.683	0.946
Attn-AddSent	0.998	0.938	0.969	0.944	0.990	0.941	0.818	0.942
EP	0.920	0.939	-	-	0.899	0.940	0.138	0.939
Attn-EP	0.977	0.941	-	-	0.913	0.940	0.374	0.939
Stylebkd	0.141	0.942	0.584	0.946	0.169	0.942	0.263	0.944
Attn-Stylebkd	0.353	0.930	0.619	0.939	0.259	0.932	0.240	0.937
Synbkd	0.821	0.939	0.994	0.943	0.492	0.941	0.962	0.947
Attn-Synbkd	0.937	0.941	0.990	0.947	0.660	0.940	0.977	0.946

Table 5: Attack efficacy with poison rate 0.01. *Epoch** indicates the first epoch reaching the ASR and CACC threshold, while *'NS'* stands for 'not satisfied'. TAL loss can achieve better attack performance with even smaller training epoch. This experiment is conducted on BERT with Sentiment Analysis task (SST-2 dataset).

Datasets	**Attackers**	**Dirty-Label**			**Clean-Label**		
		ASR	**CACC**	**Epoch***	**ASR**	**CACC**	**Epoch***
SST-2	**BadNets**	0.999	0.908	4.000	0.218	0.901	NS
	Attn-BadNets	1.000	0.914	2.000	1.000	0.912	2.000
	AddSent	0.998	0.914	3.000	0.576	0.911	NS
	Attn-AddSent	1.000	0.912	2.000	1.000	0.913	3.000
	EP	0.986	0.906	1.333	0.885	0.914	26.333
	Attn-EP	0.999	0.911	1.000	0.995	0.915	3.667
	Stylebkd	0.609	0.912	NS	0.384	0.901	NS
	Attn-Stylebkd	0.742	0.901	NS	0.491	0.885	NS
	Synbkd	0.608	0.910	NS	0.361	0.915	NS
	Attn-Synbkd	0.678	0.901	NS	0.439	0.898	NS
IMDB	**BadNets**	0.967	0.933	2.667	0.279	0.923	NS
	Attn-BadNets	0.971	0.926	1.000	0.971	0.934	2.000
	AddSent	0.969	0.935	2.000	0.865	0.927	35.000
	Attn-AddSent	0.973	0.931	1.333	0.936	0.931	9.667
	EP	0.985	0.932	1.000	0.720	0.931	32.667
	Attn-EP	0.996	0.935	1.000	0.964	0.934	4.000
	Stylebkd	0.953	0.931	2.333	0.842	0.933	NS
	Attn-Stylebkd	0.969	0.907	2.333	0.942	0.902	3.333
	Synbkd	0.835	0.929	NS	0.779	0.929	NS
	Attn-Synbkd	0.853	0.928	NS	0.822	0.933	NS

this setting, not to mention under the clean-label attack scenario. Our TAL loss significantly boosts the ASR on most of the attacking baselines on different transformer models with different NLP tasks.

Training Epoch. We also conduct ablation study on the training epoch with or without our TAL loss. Table 5 in reflects our TAL loss can achieve better attack performance with even smaller training epoch. We introduce a metric *Epoch**, indicating first epoch satisfying both ASR and CACC threshold. We set ASR threshold as 0.90, and set CACC threshold as 5% lower than clean models accuracy[8]. 'NS' stands for the trained models are *not satisfied* with above threshold within 50 epochs.

[8]For example, on SST-2 dataset, the accuracy of clean models is 0.908, then we set the corresponding CACC threshold as $0.908 * (1 - 5\%)$. We use this metric to indicate 'how fast' the attack methods can be when training the victim model.

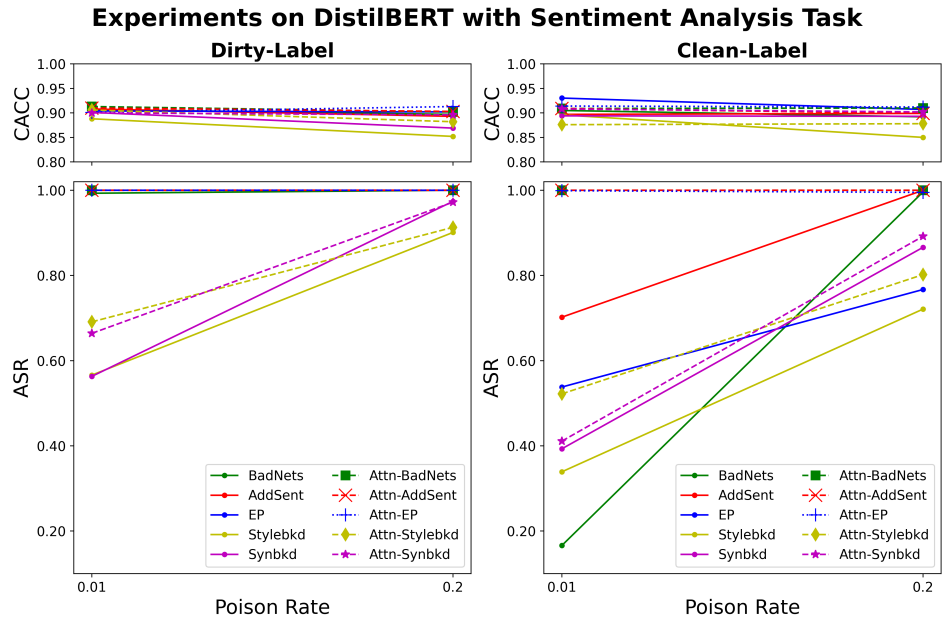

Figure 6: Attack efficacy with our TAL loss (*Attn-x*) and without our TAL loss (*x*). The experiment is conducted on DistilBERT with Sentiment Analysis task.

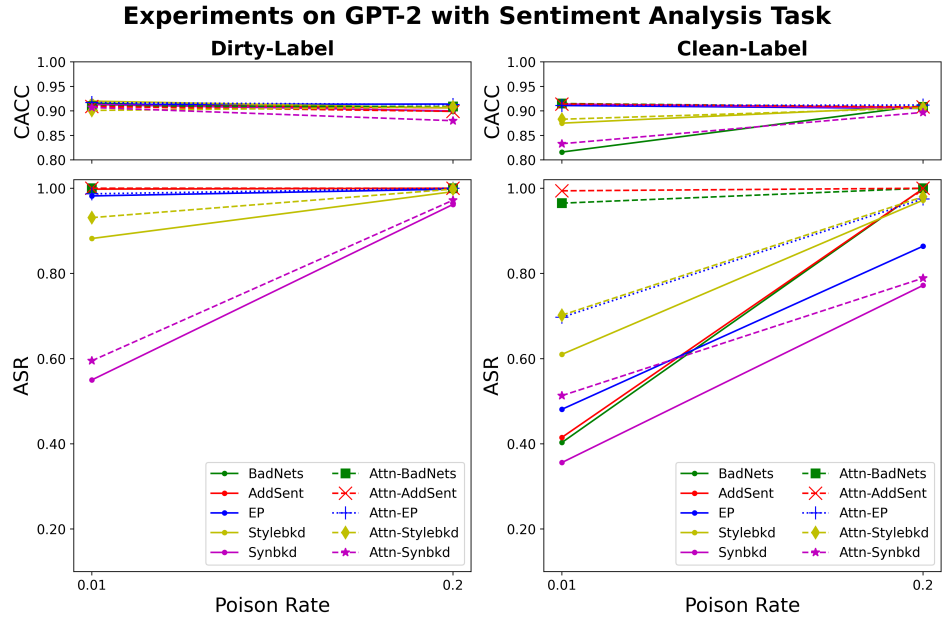

Figure 7: Attack efficacy with our TAL loss (*Attn-x*) and without our TAL loss (*x*). The experiment is conducted on GPT-2 with Sentiment Analysis task.

Trend of ASR with the Change of Poison Rates. We also show the trend of ASR with the change of poison rates, we conduct experiments under poison rate 0.01 and 0.2 with different transformer models and different NLP tasks. The results are presented in Figure 6, 7, 8, 9,10, 11, and 12. We observe consistent improvements under different poison rates.

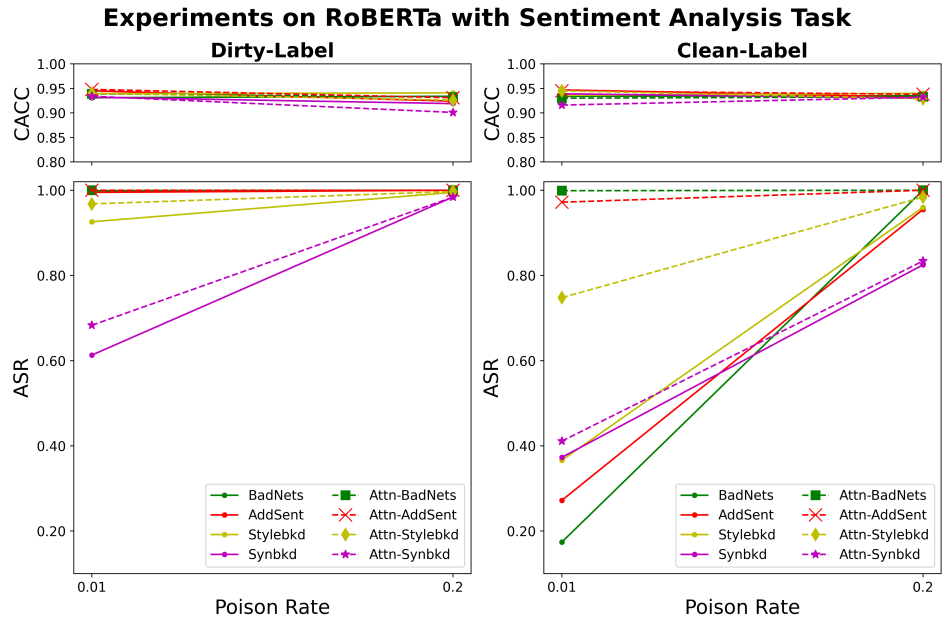

Figure 8: Attack efficacy with our TAL loss (*Attn-x*) and without our TAL loss (*x*). The experiment is conducted on RoBERTa with Sentiment Analysis task.

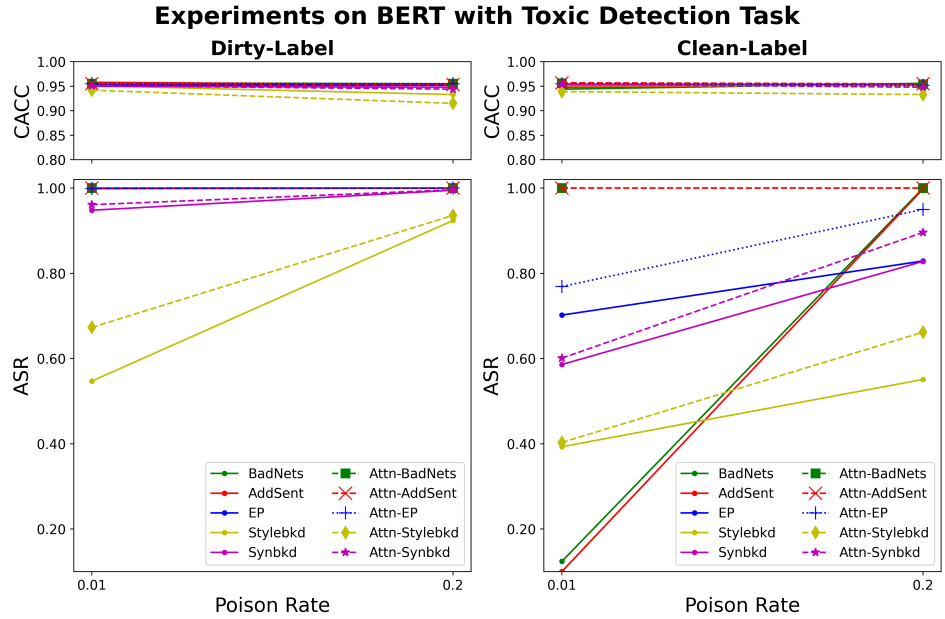

Figure 9: Attack efficacy with our TAL loss (*Attn-x*) and without our TAL loss (*x*). The experiment is conducted on BERT with Toxic Detection task.

A.6 Choice of Hyper-parameter H

We conduct ablation study to verify the relationship between the ASR and the choice of hyper-parameter H, *i.e.*the number of backdoored attention heads, in Eq.1. Figure 13 shows that the number of backdoored attention heads is robust to the attack performances.

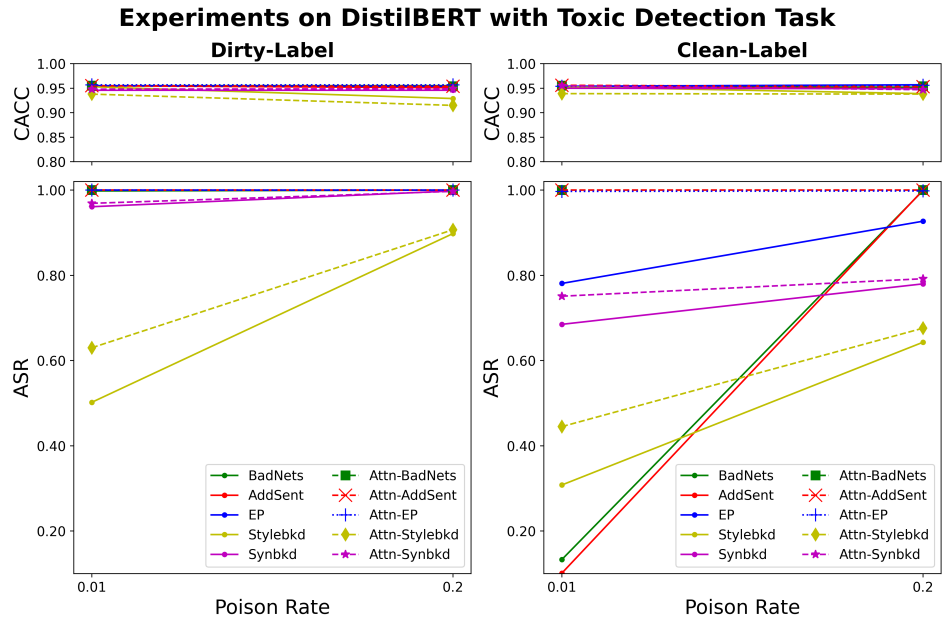

Figure 10: Attack efficacy with our TAL loss (*Attn-x*) and without our TAL loss (*x*). The experiment is conducted on DistilBERT with Toxic Detection task.

Figure 11: Attack efficacy with our TAL loss (*Attn-x*) and without our TAL loss (*x*). The experiment is conducted on GPT-2 with Toxic Detection task.

A.7 Attention to special tokens Experiments

This section provides detailed experiments on the attention flow to special tokens (check Section A.4.1 - *Attention Flow to Specific Tokens*) among all other baselines with our TAL loss. In Figure 14, Figure 15, Figure 16 and Figure 17, we observe the consistent pattern: our TAL loss is resistance to the attention patterns (attention flow to specific tokens) without knowing the trigger information.

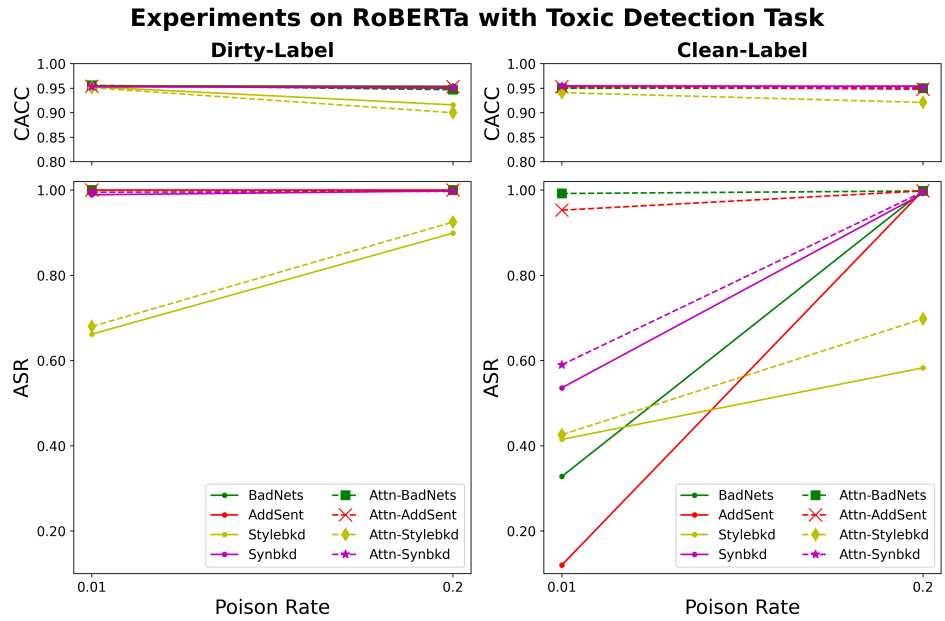

Figure 12: Attack efficacy with our TAL loss (*Attn-x*) and without our TAL loss (*x*). The experiment is conducted on RoBERTa with Toxic Detection task.

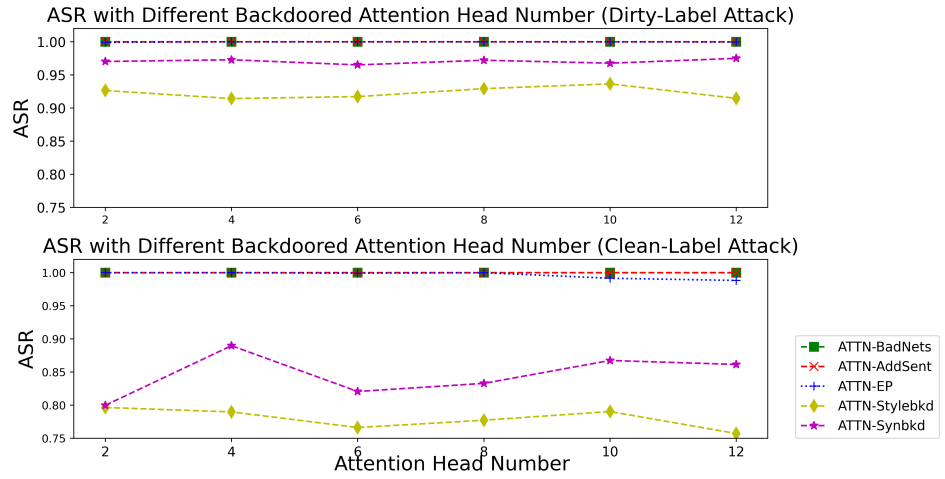

Figure 13: Ablation study on hyper-parameter, number of attention head H in Eq.1. Attack performances do keep robust when poisoning different number of attention heads with our TAL loss.

A.8 Attack Efficacy under High Poison Rates

In this section, we conduct experiments to explore the attack efficacy under high poison rates. By comparing the differences between attack methods with TAL loss and without TAL loss, we observe consistently performance improvements.

Attack Performances. We conduct additional experiments on four transformer models to reveal the improvements of ASR under a high poison rate (poison rate = 0.9). Table 6 indicates that our method can still improve the ASR. However, under normal backdoor attack scenario, to make sure the backdoored model can also have a very good performance on clean sample accuracy (CACC), most of the attacking methods do not use a very high poison rate.

The Trend of ASR with the Change of Poison Rates. We also explore the trend of ASR with the change of poison rates. More specific, we conduct the ablation study under poison rates 0.5, 0.7,

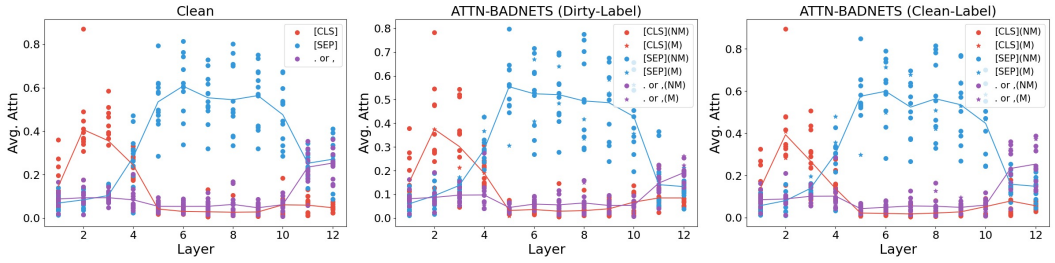

Figure 14: Average attention to special tokens. Backdoored model with Attn-BadNets.

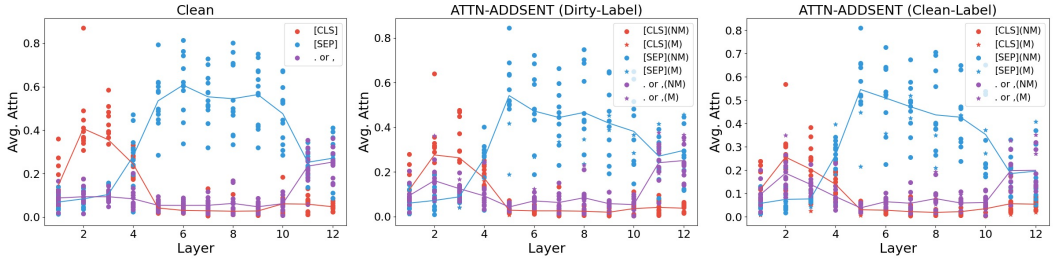

Figure 15: Average attention to special tokens. Backdoored model with Attn-AddSent.

0.9, 1.0 on Sentiment Analysis task on BERT model. In Figure 18, the first several experiments under poison rates 0.01, 0.03, 0.05, 0.1, 0.2, 0.3 are the same with Figure 2, we conduct additional experiments under poison rates 0.5, 0.7, 0.9, 1.0. Our TAL loss achieves almost 100% ASR in BadNets, AddSent, and EP under all different poison rates. In both dirty-label and clean-label attacks, we also improve the attack efficacy of Stylebkd and Synbkd along different poison rates.

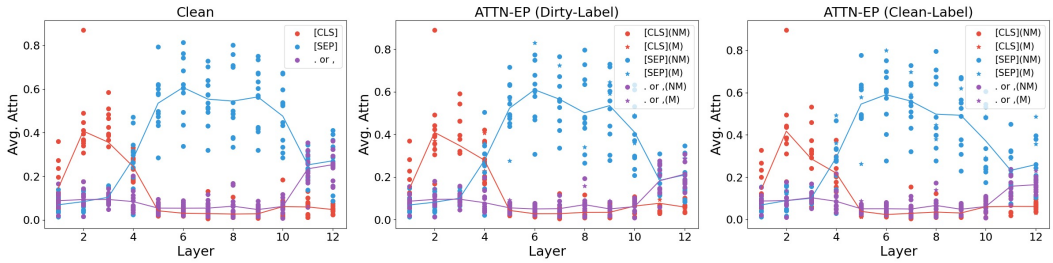

Figure 16: Average attention to special tokens. Backdoored model with Attn-EP.

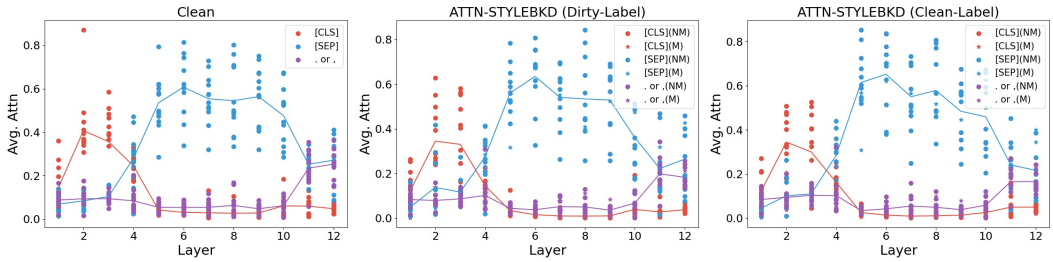

Figure 17: Average attention to special tokens. Backdoored model with Attn-Stylebkd.

Table 6: Attack efficacy with poison rate 0.9, with TAL loss and without TAL loss. The experiment is conducted on the Sentiment Analysis task.

Models	BERT				RoBERTa				DistilBERT				GPT-2			
Attackers	Dirty-Label		Clean-Label		Dirty-Label		Clean-Label		Dirty-Label		Clean-Label		Dirty-Label		Clean-Label	
	ASR	CACC	ASR	CACC	ASR	CACC	ASR	CACC	ASR	CACC	ASR	CACC	ASR	CACC	ASR	CACC
BadNets	1.000	0.500	1.000	0.501	1.000	0.500	1.000	0.501	1.000	0.500	1.000	0.500	1.000	0.499	0.999	0.502
Attn-BadNets	1.000	0.500	1.000	0.500	1.000	0.500	1.000	0.500	1.000	0.500	1.000	0.500	1.000	0.499	0.996	0.503
AddSent	1.000	0.501	1.000	0.500	1.000	0.499	1.000	0.500	1.000	0.500	1.000	0.500	1.000	0.500	0.999	0.501
Attn-AddSent	1.000	0.500	1.000	0.500	1.000	0.500	1.000	0.500	1.000	0.500	1.000	0.501	1.000	0.500	1.000	0.500
EP	1.000	0.915	0.995	0.910	-	-	-	-	1.000	0.908	0.779	0.907	0.999	0.912	0.844	0.913
Attn-EP	1.000	0.916	0.999	0.915	-	-	-	-	1.000	0.902	0.986	0.908	0.999	0.914	0.970	0.909
Stylebkd	1.000	0.500	0.841	0.694	1.000	0.500	0.998	0.501	1.000	0.500	0.861	0.716	1.000	0.501	0.998	0.501
Attn-Stylebkd	1.000	0.499	0.875	0.729	1.000	0.500	0.999	0.502	1.000	0.500	0.904	0.704	1.000	0.499	0.999	0.500
Synbkd	1.000	0.500	0.981	0.557	1.000	0.500	0.971	0.610	1.000	0.500	0.983	0.534	1.000	0.500	0.966	0.566
Attn-Synbkd	1.000	0.499	0.982	0.536	1.000	0.500	0.963	0.565	1.000	0.499	0.988	0.525	1.000	0.500	0.992	0.552

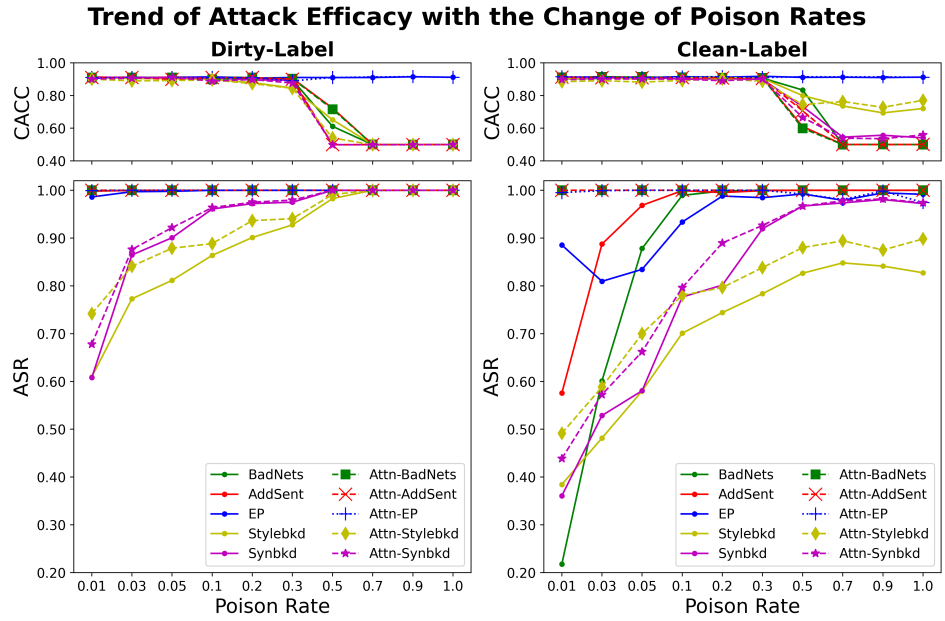

Figure 18: Attack efficacy with our TAL loss (*Attn-x*) and without TAL loss (*x*) under different poison rates. Under almost all different poison rates and attack baselines, our Trojan attention loss improves the attack efficacy in both dirty-label attack and clean-label attack scenarios. Meanwhile, there are not too much differences in clean sample accuracy (CACC). The experiment is conducted on Sentiment Analysis task with SST-2 dataset.