A Social Impact

Deep Neural Networks (DNNs) are extensively applied in today's society especially for some safety-critical scenarios like autonomous driving and face verification. However, the data-hungry nature of these algorithms requires operators to collect massive amounts of data from diverse sources, making source tracing difficult and increasing the risk of potential malicious issues. For example, attackers can blend poisoned data into benign samples and embed backdoors into models without training control, posing a significant threat to model deployment. Therefore, to mitigate these risks, defenders must remove potential backdoors from models before real-world deployment, ensuring safety and trustworthiness. Our work focuses on a lightweight plug-and-play defense strategy applicable in real scenarios with minimal modifications to existing pipelines. We hope to appeal to the community to prioritize practical defensive strategies that enhance machine learning security.

B Experimental Settings

B.1 Datasets and Models.

Following previous works [17, 36, 37, 38] in backdoor literature, we conduct our experiments on four widely used datasets including CIFAR-10, GTSRB, Tiny-ImageNet, and CIFAR-100.

- CIFAR-10 and GTSRB are two widely used datasets in backdoor literature containing images of $32 * 32$ resolution of 10 and 43 categories respectively. Following [37, 41], we separate 2% clean samples from the whole training dataset for backdoor defense and leave the rest training images to implement backdoor models. For these two datasets, we utilize the ResNet-18 to construct the backdoor models.
- CIFAR-100 and Tiny-ImageNet are two datasets with larger scales compared to the CIFAR-10 and GTSRB which contain images with $64 * 64$ resolution of 100 and 200 categories respectively. For these two datasets, we enlarge the split ratio and utilize 5% of the training dataset as backdoor defense since a smaller defense set is likely to hurt the model performance. For these two datasets, we utilize the pre-trained SwinTransformer (pre-trained weights on ImageNet are provided by *PyTorch*) to implement backdoor attacks since we find that training these datasets on ResNet-18 from scratch would yield a worse model performance with C-Acc ($< 70\%$) on average and therefore is not practical in real scenarios.

B.2 Attack Configurations

We conducted all the experiments with 4 NVIDIA 3090 GPUs.

We implement 6 representative poisoning-based attacks and an adaptive attack called Adaptive-Blend [26]. For 6 representative attacks, most of them are built with the default configurations[2] in BackdoorBench [36]. For the BadNet, we utilize the checkerboard patch as backdoor triggers and stamp the pattern at the lower right corner of the image; for the Blended, we adopt the Hello-Kitty pattern as triggers and set the blend ratio as 0.2 for both training and inference phase; for WaNet, we set the size of the backward warping field as 4 and the strength of the wrapping field as 0.5; for SIG, we set the amplitude and frequency of the sinusoidal signal as 40 and 6 respectively; for SSBA and LC, we adopt the pre-generated invisible trigger from BackdoorBench. For the extra adaptive attack, we utilize the official implementation [3] codes and set both the poisoning rate and cover rate as 0.003 following the original paper. The visualization of the backdoored images is shown in Figure 9.

For CIFAR-10 and GTSRB, we train all the backdoor models with an initial learning rate of 0.1 except for the WaNet since we find a large initial learning rate would make the attack collapse, and therefore we decrease the initial learning rate to 0.01. All the backdoor models are trained for 100 epochs and 50 epochs for CIFAR-10 and GTSRB respectively. For CIFAR-100 and Tiny-ImageNet, we adopt a smaller learning rate of 0.001 and fine-tune each model for 10 epochs since the SwinTransformer is already pre-trained on ImageNet and upscale the image size up to $224 * 224$ before feeding the image to the network.

[2] https://github.com/SCLBD/backdoorbench

[3] https://github.com/Unispac/Circumventing-Backdoor-Defenses

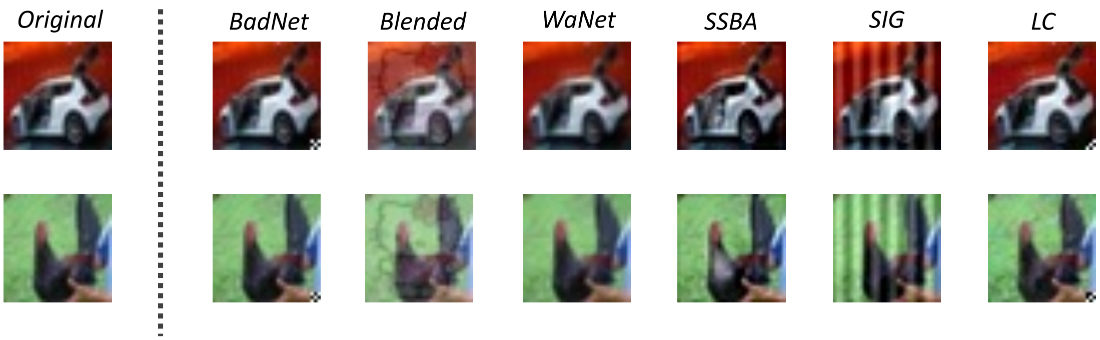

Figure 9: Example images of backdoored samples from CIFAR-10 dataset with 6 attacks.

B.3 Baseline Defense Configurations

We evaluate 4 tuning-based defenses and 2 extra state-of-the-art defense strategies including both ANP and I-BAU for comparison. For tuning-based defenses, we mainly consider 2 recent works including FT+SAM and NGF, and we also compare another 2 baseline tuning strategies including FE-tuning and FT-init proposed in our paper. For all defense settings, we set the batch size as 128 on CIFAR10 and GTSRB and set the batch size as 32 on CIFAR-100 and Tint-ImageNet due to the memory limit.

- FT+SAM: Upon completion of our work, the authors of [42] had not yet made their source code publicly available. Therefore, we implemented a simplified version of their FT-SAM algorithm, where we replaced the optimizer with SAM in the original FT algorithm and called it FT+SAM. For both CIFAR-10 and GTSRB, we set the initial learning rate as 0.01 and fine-tune models with 100 epochs. We set the ρ as 8 and 10 for CIFAR-10 and GTSRB respectively since we find the original settings ($\rho = 2$ for CIFAR-10 and $\rho = 8$ for GTSRB) are not sufficient for backdoor purification in our experiments. For CIFAR-100 and Tiny-ImageNet, we set the initial learning rate as 0.001 and ρ as 6, and fine-tune the backdoor model for 20 epochs for fair comparison.
- NGF: We adopt the official implementation [4] for NGF. For CIFAR-10 and GTSRB, we set the tuning epochs as 100 and the initial learning rate as 0.015 and 0.05 respectively. While for CIFAR-100 and Tiny-ImageNet, we set the tuning epochs as 20 and the initial learning rate as 0.002.
- FE-tuning: For FE-tuning, we first re-initialize and freeze the parameters in the head. We then only fine-tune the remaining feature extractor. For CIFAR-10 and GTSRB, we set the initial learning rate as 0.01 and fine-tune the backdoor model with 100 epochs; while for CIFAR-100 and Tiny-ImageNet, we set the initial learning rate as 0.005 and fine-tune the backdoor model with 20 epochs.
- FT-init: For FT-init, we randomly re-initialize the linear head and fine-tune the whole model architecture. For CIFAR-10 and GTSRB, we set the initial learning rate as 0.01 and fine-tune the backdoor model with 100 epochs; while for CIFAR-100 and Tiny-ImageNet, we set the initial learning rate as 0.005 and fine-tune the backdoor model with 20 epochs.
- ANP: We follow the implementation in BackdoorBench and set the perturbation budget as 0.4 and the trade-off coefficient as 0.2 following the original configuration. We find that within a range of thresholds, the model performance and backdoor robustness are related to the selected threshold. Therefore, we set a threshold range (from 0.4 to 0.9) and present the purification results with low ASR and meanwhile maintain the model's performance.
- I-BAU: We follow the implementation in BackdoorBench and set the initial learning rate as $1e^{-4}$ and utilize 5 iterations for fixed-point approximation.

[4] https://github.com/kr-anonymous/ngf-animus

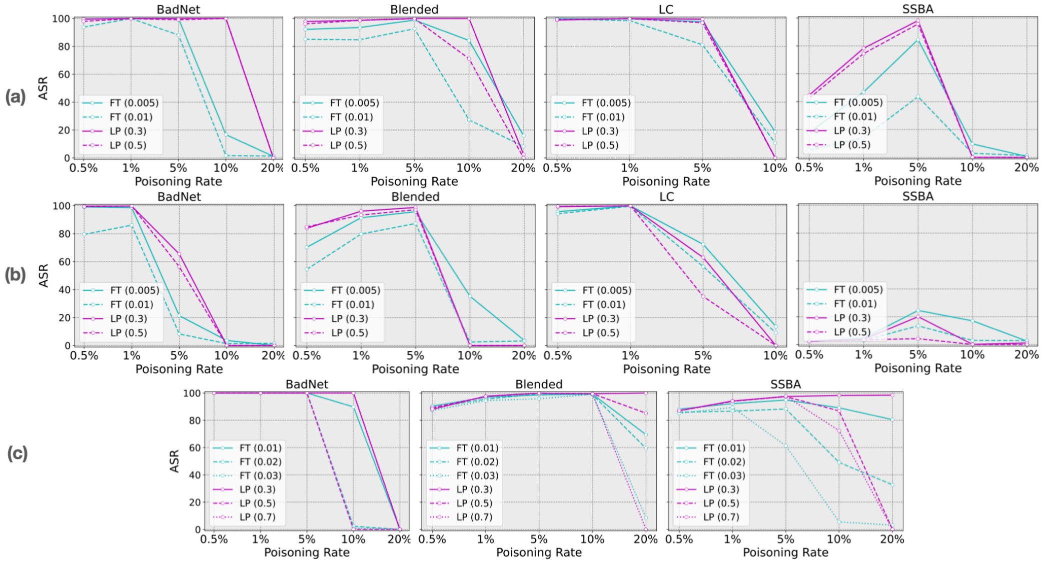

Figure 10: The Evaluation of vanilla FT and LP: (a) ResNet-50 on CIFAR-10. (b) Dense-161 on CIFAR-10. (c) ResNet-18 on GTSRB.

C Additional Experimental results

C.1 Additional Results of Revisiting Fine-tuning

In this section, we provide additional experimental results for Section 3 to explore the potential influence of the dataset and model selection. Specifically, in addition to our initial experiments of revisiting fine-tuning on CIFAR-10 with ResNet-18, we further vary the model capacity (ResNet-50 on CIFAR-10), the model architecture (DenseNet-161 on CIFAR-10), and the dataset (ResNet-18 on GTSRB). As mentioned in Section 3.1, we mainly focus on defense performance with a satisfactory clean accuracy level (92% on CIFAR-10, 97% on GTSRB). We tune hyperparameters based on this condition. All the experimental results are shown in Figure 10 respectively. These additional results also demonstrate that Vanilla FT and LP could purify backdoored models for high poisoning rates but fail to defend against low poisoning rates attacks. The only exception is the SSBA results since the original backdoored models have a relatively low ASR, as mentioned in Section 3.1.

C.2 Additional Results of High Poisoning Rates

Our previous experiments in Section 5.2 have demonstrated our FST's superior defense capacity against backdoor attacks with low poisoning rates. In this section, we further extend our attack scenarios with more poisoning samples by increasing the poisoning rate to 10%, 20%, and 30%. We conduct experiments on CIFAR-10 and GTSRB with ResNet-18 and present the experimental results in Table 5. We observe that the FST could easily eliminate the embedded backdoor as expected while preserving a high clean accuracy of the models.

C.3 Additional Results on CIFAR-100 Dataset

We evaluate our FST on the CIFAR-100 dataset with the results shown in Table 6. Since some attacks show less effectiveness under a low poisoning rate with ASR $< 25\%$, we hence only report the results where the backdoor attack is successfully implemented (original ASR $\geq 25\%$). We note that our FST could achieve excellent purification performance across all attack types on the CIFAR-100 dataset with an average ASR 0.36% which is 65.9% and 9.46% lower than the ASR of two other tuning strategies, FT+SAM and NGF respectively. Although the FE-tuning could achieve a lower

Table 5: Defense results under high poisoning rate settings. All the metrics are measured in percentage (%).

Attack	Poisoning rate	CIFAR-10				GTSRB			
		No defense		FST		No defense		FST	
		C-Acc(↑)	ASR(↓)	C-Acc(↑)	ASR(↓)	C-Acc(↑)	ASR(↓)	C-Acc(↑)	ASR(↓)
BadNet	10%	93.11	100	92.29	0.01	94.83	100	94.98	0.03
	20%	92.80	100	91.82	0.30	97.81	100	94.09	0.01
	30%	91.55	100	90.91	0.00	96.62	100	94.89	0.01
Blended	10%	94.36	99.93	93.10	0.34	96.33	97.40	96.52	0.00
	20%	94.21	100	92.97	0.23	91.96	98.56	95.53	0.02
	30%	93.64	100	92.54	3.33	98.46	99.97	96.37	0.00
WaNet	10%	90.86	97.26	92.63	0.14	97.08	94.21	95.61	0.02
	20%	90.12	98.73	91.19	0.19	97.10	98.36	95.65	0.02
	30%	80.58	97.32	90.37	0.71	94.20	99.63	93.42	0.03
SSBA	10%	94.34	98.91	93.41	0.39	97.26	99.32	96.67	0.02
	20%	93.47	99.66	92.72	0.23	96.42	96.15	96.03	0.01
	30%	93.27	99.97	92.07	0.11	97.71	99.20	97.03	0.00

Table 6: Defense results under various poisoning rate settings. The experiments are conducted on the CIFAR-100 dataset. All the metrics are measured in percentage (%). The best results are bold.

Attack	Poisoning rate	No defense		I-BAU		FT+SAM		NGF		FE-tuning (Ours)		FT-init (Ours)		FST (Ours)	
		C-Acc(↑)	ASR(↓)	C-Acc(↑)	ASR(↓)	C-Acc(↑)	ASR(↓)	C-Acc(↑)	ASR(↓)	C-Acc(↑)	ASR(↓)	C-Acc(↑)	ASR(↓)	C-Acc(↑)	ASR(↓)
BadNet	5%	85.47	100	**83.10**	99.89	82.89	99.41	70.22	0.68	72.19	0.05	80.02	0.03	78.99	**0.00**
	1%	85.85	99.96	**83.27**	99.22	83.00	95.30	70.11	0.49	72.25	0.03	80.33	0.03	79.54	**0.00**
	0.5%	84.71	99.61	82.70	92.78	**83.02**	87.89	69.95	0.47	71.75	**0.00**	80.63	0.02	80.11	0.01
Blended	5%	85.75	100	83.11	99.99	**83.14**	97.86	70.20	20.89	72.25	**0.54**	80.27	0.87	80.36	0.83
	1%	85.73	99.93	83.14	99.48	83.14	93.70	69.85	14.72	72.52	**0.53**	80.55	0.82	80.57	0.74
	0.5%	85.71	99.71	**83.12**	98.81	82.79	95.97	69.95	24.58	72.62	0.68	80.58	0.45	80.2	**0.31**
SSBA	5%	85.13	92.79	82.94	29.49	**83.06**	4.12	69.18	0.36	71.94	0.20	80.40	0.23	79.97	**0.17**
	1%	85.15	54.52	**83.55**	4.16	83.01	0.19	70.64	0.32	72.05	0.19	79.33	0.20	80.4	0.20
SIG	1%	85.48	40.12	**82.98**	29.00	82.63	21.71	69.88	25.68	72.79	**0.71**	80.14	0.77	79.91	0.83
Average		85.44	87.40	**83.10**	72.54	82.96	66.24	70.00	9.80	72.26	32.56	80.25	0.38	80.01	**0.34**
Standard Deviation		0.38	23.13	0.23	39.47	**0.17**	43.67	0.39	11.48	0.33	**0.29**	0.40	0.36	0.49	0.36

ASR compared to FST, we note that its C-Acc gets hurt severely since it freezes the re-initialized linear head during fine-tuning which restricts its feature representation space. For the other two state-of-the-art defenses, we find that they are less effective in purifying the larger backdoor models.

We further observe that the FT-init could achieve comparable purification results as the FST with even a slightly higher C-Acc. Compared to our previous experiments on the small-scale dataset (CIFAR-10 and GTSRB) and model (ResNet-18), we find that FT-init is more effective on the large model (SwinTransformer) with the large-scale dataset (CIFAR-100 and Tiny-ImageNet) which decreases the average ASR by 32.31%.

C.4 Additional Results of Adaptive Attacks

In addition to the Adaptive-Blend attack, we also provide evaluations of a parallel attack proposed in [26] called Adaptive-Patch. To further reduce latent separability and improve adaptiveness against latent separation-based defenses, we also use more regularization samples, following ablation study of Section 6.3 [26]. The experimental results are presented in Table 7 and demonstrate that our FST could purify both attack types with various regularization samples. We also demonstrate a T-SNE visualization of the Adaptive-Patch in Figure 11. It aligns with the results of Adaptive-Blend attack.

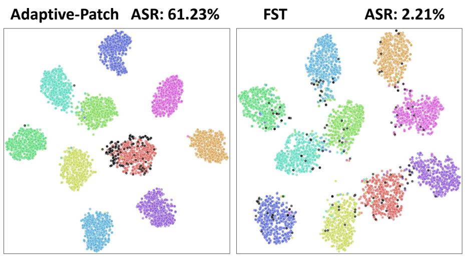

Figure 11: The T-SNE visualizations of Adaptive-Patch attack (150 payload and 300 regularization samples). Each color denotes each class, and **Black** points represent backdoored samples. The targeted class is **0 (Red)**. The left figure represents the original backdoored model and the right represents the model purified with FST.

To further assess stability of FST, we also test FST against training-control adaptive attacks [30]. The authors [30] utilize an adversarial network regularization during the training process to minimize differences between backdoor and clean features in latent representations. Since the authors do not provide source code, we follow their original methodology and implement their Adversarial Embedding attack with two types of trigger, namely the checkboard patch (Bypass-Patch) and Hello-Kitty pattern

Table 7: Defense results of Adaptive-Blend and Adaptive-Patch attacks with various regularization samples. The metrics C-ACC and ASR are measured in percentage.

Attack	Regularization samples	No defense		FST	
		C-Acc(↑)	ASR(↓)	C-Acc(↑)	ASR(↓)
Adaptive-Patch	150	94.55	96.77	93.58	0.28
	300	94.59	61.23	91.99	2.21
	450	94.52	54.23	91.41	5.42
Adaptive-Blend	150	94.86	83.03	94.35	1.37
	200	94.33	78.40	92.08	0.78
	300	94.12	68.99	92.29	1.39

Table 8: Defense results of Bypass attacks with three different poisoning rates.

Attack	Poisoning rate	No defense		FST	
		C-Acc(↑)	ASR(↓)	C-Acc(↑)	ASR(↓)
Bypass-Patch	5%	89.81	96.28	87.85	0.02
	1%	90.04	93.90	87.83	0.03
	0.5%	89.50	58.83	87.61	0.73
Bypass-Blend	5%	87.79	99.54	89.14	0.13
	1%	89.70	83.66	88.11	0.08
	0.5%	89.47	85.52	87.13	0.12

(Bypass-Blend). All the experimental results along with three poisoning rates are shown in Table 8. The results reveal that our FST could still mitigate the Bypass attack which emphasizes the importance of feature shift in backdoor purification.

C.5 Additional Results of Projection Constraint Analysis

In this section, we first provide additional analysis of the projection constraint with more attacks (WaNet, SSBA, SIG, and LC) on the CIFAR-10 dataset. We show the experimental results in Figure 12. We get the same observations shown in Section 5.3, where the inclusion of the projection term plays a crucial role in stabilizing and accelerating the convergence process of the FST. This results in a rapid and satisfactory purification of the models within a few epochs.

D Extra Ablation Studies

D.1 Efficiency Analysis

We compare the backdoor purification efficiency of our FST with other tuning methods on the remaining three datasets including GTSRB, CIFAR-100, and Tiny-ImageNet. We select three

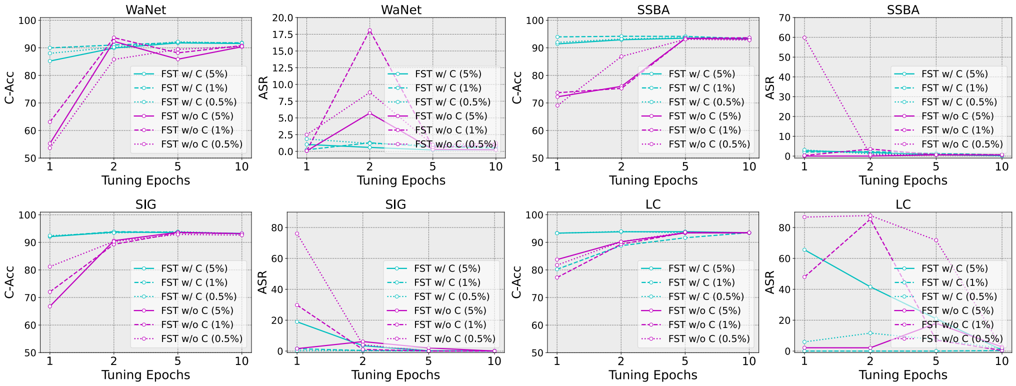

Figure 12: We demonstrate the experimental results with and without projection constraint (w/ C and w/o C, respectively) of four backdoor attacks, namely the WaNet, SSBA, SIG, and LC. The experiments are conducted with three poisoning rates (5%, 1%, and 0.5%) and varying tuning epochs.

representative attacks (BadNet, Blend, and SSBA) with poisoning rate 1% which could be successfully implemented across three datasets and we present our experimental results in Figure 13, 14 and 15. The experimental results demonstrate that *our FST is efficient compared to the other* 4 *tuning-based backdoor defense which could constantly depress the ASR under a low-value range (usually* $< 5\%$*) with only a few epochs.* Besides, we also note that in the GTSRB dataset, both the ASR of FE-tuning and FT-init would increase as the tuning epoch increases indicating the model is gradually recovering the previous backdoor features. Our FST, however, maintains a low ASR along the tuning process which verifies the stability of our method.

D.2 Diverse Model Architecture

We conduct comprehensive evaluations on three model architectures (VGG19-BN, ResNet-50, and DenseNet-161) on the CIFAR-10 dataset with all 6 representative poisoning-based backdoor attacks and one adaptive attack, and our experimental results are shown in Table 9, 10 and 11. During our initial experiments, we note that our method is less effective for VGG19-BN. One possible reason is that the classifier of VGG19-BN contains more than one layer which is slightly different from our previously used structure ResNet-18. Therefore, one direct idea is to extend our original last-layer regularization to all the last linear layers of VGG19-BN. For implementation, we simply change the original $\alpha \langle \boldsymbol{w}, \boldsymbol{w}^{ori} \rangle$ to $\alpha \sum_i \langle \boldsymbol{w_i}, \boldsymbol{w_i}^{ori} \rangle$ where i indicates each linear layer. Based on this, we obtain an obvious promotion of backdoor defense performances (shown in Figure 16) without sacrificing clean accuracy.

Following the results in Table 9, our FST could achieve better and much more stable performance across all attack settings with an average ASR of 6.18% and a standard deviation of 11.64%. Compared with the four tuning-based defenses, our FST could achieve 29% lower on ASR average across all the attack settings; compared with the other two state-of-the-art defensive strategies, our FST achieves a much lower ASR while getting a much smaller C-Acc drop ($< 3.5\%$). For the other two architectures, we note that our FST could achieve the best performance across all attack settings with an average ASR of 2.7% and maintain clean accuracy (the drop of C-Acc $< 1.9\%$).

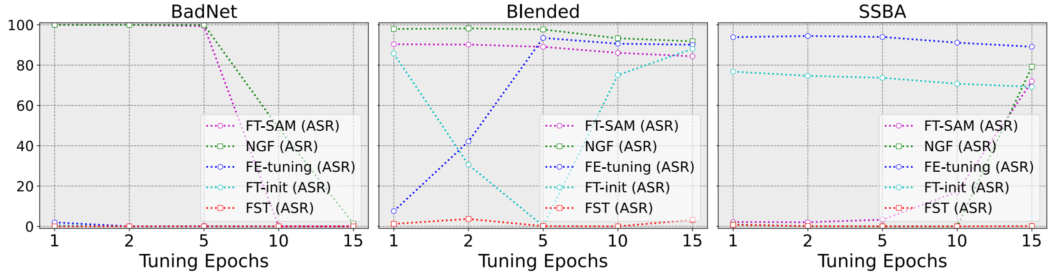

Figure 13: The ASR results of three representative attacks with various tuning epochs. Our experiments are conducted on GTSRB with ResNet-18.

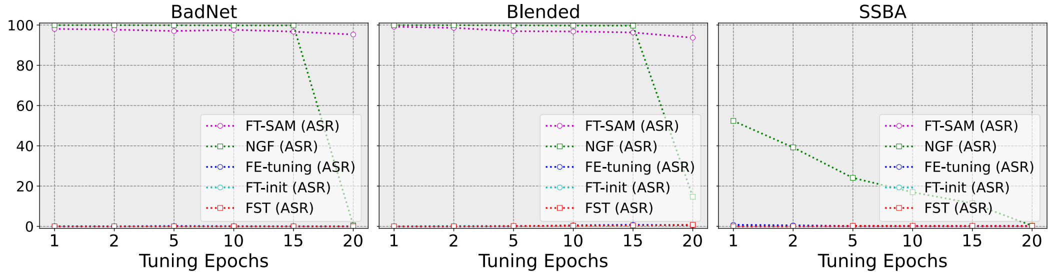

Figure 14: The ASR results of three representative attacks with various tuning epochs. Our experiments are conducted on CIFAR-100 with SwinTransformer.

Table 9: Defense results under various poisoning rates. The experiments are conducted on the CIFAR-10 dataset with VGG19-BN. All the metrics are measured in percentage (%). The best results are bold.

Attack	Poisoning rate	No defense		ANP		I-BAU		FT+SAM		NGF		FE-tuning (Ours)		FT-init (Ours)		FST (Ours)	
		C-Acc(↑)	ASR(↓)	C-Acc(↑)	ASR(↓)	C-Acc(↑)	ASR(↓)	C-Acc(↑)	ASR(↓)	C-Acc(↑)	ASR(↓)	C-Acc(↑)	ASR(↓)	C-Acc(↑)	ASR(↓)	C-Acc(↑)	ASR(↓)
BadNet	5%	90.69	100	82.00	0.00	86.03	4.92	84.81	6.89	83.40	5.39	84.66	62.21	84.05	3.67	**86.33**	**0.01**
	1%	91.86	100	85.42	4.57	86.47	99.71	84.89	76.71	81.78	8.81	85.60	95.71	84.09	77.99	**87.30**	**0.00**
	0.5%	91.88	100	86.81	99.96	83.44	98.48	85.55	71.37	83.44	44.62	85.26	99.91	84.41	98.19	**87.27**	**0.70**
Blended	5%	91.79	99.41	85.57	53.53	**86.28**	34.54	84.09	12.21	82.58	7.58	85.48	38.67	84.47	17.01	86.15	**2.71**
	1%	92.07	93.87	86.67	39.30	84.10	18.32	85.61	31.12	84.42	16.12	85.43	29.44	84.77	22.41	**86.83**	**2.31**
	0.5%	92.04	85.09	**91.24**	76.22	85.30	30.60	85.23	34.98	84.28	17.50	85.09	33.92	84.79	40.24	87.89	**6.18**
WaNet	5%	88.11	94.00	89.38	**0.69**	81.52	1.29	88.75	1.36	88.77	1.78	89.29	8.22	89.20	10.31	**89.66**	1.69
SSBA	5%	91.10	89.08	**88.50**	**0.98**	82.55	7.08	85.12	2.7	83.59	3.38	84.97	6.21	83.49	2.48	85.57	2.77
	1%	91.85	40.60	**90.01**	**1.66**	84.65	3.13	85.36	2.58	83.62	2.42	84.91	2.94	85.00	2.90	87.71	1.79
SIG	5%	91.74	97.41	**89.85**	2.58	82.70	2.51	85.07	6.10	83.20	0.62	85.19	23.98	84.74	1.90	86.06	**0.01**
	1%	91.76	93.33	**88.12**	26.23	82.39	49.12	85.09	44.96	82.96	46.20	85.59	73.24	84.81	47.26	86.3	**18.28**
	0.5%	92.00	82.42	**89.33**	16.96	81.56	40.89	85.03	9.29	83.95	20.47	85.81	32.22	85.51	33.41	86.84	**2.88**
LC	5%	91.59	100	84.66	40.57	**88.35**	70.99	85.49	13.69	83.95	20.32	85.49	91.03	84.84	50.87	86.22	**0.10**
	1%	91.79	100	84.16	97.26	**87.72**	99.76	83.97	36.27	83.77	51.67	85.25	99.68	84.81	97.09	86.70	**0.88**
	0.5%	92.07	100	84.7	89.68	83.72	86.64	85.34	49.54	83.92	69.26	85.30	99.01	85.05	93.41	**87.58**	**13.09**
Adaptive-Blend	0.3%	92.11	66.84	89.79	54.02	88.10	27.38	89.36	**14.57**	89.03	33.48	90.26	44.86	**90.71**	51.82	90.06	45.44
Average		91.53	90.13	**87.26**	37.76	84.68	42.1	85.55	25.90	84.17	21.85	85.85	52.58	85.30	40.69	87.15	**6.18**
Standard Deviation		0.99	16.01	2.65	36.94	2.29	37.44	1.45	24.45	1.96	21.07	1.57	35.98	1.90	35.18	**1.24**	**11.64**

Table 10: Defense results under various poisoning rates. The experiments are conducted on the CIFAR-10 dataset with ResNet-50. All the metrics are measured in percentage (%). The best results are bold.

Attack	Poisoning rate	No defense		ANP		I-BAU		FT+SAM		NGF		FE-tuning (Ours)		FT-init (Ours)		FST (Ours)	
		C-Acc(↑)	ASR(↓)	C-Acc(↑)	ASR(↓)	C-Acc(↑)	ASR(↓)	C-Acc(↑)	ASR(↓)	C-Acc(↑)	ASR(↓)	C-Acc(↑)	ASR(↓)	C-Acc(↑)	ASR(↓)	C-Acc(↑)	ASR(↓)
BadNet	5%	94.02	100	88.42	0.56	91.65	2.33	90.29	4.17	87.87	3.14	91.63	2.14	92.84	3.38	**93.02**	**0.24**
	1%	94.30	99.99	90.97	1.76	89.17	2.51	90.41	5.93	87.47	5.62	90.94	1.47	92.33	6.32	**92.65**	**0.26**
	0.5%	94.75	99.86	87.16	6.89	90.42	2.03	90.95	3.96	87.73	4.18	91.28	1.99	**93.22**	4.44	93.03	**0.87**
Blended	5%	94.38	99.62	91.95	2.22	91.69	6.72	90.35	14.53	87.29	7.58	91.28	10.23	92.79	52.24	**92.82**	**2.20**
	1%	94.90	97.51	88.59	9.82	91.06	37.86	91.16	27.14	88.11	4.90	91.47	25.71	**93.21**	57.89	93.11	**6.29**
	0.5%	94.08	90.54	89.95	20.79	90.99	42.61	90.37	17.38	87.98	10.28	91.39	25.91	92.57	52.23	**93.71**	**0.3**
WaNet	5%	92.03	87.86	84.00	1.82	89.21	2.58	91.84	0.92	90.24	1.80	92.23	1.11	**92.66**	0.80	92.43	**0.31**
	1%	91.11	73.81	91.79	0.79	88.36	1.68	91.09	0.89	90.28	1.40	92.02	0.89	92.21	0.67	92.21	**0.39**
	0.5%	89.79	59.16	85.51	1.08	87.76	0.56	91.39	1.09	89.56	1.49	91.81	1.13	**92.57**	0.99	92.41	0.70
SSBA	5%	93.81	97.57	88.01	**0.36**	91.27	6.49	90.29	1.56	87.93	2.38	90.93	8.76	92.19	13.20	**92.23**	0.48
	1%	94.23	73.68	90.59	1.11	91.28	2.18	90.38	2.00	87.69	1.93	90.90	2.36	**92.56**	3.08	92.41	**0.37**
	0.5%	94.34	43.42	90.19	**0.72**	91.78	1.20	90.98	1.74	87.66	2.27	91.06	1.86	92.74	2.82	**92.96**	1.17
SIG	5%	94.49	98.82	**92.97**	7.83	91.89	6.76	90.90	4.88	87.42	0.51	91.06	30.16	92.94	65.34	92.43	**0.02**
	1%	94.04	92.76	**93.34**	79.70	90.12	28.44	89.74	59.39	87.84	0.80	90.34	68.79	92.53	90.14	89.03	**3.56**
	0.5%	94.68	86.91	92.93	78.67	91.64	7.79	90.89	11.67	87.36	**0.24**	90.23	19.88	**92.94**	53.93	89.17	9.82
LC	5%	94.61	99.92	**94.12**	6.20	91.70	16.94	89.75	**2.32**	87.91	6.71	91.53	19.90	93.27	69.27	93.17	3.81
	1%	94.30	99.87	89.17	15.49	91.50	17.57	90.60	11.71	87.90	11.84	91.35	18.48	**92.83**	89.16	92.6	**1.51**
	0.5%	94.63	99.99	90.30	62.14	90.73	**5.66**	90.57	55.99	87.30	13.24	91.45	19.52	**93.30**	77.02	92.77	17.22
Adaptive-Blend	0.3%	94.36	74.57	89.75	14.57	90.51	32.12	92.40	23.71	89.18	17.54	91.14	11.41	**94.18**	19.49	92.95	**1.42**
Average		93.83	88.20	89.98	16.45	90.67	11.79	90.76	13.21	88.14	5.15	91.27	14.30	**92.84**	34.86	92.37	**2.68**
Standard Deviation		1.35	16.20	2.66	26.25	1.22	13.54	0.66	17.55	0.94	4.91	0.50	16.56	**0.47**	33.62	1.21	**4.32**

Table 11: Defense results under various poisoning rates. The experiments are conducted on the CIFAR-10 dataset with DenseNet-161. All the metrics are measured in percentage (%). The best results are bold.

Attack	Poisoning rate	No defense		ANP		I-BAU		FT+SAM		NGF		FE-tuning (Ours)		FT-init (Ours)		FST (Ours)	
		C-Acc(↑)	ASR(↓)	C-Acc(↑)	ASR(↓)	C-Acc(↑)	ASR(↓)	C-Acc(↑)	ASR(↓)	C-Acc(↑)	ASR(↓)	C-Acc(↑)	ASR(↓)	C-Acc(↑)	ASR(↓)	C-Acc(↑)	ASR(↓)
BadNet	5%	89.38	99.99	**88.46**	99.70	84.72	5.00	83.29	14.53	83.94	2.49	86.03	2.48	87.96	5.32	87.62	**1.4**
	1%	89.86	99.93	85.34	97.42	84.45	28.07	83.09	3.76	84.53	11.73	85.74	1.97	**88.37**	43.50	88.22	**1.63**
	0.5%	89.62	99.58	**88.50**	98.00	84.08	50.31	82.85	**4.30**	83.68	19.46	85.16	9.28	87.78	45.86	87.16	5.53
Blended	5%	89.93	99.13	85.35	6.57	83.38	6.31	83.07	4.43	83.56	2.47	84.88	2.03	87.76	40.61	**86.81**	**0.02**
	1%	89.82	92.69	83.62	64.44	86.51	4.97	82.50	3.76	83.82	2.84	84.74	1.62	**87.85**	23.34	87.19	**0.47**
	0.5%	90.15	82.44	85.61	9.28	84.99	23.76	83.79	4.43	84.5	1.14	86.20	1.33	**88.53**	21.78	88.33	**0.31**
WaNet	5%	82.76	64.86	83.77	1.79	81.96	3.48	80.51	1.71	84.03	1.51	83.68	2.51	**85.22**	1.70	85.02	**0.92**
SSBA	5%	88.71	81.09	**87.49**	1.48	84.12	5.44	81.92	2.32	82.60	2.67	83.50	1.66	86.73	3.42	86.39	**1.14**
SIG	5%	89.74	97.71	**89.01**	0.54	83.71	24.43	82.96	0.79	83.95	0.91	85.15	0.34	87.33	15.06	87.75	**0.10**
	1%	89.04	95.34	86.9	7.91	83.33	23.77	82.09	7.76	83.05	33.62	83.85	4.92	**87.01**	87.91	86.16	**4.11**
	0.5%	89.46	76.17	82.32	61.93	85.96	67.30	82.78	8.86	83.94	11.08	84.93	10.91	**87.82**	60.51	84.60	**4.00**
LC	5%	89.56	99.71	**89.39**	98.16	87.19	**1.68**	82.19	4.21	83.62	4.81	84.53	5.37	87.47	3.96	86.46	3.67
	1%	89.76	99.96	**89.67**	99.72	86.36	90.44	82.84	3.79	83.21	14.02	84.83	12.87	86.96	97.90	87.15	**2.52**
	0.5%	88.24	99.04	**88.40**	98.52	81.40	60.78	79.64	**5.83**	81.72	21.26	82.38	19.17	85.64	74.03	84.63	13.68
Adaptive-Blend	0.3%	88.18	51.77	85.74	23.56	86.22	7.87	86.89	15.19	85.19	4.05	83.70	2.37	**89.10**	3.95	86.41	**1.33**
Average		88.95	89.29	86.64	51.27	84.56	26.91	82.69	5.71	83.69	8.94	84.62	5.26	**87.44**	35.26	86.66	**2.72**
Standard Deviation		1.81	15.02	2.30	44.43	1.68	27.75	1.58	4.24	**0.83**	9.60	1.04	5.41	1.03	32.48	1.18	**3.47**

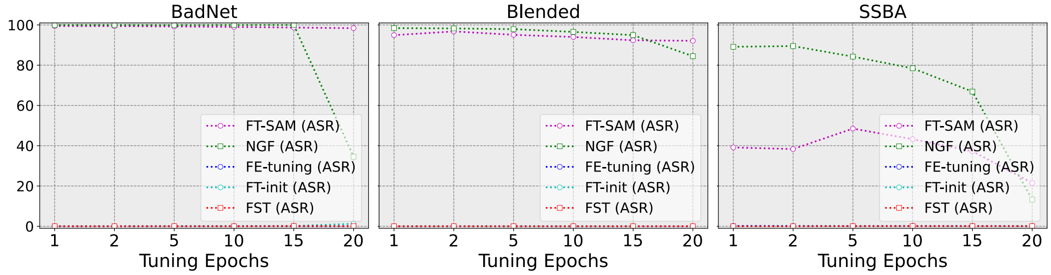

Figure 15: The ASR results of three representative attacks with various tuning epochs. Our experiments are conducted on Tiny-ImageNet with SwinTransformer.

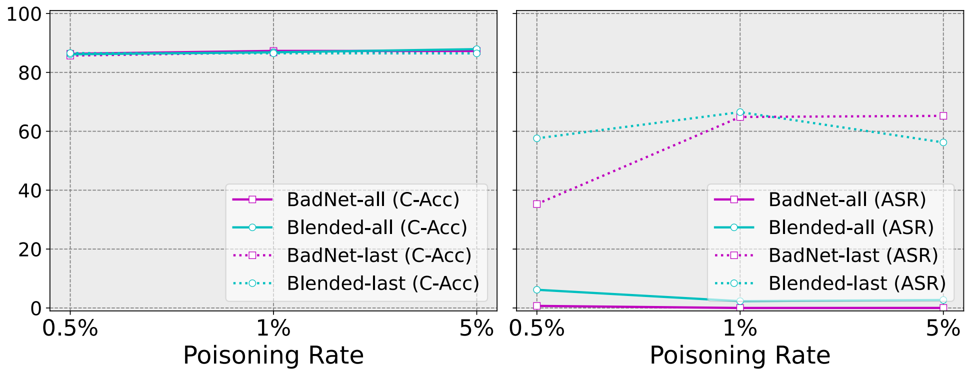

Figure 16: We compare regularizing the whole linear layers (denoted as -all) with regularizing only the last linear layer (denoted as -last). We evaluate on CIFAR-10 dataset using BadNet and Blended attacks with 3 poisoning rate settings. Experimental results demonstrate that we could achieve a superior purification performance by regularizing the whole linear layers than the last-layer-only regularization without sacrificing the model performance.