FLIP: A Provable Defense Framework for Backdoor Mitigation in Federated Learning

Kaiyuan Zhang[1], Guanhong Tao[1], Qiuling Xu[1], Siyuan Cheng[1], Shengwei An[1],
Yingqi Liu[1], Shiwei Feng[1], Pin-Yu Chen[2], Shiqing Ma[3], Xiangyu Zhang[1]
[1]Purdue University, [2]IBM Research, [3]Rutgers University
{zhan4057, taog, xu1230, cheng535, an93, liu1751, feng292, xyzhang}@cs.purdue.edu,
pin-yu.chen@ibm.com, sm2283@cs.rutgers.edu

Abstract

Federated Learning (FL) is a distributed learning paradigm that enables different parties to train a model together for better quality and strong privacy protection. In this scenario, individual participants may get compromised and perform backdoor attacks by poisoning the data (or gradients). Existing work on robust aggregation and certified FL robustness does not study how hardening benign clients can affect the global model (and the malicious clients). In this work, we theoretically analyze the connection among cross-entropy loss, attack success rate, and clean accuracy in this setting. Moreover, we propose a trigger reverse-engineering-based defense and show that our method can provide a guaranteed robustness increase (i.e., lower the attack success rate) without affecting benign accuracy. We conduct comprehensive experiments across different datasets and attack settings. Our results on eight competing SOTA defenses show the empirical superiority of our method on both single-shot and continuous FL backdoor attacks.

1. Introduction

Federated Learning (FL) is an emerging distributed learning paradigm. However, due to the decentralized nature of FL, recent studies demonstrate that individual participants may be compromised and become susceptible to backdoor attacks [2, 33, 38, 43] that aim to make any inputs stamped with a backdoor pattern misclassified as a target label. Such backdoors are becoming a prominent security threat to the real-world deployment of federated learning.

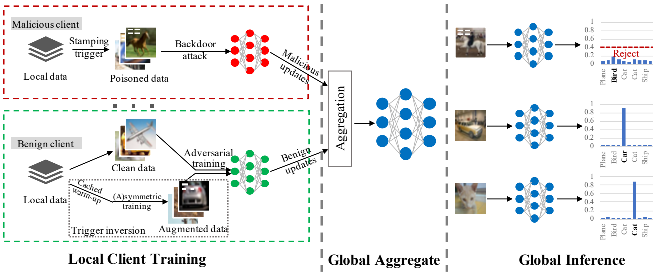

Figure 1. Overview of FLIP. The left upper part (red box) performs the malicious client backdoor attack and the left lower part (green box) illustrates the main steps of benign client model training.

Deficiencies of Existing Defenses. Existing FL backdoor defense works mainly fall into two categories, robust aggregation [10, 31] which detects and rejects malicious weights, and certified defense [7, 16, 30, 40] which provides robustness certification on backdoors with limited magnitude. Although there have been a lot of existing works on robust aggregation and empirical FL robust certification, most of them require inspecting model weights. This may cause information leakage of local clients by model inversion techniques. Besides, existing defense methods based on weights clustering [3, 28] either reject benign weights, causing degradation on mask task performance, or accept malicious weights, leaving backdoor effective.

FLIP. In this paper, we propose a **F**ederated **L**earn**I**ng **P**rovable defense framework (**FLIP**) equipped with a theoretical guarantee. For each benign local client, FLIP adversarially trains the local model on generated backdoor triggers that can cause misclassification, which is contradict to malicious local clients poisoning. Once local weights are aggregated in the global server, the injected backdoor features in aggregated global model will be mitigated by the benign clients' hardening. Therefore, FLIP can reduce the prediction confidence for backdoor samples from high to low. The overview of FLIP is shown in Figure 1. On top of the framework, we provide a theoretical analysis of how our training on a benign client can affect a malicious local client as well as the global model, which has not been studied in the literature. Certified accuracy is commonly used in evasion attacks that do not involve training. As backdoor attacks are training-time attacks, it is more reasonable to certify the behavior of models during training rather than the accuracy, which is the focus of our theoretical analysis.

Our Contributions. We make contributions on both theoretical and empirical fronts.

- We propose **FLIP**, a new provable defense framework that can provide a sufficient condition on the quality of trigger recovery such that the proposed defense is provably effective in mitigating backdoor attacks.
- We present a new theoretical understanding on formally quantifying the loss changes (with defense and without defense) in both backdoor and clean data evaluation.
- We empirically evaluate the effectiveness of our framework at scale across MNIST, Fashion-MNIST and CIFAR-10, trained with non-linear neural networks. The results significantly outperform prior works on the SOTA continuous FL backdoor attack setting. ASRs of SOTA defense techniques remain at 100% in most cases, while our technique can generally reduce ASRs to around 15%.
- We design an adaptive attack that is aware of the proposed defense and show that FLIP still remains effective and is resilient to adaptive attacks.
- We conduct ablation studies on each component of FLIP and validate FLIP is generally effective with various downstream trigger reverse techniques.

Threat Model. We consider FL backdoor attacks performed by malicious local clients, which manipulate local models by training with poisoned samples. On the benign clients side, we do not assume any knowledge about the ground truth trigger, benign clients generate the trigger based on received model weights and their local data (*non-i.i.d.*), then perform both standard training on clean data and adversarial training on augmented data (clean samples stamped with inverted triggers). On global server side, it knows nothing about the returned local weights and no assumption about the data, and thus there is no information leakage or privacy violence. On the attacker's side, the attack goal is to backdoor the global model with a high attack success rate and maintain a similar clean accuracy on the main task. We consider the practical oblivious but honest attack setting that a defender has no control over malicious clients, and they can perform any kind of attack, e.g. model replacement or scale weights. They can attack any round in FL, or even in the extreme case that they attack in every round after the global model converges (if an attack from the initial round, the model won't converge [43]), as long as attackers follow the federated learning protocol. In this paper, we consider static backdoors, i.e. patch backdoors [11]. Dynamic backdoors such as reflection backdoors [22], composite backdoors [18], and feature space backdoors [6] will be our future work.

2. Background & Related Work

Federated Learning Backdoor Attack and Defense. Given federated learning private local model training, the attacker could hijack some local clients and inject backdoor into global aggregated model [2, 38, 43]. To defend against federated learning backdoor attacks, a number of defense methods have been proposed. They mainly focus on robust aggregation [3, 31, 33] or detecting abnormal gradients update [10].

Federated Learning Robustness. Recently provable defense and certification methods have also been applied to federated learning. SparseFed [30] proposes global top-k update sparsification and provides a theoretical framework. CRFL [42] uses clipping and smoothing on model parameters to provide a sample-wise robustness certification.

3. Methodology & Theoretical analysis

We presented a new provable defense framework with theoretical guarantees and a novel trigger inversion technique under FL. The key insight is to combine trigger inversion techniques with FLIP, as long as the reversed trigger satisfies our given bound, then we can guarantee attack success rate will decrease and in the meantime the model can maintain similar accuracy on clean data.

3.1. Methodology

As illustrated in Figure 1, benign local clients apply trigger inversion techniques to recover the triggers, stamp them on clean images and assign the correct ground truth label, then combine with the clean data to perform model hardening (adversarial training). Trigger inversion is an effective and widely used technique in backdoor defense. Given a model and a few clean images, trigger inversion uses optimization to identify universal input perturbations that can flip the classification results of the clean images to a target class. Model hardening can force a model to unlearn unrobust (low-level) features. Cached Warm-up is designed for benign local clients and intends to reverse engineer triggers from the received global model (which could potentially be poisoned by malicious clients after each round aggregation). Existing work [34] shows that symmetric training of the two directions of a pair substantially alleviates oscillation and improves effectiveness. However, symmetric training needs the data from two directions of the selected pair, it is not feasible due to the *non-i.i.d.* nature of FL, then we propose asymmetric training when data is not sufficient for training to fit federated learning scenarios. More details can be found in Appendix A.4.

3.2. Theoretical analysis

In this section, we develop a theoretical analysis in an oracle view to study the effectiveness of our proposed defense in a simple but representative FL setting. The key insights are: (i) developing upper and lower bounds quantifying the cross-entropy loss changes on backdoored and clean data in the settings of with and without the defense in place (Theorem 1); (ii) showing a sufficient condition on the quality of trigger recovery such that the proposed defense is provably effective in mitigating backdoor attacks (Theorem 2); (iii) following (ii), we show that data inference with confidence thresholding on models trained with our proposed defense can provably reduce the backdoor attack success rate while maintaining similar accuracy on clean data.

Table 1. Table of main notations

Notation	Description
$\mathbf{x}_{k,j}, y_{k,j}$	k-th client device j-th data sample and its label
$q_{s,i}$	s-th sample i-th label index
W_r^k	k-th client device in r-th round weights
W	local model weights *without* defense
W'	local model weights *with* defense
τ	confidence threshold
R_b	number of rejected backdoor samples *without* defense
R_b'	number of rejected backdoor samples *with* defense
R_c	number of rejected clean samples *without* defense
R_c'	number of rejected clean samples *with* defense
δ	ground truth trigger
ϵ	difference of reversed trigger and ground truth trigger
$\delta + \epsilon$	various trigger inversion technique recovered trigger
$\mathbf{z}$	benign samples stamped with recovered trigger
$\mathcal{L}_g$	global model loss *without* defense
$\mathcal{L}_g'$	global model loss *with* defense

Setting. In theoretical analysis, we assume that we are under the FedAvg [25] protocol. In order to simplify analysis without the loss of generality, we assume there is one global server and two local clients; one is benign and the other is malicious. We conduct the analysis on multi-class classification using logistic regression. Our analysis focuses on per-step updates for local clients and a global learner.

Theorem 1 develops upper and lower bounds quantifying the loss changes on backdoored and clean data in the settings with and without the defense in place.

Theorem 1 (Bounds on Loss Changes) *Let $\mathcal{L}_g'$ denote the global model loss with defense, $\mathcal{L}_g$ as without defense, let $\Delta W = W' - W$ denote the weight differences with and without defense. The loss difference with and without defense can be upper and lower bounded by*

$$\min_t(\mathbf{x}\Delta W)_t - \sum_{i=1}^{I} q_i(\mathbf{x}\Delta W)_i \leq \mathcal{L}_g' - \mathcal{L}_g \leq \max_t(\mathbf{x}\Delta W)_t - \sum_{i=1}^{I} q_i(\mathbf{x}\Delta W)_i \tag{1}$$

The above theorem bounds the loss changes. The detailed proof is provided in Appendix A.2. To facilitate the analysis, we denote the upper bound as $\Delta \max\_loss$ and the lower bound as $\Delta \min\_loss$. To efficiently reduce the attack success rate and maintain the clean accuracy, we studied this lower bound on backdoor data, which indicates the minimal improvements on the backdoor defense. Similarly we studied the upper bound for clean data, as they indicates the worst-case accuracy degradation.

Denote the number of backdoor samples as n_b and the number of benign samples as n_c. Note backdoor samples are written as $\mathbf{x}_s + \delta$. By using Theorem 1, we have $\Delta \min\_loss = \min_t[\sum_{s=1}^{n_b}(\mathbf{x}_s + \delta)\Delta W]_t - \sum_{s=1}^{n_b}\sum_{i=1}^{I} q_{s,i}[(\mathbf{x}_s + \delta)\Delta W]_i$. And similarly on benign data, we have $\Delta \max\_loss$ $\mathbf{x}_s$, $\Delta \max\_loss = \max_t(\sum_{s=1}^{n_c}\mathbf{x}_s\Delta W)_t - \sum_{s=1}^{n_c}\sum_{i=1}^{I} q_{s,i}(\mathbf{x}_s\Delta W)_i$.

Next, we aim to develop a sufficient condition on the quality of trigger recovery such that the proposed defense is provably effective in mitigating backdoor attack and in the meantime maintaining similar accuracy on clean data, based on Theorem 1.

Theorem 2 (General Robustness Condition) *Let $\alpha =$*

$$\frac{\eta_r \sum_{s=1}^{n_b}\sum_{i=1}^{I}(q_{t^*,i} - q_{s,i})\{\mathbf{z_s}\sum_{j=1}^{n_1}[\mathbf{z_j}^T(\mathbf{q_j} - p(\mathbf{z_j}))]\}_i}{\mathbf{b}\left\{\eta_r \sum_{s=1}^{n_b}\sum_{i=1}^{I}(q_{t^*,i} - q_{s,i})\{\sum_{j=1}^{n_1}[\mathbf{z_j}^T(\mathbf{q_j} - p(\mathbf{z_j}))]\}_i\right\}}$$

where $\mathbf{b} = [b_1, ..., b_d]$, d is the sample dimension, let $b_v =$ $\mathrm{sign}\left\{\eta_r \sum_{s=1}^{n_b}\sum_{i=1}^{I}(q_{t^*,i} - q_{s,i})\sum_{j=1}^{n_1}[\mathbf{z_j}^T\mathbf{q_j} - p(\mathbf{z_j})]]_{i,v}\right\}$, on all dimensions v of the vector. For all $||\epsilon||_\infty \leq \alpha$, we have $\Delta \min\_loss \geq 0$.
And we have $\Delta \max\_loss \leq \eta_r \sum_{s=1}^{n_c}\sum_{i=1}^{I}(q_{t',i} - q_{s,i})\mathbf{x_s}\sum_{j=1}^{n_1}[\mathbf{z_j}^T(\mathbf{q_j} - p(\mathbf{z_j}))]_i$.

The detailed proof is provided in Appendix A.3. Denote $\mathbf{z}$ as $\mathbf{x} + \delta + \epsilon$ for simplicity, that is the benign sample stamped with the recovered trigger. Note that $\Delta \min\_loss \geq 0$ indicates that the defense is provably effective than without defense. Since benign local clients training can increase global model backdoor loss, and they have positive effects on mitigating malicious poisoning effect. The second condition $\Delta \max\_loss \leq \eta_r \sum_{s=1}^{n_c}\sum_{i=1}^{I}(q_{t',i} - q_{s,i})\mathbf{x_s}\sum_{j=1}^{n_1}[\mathbf{z_j}^T(\mathbf{q_j} - p(\mathbf{z_j}))]_i$ indicates that the defense is provably guarantee maintaining similar accuracy on clean data.

Corollary 1 *Assume ϵ satisfies Theorem 2, let n_b as backdoored samples, n_c as benign samples, τ as confidence threshold. Then the number of backdoored samples that are rejected is $R_{bd} = R_b' - R_b$, the number of benign samples that are rejected is $R_{bn} = R_c' - R_c$*

R_b' and R_b denote the rejected backdoor samples with and without defense. Similarly, R_c' and R_c denote the number of rejected benign samples with and without defense. With defense, R_b' is $\sum_{j=1}^{n_b}\mathbf{1}(\mathcal{L}_g + \Delta\min\_loss > \mathcal{L}_\tau)$; and without defense, R_b is $\sum_{j=1}^{n_b}\mathbf{1}(\mathcal{L}_g > \mathcal{L}_\tau)$. Thus, the exact value of rejected backdoored samples can be calculated through $R_{bd} = R_b' - R_b$. Similarly, the exact value of rejected benign samples can be calculated through $R_{bn} = R_c' - R_c$.

4. Experiment

Experiment Setup In this section, we empirically evaluate FLIP under two existing attack settings, i.e. single-shot attack [2] and continuous attack [43]. Single-shot backdoor attack means that every adversary only participates in one single round, while there can be more than one attacker. Continuous backdoor attack means in each round the attackers will be selected and continuously participate in the FL training from beginning to the end, which is a much more aggressive attack than single-shot. We compare the performance of FLIP with 8 state-of-the-art defenses, i.e. Krum [3], Bulyan Krum (Buly-Krum) [8], RFA [31], FoolsGold [10], Median [44], Trimmed Mean [44], Bulyan Trimmed Mean (Buly-Trim-M) [8], and FLTrust [4].

Evaluation on Backdoor Mitigation We consider the backdoor attack via model replacement approach where the attackers train their local models with backdoored samples.

The result of single-shot attack is shown in Table 2. Line 2 illustrates the attack performance with no defense. Observe that the single-shot attack can achieve more than 80% ASR throughout all the datasets while preserve high main task accuracy over 77%. The following lines show the defense performance of several existing techniques and last line denotes the result of FLIP. We can find that FLIP can reduce the ASR to below 8% on all 3 datasets and keep benign accuracy degradation within 5%. FLIP outperforms all the baselines on both MNIST and Fashion-MNIST while is slightly worse on CIFAR-10.

Table 2. Single-shot attack evaluation

Baselines	MNIST		F-MNIST		CIFAR-10	
	ACC	ASR	ACC	ASR	ACC	ASR
No Defense	97.55	80.12	81.01	96.72	77.52	80.46
Krum	97.50	0.35	79.49	10.79	77.00	9.51
Bulyan Krum	97.76	0.39	81.45	6.42	79.65	5.77
RFA	97.93	0.39	81.82	4.39	79.54	6.13
Trimmed Mean	97.81	0.38	81.81	5.40	79.95	5.81
Buly-Trim-M	97.02	90.75	79.84	99.38	66.69	84.05
FoolsGold	97.51	0.39	80.59	5.64	78.67	3.70
Median	97.76	0.37	81.76	5.97	64.31	2.39
FLTrust	97.26	0.48	79.92	7.69	**72.44**	**2.18**
FLIP	**96.05**	**0.13**	**78.20**	**3.16**	73.41	7.83

We show the results of continuous attack in Table 3. Continuous attack is more aggressive than the single-shot one where the ASR is raised 3-20% without defense. Note that the existing defense techniques all fail under the continuous attack setting. The ASR remains nearly 100% in most cases on MNIST and Fashion-MNIST and is higher than 63% on CIFAR-10. However, FLIP reduces ASR to a low level and keeps the accuracy degradation within an acceptance range. We observe that FLIP reduces the ASR on MNIST to 2% and the accuracy drop is within 2%. In Fashion-MNIST and CIFAR-10 dataset, the ASR is reduced to below 18% and 23% respectively while the accuracy decreases a bit more compared to the results of MNIST and single-shot attack. This is reasonable due to the reasons as follows. First the complexity of the dataset and continuous backdoor attacks may add to the difficulty of recovering good quality triggers. In addition, there is a trade-off between adversarial training accuracy and standard accuracy of a model as discussed in [36]. Benign local adversarial training can cause negative effects on the accuracy. However, we argue that FLIP still outperforms existing defenses as the ASR is largely reduced to a low level.

Table 3. Continuous attack evaluation

Baselines	MNIST		F-MNIST		CIFAR-10	
	ACC	ASR	ACC	ASR	ACC	ASR
No Defense	98.71	100.00	80.35	99.99	77.83	84.73
Krum	**97.59**	**0.14**	73.18	20.03	40.29	18.79
Bulyan Krum	98.15	94.01	82.17	99.46	68.61	97.31
RFA	98.54	100.00	85.69	100.00	79.39	63.10
Trimmed Mean	98.52	100.00	84.59	99.99	75.18	91.84
Buly-Trim-M	98.80	100.00	76.18	99.93	71.91	68.83
FoolsGold	97.91	99.99	80.58	99.98	74.57	78.30
Median	98.14	66.01	84.07	99.34	57.01	69.99
FLTrust	91.96	20.60	74.63	35.36	74.85	68.70
FLIP	96.62	1.93	**72.99**	**17.65**	**71.28**	**22.90**

Evaluation on the Same Setting as Theoretical Analysis. We conduct an experiment that follows the same setting as our assumptions to validate our theoretical analysis. Details can be found in Appendix A.5.2.

Adaptive Attacks We study an attack scenario where the adversary has the knowledge of FLIP, our results show that FLIP still mitigates the backdoor attacks in most cases. For those that ACC does degrade, the adaptive attack is not effective. Details can be found in Appendix A.5.3.

Ablation Study We study both adversarial training and thresholding is critical in FLIP framework, in Appendix A.5.5 and A.5.6. We study another different trigger inversion technique in FLIP, which can still mitigate backdoors, we find that FLIP is compatible with any trigger inversion techniques, in Appendix A.5.7. We study different sizes of triggers effect and show that our defense can cause a significant ASR degradation while maintaining comparable benign classification performance, in Appendix A.5.8. We also study different threshold influences on ACC and ASR and show the trade-off between attack success rate and accuracy, in Appendix A.5.9.

Acknowledgements

We thank the anonymous reviewers for their constructive comments. This research was supported, in part by IARPA TrojAI W911NF-19-S-0012, NSF 1901242 and 1910300, ONR N000141712045, N000141410468 and N000141712947. Any opinions, findings, and conclusions in this paper are those of the authors only and do not necessarily reflect the views of our sponsors.

References

[1] Sebastien Andreina, Giorgia Azzurra Marson, Helen Möllering, and Ghassan Karame. Baffle: Backdoor detection via feedback-based federated learning. In *2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS)*, pages 852–863. IEEE, 2021. 7

[2] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In *International Conference on Artificial Intelligence and Statistics*, pages 2938–2948. PMLR, 2020. 1, 2, 4, 7, 15

[3] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. *Advances in Neural Information Processing Systems*, 30, 2017. 1, 2, 4, 7, 17

[4] Xiaoyu Cao, Minghong Fang, Jia Liu, and Neil Zhenqiang Gong. Fltrust: Byzantine-robust federated learning via trust bootstrapping. *arXiv preprint arXiv:2012.13995*, 2020. 4, 7

[5] Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. Provably secure federated learning against malicious clients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 6885–6893, 2021. 7

[6] Siyuan Cheng, Yingqi Liu, Shiqing Ma, and Xiangyu Zhang. Deep feature space trojan attack of neural networks by controlled detoxification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1148–1156, 2021. 2, 7

[7] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning*, pages 1310–1320. PMLR, 2019. 1, 7

[8] El Mahdi El Mhamdi, Rachid Guerraoui, and Sébastien Rouault. The hidden vulnerability of distributed learning in Byzantium. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 3521–3530. PMLR, 10–15 Jul 2018. 4

[9] Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Gong. Local model poisoning attacks to byzantine-robust federated learning. In *29th {USENIX} Security Symposium*, 2020. 7

[10] Clement Fung, Chris J. M. Yoon, and Ivan Beschastnikh. The Limitations of Federated Learning in Sybil Settings. In *Symposium on Research in Attacks, Intrusion, and Defenses*, RAID, 2020. 1, 2, 4, 7, 18

[11] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017. 2, 7

[12] Andrew Hard, Kanishka Rao, Rajiv Mathews, Swaroop Ramaswamy, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*, 2018. 7

[13] Shanjiaoyang Huang, Weiqi Peng, Zhiwei Jia, and Zhuowen Tu. One-pixel signature: Characterizing cnn models for backdoor detection. In *European Conference on Computer Vision*, pages 326–341. Springer, 2020. 7

[14] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 14

[15] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 14

[16] Alexander Levine and Soheil Feizi. Deep partition aggregation: Provable defense against general poisoning attacks. *arXiv preprint arXiv:2006.14768*, 2020. 1, 7

[17] Yiming Li, Tongqing Zhai, Baoyuan Wu, Yong Jiang, Zhifeng Li, and Shutao Xia. Rethinking the trigger of backdoor attack. *arXiv preprint arXiv:2004.04692*, 2020. 7

[18] Junyu Lin, Lei Xu, Yingqi Liu, and Xiangyu Zhang. Composite backdoor attack for deep neural network by mixing existing benign features. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pages 113–131, 2020. 2

[19] Yang Liu, Mingyuan Fan, Cen Chen, Ximeng Liu, Zhuo Ma, Li Wang, and Jianfeng Ma. Backdoor defense with machine unlearning. *arXiv preprint arXiv:2201.09538*, 2022. 7

[20] Yingqi Liu, Wen-Chuan Lee, Guanhong Tao, Shiqing Ma, Yousra Aafer, and Xiangyu Zhang. Abs: Scanning neural networks for back-doors by artificial brain stimulation. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, CCS '19, page 1265–1282, New York, NY, USA, 2019. Association for Computing Machinery. 7, 17

[21] Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. Trojaning attack on neural networks. In *NDSS*, 2018. 7

[22] Yunfei Liu, Xingjun Ma, James Bailey, and Feng Lu. Reflection backdoor: A natural backdoor attack on deep neural networks. In *European Conference on Computer Vision*, pages 182–199. Springer, 2020. 2

[23] Shiqing Ma and Yingqi Liu. Nic: Detecting adversarial samples with neural network invariant checking. In *Proceedings of the 26th network and distributed system security symposium (NDSS 2019)*, 2019. 7

[24] Jayawant N Mandrekar. Receiver operating characteristic curve in diagnostic test assessment. *Journal of Thoracic Oncology*, 5(9):1315–1316, 2010. 16

[25] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017. 3, 7

[26] Thomas P. Minka. Estimating a dirichlet distribution. Technical report, 2000. 15

[27] D.S. Mitrinovic and P.M. Vasić. *Analytic Inequalities*. Die Grundlehren der mathematischen Wissenschaften in Einzeldarstellungen. Springer-Verlag, 1970. 10

[28] Thien Duc Nguyen, Phillip Rieger, Huili Chen, Hossein Yalame, Helen Möllering, Hossein Fereidooni, Samuel Marchal, Markus Miettinen, Azalia Mirhoseini, Shaza Zeitouni, et al. Flame: Taming backdoors in federated learning. *Cryptology ePrint Archive*, 2021. 1, 7, 17

[29] Mustafa Safa Ozdayi, Murat Kantarcioglu, and Yulia R Gel. Defending against backdoors in federated learning with robust learning rate. *arXiv preprint arXiv:2007.03767*, 2020. 7

[30] Ashwinee Panda, Saeed Mahloujifar, Arjun Nitin Bhagoji, Supriyo Chakraborty, and Prateek Mittal. Sparsefed: Mitigating model poisoning attacks in federated learning with sparsification. In *International Conference on Artificial Intelligence and Statistics*, pages 7587–7624. PMLR, 2022. 1, 2, 7

[31] Krishna Pillutla, Sham M. Kakade, and Zaid Harchaoui. Robust aggregation for federated learning. *IEEE Transactions on Signal Processing*, 70:1142–1154, 2022. 1, 2, 4, 7

[32] Guangyu Shen, Yingqi Liu, Guanhong Tao, Shengwei An, Qiuling Xu, Siyuan Cheng, Shiqing Ma, and Xiangyu Zhang. Backdoor scanning for deep neural networks through k-arm optimization. In *International Conference on Machine Learning*, pages 9525–9536. PMLR, 2021. 7

[33] Ziteng Sun, Peter Kairouz, Ananda Theertha Suresh, and H Brendan McMahan. Can you really backdoor federated learning? *arXiv preprint arXiv:1911.07963*, 2019. 1, 2, 7

[34] Guanhong Tao, Yingqi Liu, Guangyu Shen, Qiuling Xu, Shengwei An, Zhuo Zhang, and Xiangyu Zhang. Model orthogonalization: Class distance hardening in neural networks for better security. In *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2022. 2, 7, 13, 14

[35] Vale Tolpegin, Stacey Truex, Mehmet Emre Gursoy, and Ling Liu. Data poisoning attacks against federated learning systems. In *European Symposium on Research in Computer Security*, pages 480–501. Springer, 2020. 7

[36] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In *International Conference on Learning Representations*, 2019. 4, 17

[37] Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y. Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 707–723, 2019. 7, 13

[38] Hongyi Wang, Kartik Sreenivasan, Shashank Rajput, Harit Vishwakarma, Saurabh Agarwal, Jy-yong Sohn, Kangwook Lee, and Dimitris Papailiopoulos. Attack of the tails: Yes, you really can backdoor federated learning. *Advances in Neural Information Processing Systems*, 33, 2020. 1, 2, 16, 18

[39] Ren Wang, Gaoyuan Zhang, Sijia Liu, Pin-Yu Chen, Jinjun Xiong, and Meng Wang. Practical detection of trojan neural networks: Data-limited and data-free cases. In *European Conference on Computer Vision*, pages 222–238. Springer, 2020. 7

[40] Chong Xiang, Arjun Nitin Bhagoji, Vikash Sehwag, and Prateek Mittal. {PatchGuard}: A provably robust defense against adversarial patches via small receptive fields and masking. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2237–2254, 2021. 1, 7

[41] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017. 14

[42] Chulin Xie, Minghao Chen, Pin-Yu Chen, and Bo Li. Crfl: Certifiably robust federated learning against backdoor attacks. In *International Conference on Machine Learning*, pages 11372–11382. PMLR, 2021. 2, 7

[43] Chulin Xie, Keli Huang, Pin-Yu Chen, and Bo Li. Dba: Distributed backdoor attacks against federated learning. In *International Conference on Learning Representations*, 2019. 1, 2, 4, 7, 15

[44] Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In *International Conference on Machine Learning*, pages 5650–5659. PMLR, 2018. 4

A. Appendix

We provide a simple table of contents below for easier navigation of the appendix.

CONTENTS

A.1. Background Related Work

Backdoor Attack and Defense In general, the goal of backdoor attack is to inject a backdoor pattern and a target label to the dataset used for training. During testing phase, any inputs with such pattern will be classified as the target label. There are a number of existing backdoor attacks, like patch attacks [11, 21], feature space attacks [6], etc. To identify whether a model is poisoned, existing works inverse triggers [19, 20, 32, 34, 37], compare difference between clean models and backdoored models [13, 39], and some methods detect and reject inputs stamped with triggers [17, 23].

Federated Learning Backdoor Attack and Defense. Federated learning [12, 25] is recently proposed to train a deep learning model without direct access to training data due to privacy concerns. Federated learning distributes the model training to multiple local clients and iteratively aggregates the local models to a shared global model. Since FL local model training is private, attackers could hijack some local clients and inject backdoor into global aggregated model [2, 9, 35, 43]. To defend against FL backdoor attacks, a number of defenses have been proposed. They mainly focus on robust aggregation [3, 28, 31, 33], robust learning rate method [29] or detecting abnormal gradients update [1, 4, 10]. FLAME [28] proposes a resilient aggregation framework that applies model clustering and clipping to minimize the injected noise and ensure backdoor elimination. FLTrust [4] proposes to inject noise to global model to neutralize backdoors and apply dynamic clustering and adaptive clipping to preserve the benign accuracy.

Federated Learning Robustness. Certified and provable defenses have been proposed to analyze adversarial robustness. Our certified loss changes for backdoor mitigation conduct on training-time, while certified accuracy is more used in test-time attack. Certifying changes in loss is more suitable for backdoor attack since it involves model training; while certified accuracy is more used in evasion attacks that do not involve training, it is less meaningful for backdoor attacks.

Random smooth [7] provides certifiably robustness to adversarial examples on l_2 distance. PatchGuard [40] provides provable robustness against localized adversarial patches. Deep partition aggregation [16] propose a certified defense against general poisoning. Recently provable defense and certification methods have also been applied to federated learning. SparseFed [30] proposes global top-k update sparsification and provides a theoretical framework for analyzing the robustness of defenses against poisoning attacks. Ensemble federated learning [5] proposes to learn multiple global models and provide provably secure against malicious clients. CRFL [42] uses clipping and smoothing on model parameters to provide a sample-wise robustness certification on backdoors with limited magnitude.

A.2. Proof of Bounds on Loss Changes

In this section, we will present the bound on loss changes, formulate the benign local clients training and global model aggregation process, and then provide the detailed proofs for our Theorem 1 that are related to loss changes bound. Note that, we list all the notations used in the paper in Table 4.

Generalize Proof to complex model architectures Given that extending to multi-layer is not trivial, this will make the proof becomes challenging to generalize, we focus on multi-class logistic regression (one linear layer with softmax function and cross-entropy loss), which is a convex classification problem. Meanwhile, we empirically evaluate the effectiveness of our framework at scale across MNIST, Fashion-MNIST, and CIFAR-10, trained with non-linear neural networks. It shows that the results significantly outperform prior works on the SOTA continuous FL backdoor attack setting.

Throughout this paper, ”clean training” refers to benign local clients training with clean data; ”adversarial training” refers to benign local clients apply trigger inversion techniques to get reversed trigger, then stamp the trigger to their local clean image and assign with ground truth clean label to get the augmented dataset, then train with the augmented dataset.

In benign clients, we train with defense technique to gen-

Table 4. Table of notations

Notation	Description
$\mathbf{x}_{k,j}$, $y_{k,j}$	the k-th client device j-th data sample and its label
$q_{s,i}$	the s-th sample i-th label index
W_r^k	the k-th client device in r-th round weights
W	local model weights *without* defense
W'	local model weights *with* defense
τ	confidence threshold
R_b	the number of rejected backdoor samples *without* defense
R_b'	the number of rejected backdoor samples *with* defense
$R_{bd} = R_b' - R_b$	the number of backdoored samples that are rejected after defense applied
R_c	the number of rejected benign samples *without* defense
R_c'	the number of rejected benign samples *with* defense
$R_{bn} = R_c' - R_c$	the number of benign samples that are rejected after defense applied
δ	the ground truth trigger
$\delta + \epsilon$	the various trigger inversion technique recovered trigger
ϵ	the difference between the reversed trigger and the ground truth trigger
$\mathbf{z} = \mathbf{x} + \delta + \epsilon$	$\mathbf{z}$ denotes the benign sample stamped with the recovered trigger
$\mathcal{L}_g$	the global model loss *without* defense
$\mathcal{L}_g'$	the global model loss *with* defense

erate trigger, then do adversarial training and submit gradients to global server. Given model parameter W of one linear layer, k-th device holds the n_k training data $\mathbf{x}_{k,n_k}$, then denoted the loss as $\mathcal{L}(W; \mathbf{x}_{k,n_k})$. Let $Y \in \{0,1\}_i$ denote a one-hot vector of local samples. For $\mathbf{x}$, we denote $\mathbf{x}W$ as the output of the linear layer, $p_i(\mathbf{x}) = softmax(\mathbf{x}W + b)_i$ as the normalized probability for class i (the output of the softmax function). b or bias is omitted in following equations for simplicity, but it would still work if added. For one example the cross-entropy loss is calculated as:

$$\mathcal{L}(x) = -\sum_i Y_i log p_i(\mathbf{x}) \tag{2}$$

$$= -\sum_i Y_i log(softmax(\mathbf{x}W)_i) \tag{3}$$

We define G as the gradient for one sample:

$$G(\mathbf{x}) = \nabla l(W; \mathbf{x}, y) = \frac{d\mathcal{L}}{dw}(\mathbf{x}) = \mathbf{x}^T(p(\mathbf{x}) - Y) \tag{4}$$

Similarly, when defense technique get reversed trigger and stamp it on clean image, then we get the augmented dataset, denote is as $\mathbf{x}_{aug}$, then the gradient on augmented dataset G' can be written as:

$$G'(\mathbf{x}_{aug}) = \nabla l(W; \mathbf{x}_{aug}, y) = \mathbf{x}_{aug}^T(p(\mathbf{x}_{aug}) - Y)) \tag{5}$$

Here, we describe one around (say the r-th) of the standard $FedAvg$ algorithm. When the benign device in k-th receive the global weights W_r, and then performs E $(= 1)$ local updates (lets $W_r^k = W_r$), in benign clients, we training on both clean dataset and augmented dataset:

$$\begin{aligned} W_{r+1}^k &\leftarrow W_r^k - \eta_r \nabla F_k(W_r^k, \xi_r^k) \\ &\leftarrow W_r^k - \eta_r[\sum_{j=1}^{n_k}[\mathbf{x}_j^{T,k}(p(\mathbf{x}_j^k) - Y_j)] \\ &+ \sum_{j=1}^{n_k}[\mathbf{x}_{aug,j}^{T,k}(p(\mathbf{x}_{aug,j}^k) - Y_j)]] \\ &\leftarrow W_r^k - \eta_r \sum_{j=1}^{n_k}[\mathbf{x}_j^{T,k}(p(\mathbf{x}_j^k) - Y_j)] \\ &- \eta_r \sum_{j=1}^{n_k}[\mathbf{x}_{aug,j}^{T,k}(p(\mathbf{x}_{aug,j}^k) - Y_j)] \end{aligned} \tag{6}$$

where η_r is the learning rate (a.k.a. step size), n_k is the number of samples in k-th client.

In global server, define δ as the malicious clients generated trigger, $\delta + \epsilon$ as the benign clients generated trigger, then we can represent backdoored sample as $(\mathbf{x} + \delta)$ and augmented sample as $(\mathbf{x} + \delta + \epsilon)$. Benign clients updates can be written as:

$$\begin{aligned} W_{r+1}^k \leftarrow W_r^k &- \eta_r \sum_{j=1}^{n_k}[\mathbf{x}_j^{T,k}(p(\mathbf{x})_j^k) - Y_j^k)] \\ &- \eta_r \sum_{j=1}^{n_k}[(\mathbf{x}_j + \delta + \epsilon)^{T,k}(p(\mathbf{x}_j + \delta + \epsilon)^k) - Y_j^k)] \end{aligned} \tag{7}$$

In the threat model Section 1, we consider the practical oblivious but honest attack setting that a defender has no control on malicious clients and they can perform any kinds of attack, as long as attackers follow the federated learning protocol. Thus, we represent the malicious clients updates as W_M.

After each local finished their training, they submit their model updates to global. Then global aggregation step performs

$$W_{r+1} \leftarrow \eta_r \sum_{k=1}^{N} g_k W_{r+1}^{k} \tag{8}$$

g_k is the weight of the k-th device. In order to simplify, here we take g_k as 1 and assume we only have two clients (N=2), $k = 1$ is benign client, n_1 denotes the number of samples in this benign client. Then the aggregated global weight are the each local weights aggregate together. Then the aggregated global weight can be written as

$$\begin{aligned}
W_{r+1} &= \sum_{k=1}^{N} W_{r+1}^{k} \\
&= W_{r+1}^{1} + W_{r+1}^{2} \\
&= -\eta_r \sum_{j=1}^{n_1} [\mathbf{x}_j^{T,k}(p(x)_j^k) - Y_j^k)] \\
&- \eta_r \sum_{j=1}^{n_1} [(\mathbf{x}_j + \delta + \epsilon)^{T,1}(p(\mathbf{x}_j + \delta + \epsilon)^1 - Y_j^1)] \\
&+ 2W_r - W_M \\
&= -\eta_r \sum_{j=1}^{n_1} [\mathbf{x}_j^{T,k}(p(x)_j^k) - Y_j^k)] \\
&- \eta_r \sum_{j=1}^{n_1} [(\mathbf{x}_j + \delta + \epsilon)^{T,1}(p(\mathbf{x}_j + \delta + \epsilon)^1 - Y_j^1)] \\
&+ 2W_r - W_M
\end{aligned} \tag{9}$$

When we consider the without defense setting, $\delta + \epsilon$ not exists, in round $t+1$, W_{t+1} the global weight without local weights can be written as

$$W_{r+1} = -\eta_r \sum_{j=1}^{n_1} [\mathbf{x}_j^{T,1}(p(x)_j^1 - Y_j^1)] + 2W_r - W_M \tag{10}$$

When we consider the with defense setting, $\delta + \epsilon$ exists, in round $t+1$, W_{t+1} the global weight without local weights can be written as

$$\begin{aligned}
W'_{r+1} &= -\eta_r \sum_{j=1}^{n_1} [\mathbf{x}_j^{T,1}(p(x)_j^1 - Y_j^1)] \\
&- \eta_r \sum_{j=1}^{n_1} [(\mathbf{x}_j + \delta + \epsilon)^{T,1}(p(\mathbf{x}_j + \delta + \epsilon)^1 - Y_j^1)] \\
&+ 2W_r - W_M
\end{aligned} \tag{11}$$

The difference between with defense and without defense training is exactly how much adversarial training in benign will influence other clients, it can be written as

$$\begin{aligned}
W'_{r+1} - W_{r+1} = -\eta_r \sum_{j=1}^{n_1} [(\mathbf{x}_j + \delta + \epsilon)^{T,1}(p(\mathbf{x}_j + \delta + \epsilon)^1 \\
- Y_j^1)]
\end{aligned} \tag{12}$$

Given model parameter W of one linear layer, the k-th device holds n_k training data $\{\mathbf{x}_{k,j}, y_{k,j}\}_{j=1}^{n_k}$. We denote the loss as $\mathcal{L}(W; \{\mathbf{x}_{k,j}, y_{k,j}\}_{j=1}^{n_k})$. Denote $\mathbf{x}W$ as the output of the linear layer, $P_i(x) = \textit{softmax}(\mathbf{x}W + b)_i$ as the normalized probability for class i (the output of the *softmax* function). We omit b (bias) in the following theoretical analysis for simplicity. Adding the bias term to our analysis is straightforward.

Global softmax cross-entropy loss function can be written as:

$$\begin{aligned}
\mathcal{L}_{global} &= -\sum_{i=1}^{I} q_i log(p_i) \\
&= -\sum_{i=1}^{I} q_i logsoftmax(\mathbf{x}W)_i \\
&= -\sum_{i=1}^{I} q_i log(\frac{e^{(\mathbf{x}W)_i}}{\sum_{t=1}^{I} e^{(\mathbf{x}W)_t}}) \\
&= -\sum_{i=1}^{I} q_i(\mathbf{x}W)_i + log(\sum_{t=1}^{I} e^{(\mathbf{x}W)_t}) \\
&= -\sum_{i=1}^{I} q_i(\mathbf{x}W)_i + log(\sum_{t=1}^{I} e^{(\mathbf{x}W)_t})
\end{aligned} \tag{13}$$

Since we want to compare the loss changes in two different cases (e.g. with defense and without defense setting), to observe if the dedution of the loss increase or decrease, here we let the two losses (say $\mathcal{L}'_g$ is with defense, $\mathcal{L}_g$ is without defense) deduct each other:

$$\begin{aligned}\mathcal{L}'_g - \mathcal{L}_g = &-\sum_{i=1}^{I} q_i(\mathbf{x}W')_i + log(\sum_{t=1}^{I} e^{(\mathbf{x}W')_t}) \\ &+ \sum_{i=1}^{I} q_i(\mathbf{x}W)_i - log(\sum_{t=1}^{I} e^{(\mathbf{x}W)_t}) \\ = &-\sum_{i=1}^{I} q_i[(W'-W)\mathbf{x}]_i \\ &+ log(\sum_{t=1}^{I} e^{(\mathbf{x}W')_t}) - log(\sum_{t=1}^{I} e^{(\mathbf{x}W)_t}) \\ = &-\sum_{i=1}^{I} q_i[(W'-W)\mathbf{x}]_i + log(\frac{\sum_{t=1}^{I} e^{(\mathbf{x}W')_t}}{\sum_{t=1}^{I} e^{(\mathbf{x}W)_t}})\end{aligned} \tag{14}$$

Since both $\sum_{t=1}^{I} e^{(\mathbf{x}W')_t}$ and $\sum_{t=1}^{I} e^{(\mathbf{x}W)_t}$ are a sequence of positive numbers. Then from [27] we can have an inequality of

$$\min_{1\le t\le I} \frac{e^{(\mathbf{x}W')_t}}{e^{(\mathbf{x}W)_t}} \le \frac{\sum_{t=1}^{I} e^{(\mathbf{x}W')_t}}{\sum_{t=1}^{I} e^{(\mathbf{x}W)_t}} \le \max_{1\le t\le I} \frac{e^{(\mathbf{x}W')_t}}{e^{(\mathbf{x}W)_t}} \tag{15}$$

Proof.

If we denote $m = \min_t \frac{e^{(\mathbf{x}W')_t}}{e^{(\mathbf{x}W)_t}}$ and $M = \max_t \frac{e^{(\mathbf{x}W')_t}}{e^{(\mathbf{x}W)_t}}$, then we have successively

$$m \le \frac{e^{(\mathbf{x}W')_t}}{e^{(\mathbf{x}W)_t}} \le M \tag{16}$$

$$m \cdot e^{(\mathbf{x}W)_t} \le e^{(\mathbf{x}W')_t} \le M \cdot e^{(\mathbf{x}W)_t} \tag{17}$$

$$m \cdot \sum_{t=1}^{I} e^{(\mathbf{x}W)_t} \le \sum_{t=1}^{I} e^{(\mathbf{x}W')_t} \le M \cdot \sum_{t=1}^{I} e^{(\mathbf{x}W)_t} \tag{18}$$

$$m \le \frac{\sum_{t=1}^{I} e^{(\mathbf{x}W')_t}}{\sum_{t=1}^{I} e^{(\mathbf{x}W)_t}} \le M \tag{19}$$

Then we can get

$$\min_{1\le t\le I} \frac{e^{(\mathbf{x}W')_t}}{e^{(\mathbf{x}W)_t}} \le \frac{\sum_{t=1}^{I} e^{(\mathbf{x}W')_t}}{\sum_{t=1}^{I} e^{(\mathbf{x}W)_t}} \le \max_{1\le t\le I} \frac{e^{(\mathbf{x}W')_t}}{e^{(\mathbf{x}W)_t}} \tag{20}$$

By using log function monotonicity property, we can get an inequality of

$$logm \le log(\frac{\sum_{t=1}^{I} e^{(\mathbf{x}W')_t}}{\sum_{t=1}^{I} e^{(\mathbf{x}W)_t}}) \le logM \tag{21}$$

So the deduction of $\mathcal{L}'_g - \mathcal{L}_g$ can be written as:

$$\begin{aligned}logm - \sum_{i=1}^{I} q_i[\mathbf{x}(W'-W)]_i \le \mathcal{L}'_g - \mathcal{L}_g \le \\ logM - \sum_{i=1}^{I} q_i[\mathbf{x}(W'-W)]_i\end{aligned} \tag{22}$$

$$\begin{aligned}log\min_t \frac{e^{(\mathbf{x}W')_t}}{e^{(\mathbf{x}W)_t}} - \sum_{i=1}^{I} q_i[\mathbf{x}(W'-W)]_i \le \mathcal{L}'_g - \mathcal{L}_g \le \\ log\max_t \frac{e^{(\mathbf{x}W')_t}}{e^{(\mathbf{x}W)_t}} - \sum_{i=1}^{I} q_i[\mathbf{x}(W'-W)]_i\end{aligned} \tag{23}$$

Denote the left hand side of above formula as $\Delta \min \_loss$, denote the inequality's right hand side value as $\Delta \max \_loss$.

$$\begin{aligned}\Delta \min \_loss &= log\min_t \frac{e^{(\mathbf{x}W')_t}}{e^{(\mathbf{x}W)_t}} - \sum_{i=1}^{I} q_i[\mathbf{x}(W'-W)]_i \\ &= \min_t log\frac{e^{(\mathbf{x}W')_t}}{e^{(\mathbf{x}W)_t}} - \sum_{i=1}^{I} q_i[\mathbf{x}(W'-W)]_i \\ &= \min_t log e^{[\mathbf{x}(W'-W)]_t} - \sum_{i=1}^{I} q_i[\mathbf{x}(W'-W)]_i \\ &= \min_t [\mathbf{x}(W'-W)]_t - \sum_{i=1}^{I} q_i[\mathbf{x}(W'-W)]_i\end{aligned} \tag{24}$$

Then we can get the lower bound and upper bound of $\mathcal{L}'_g - \mathcal{L}_g$

$$\begin{aligned}\min_t [\mathbf{x}(W'-W)]_t - \sum_{i=1}^{I} q_i[\mathbf{x}(W'-W)]_i \le \mathcal{L}'_g - \mathcal{L}_g \le \\ \max_t [\mathbf{x}(W'-W)]_t - \sum_{i=1}^{I} q_i[\mathbf{x}(W'-W)]_i\end{aligned} \tag{25}$$

Let $\mathcal{L}'_g$ denote the global model loss with defense, $\mathcal{L}_g$ as without defense, let $\Delta W = W' - W$ denote the weight differences with and without defense. The loss difference with and without defense can be upper and lower bounded by (as shown in Theorem 1)

$$\begin{aligned}\min_t (\mathbf{x}\Delta W)_t - \sum_{i=1}^{I} q_i(\mathbf{x}\Delta W)_i \le \mathcal{L}'_g - \mathcal{L}_g \le \\ \max_t (\mathbf{x}\Delta W)_t - \sum_{i=1}^{I} q_i(\mathbf{x}\Delta W)_i\end{aligned} \tag{26}$$

To facilitate the analysis, we denote the upper bound as $\Delta \max \_loss$ and the lower bound as $\Delta \min \_loss$. To efficiently reduce the attack success rate and maintain the clean accuracy, we studied this lower bound on backdoor data, which indicates the minimal improvements on the backdoor defense. Similarly we studied the upper bound for clean data, as they indicates the worst-case accuracy degradation.

Denote the number of backdoor samples as n_b and the number of benign samples as n_c. Note backdoor samples are written as $\mathbf{x}_s + \delta$. By using Theorem 1, we have $\Delta \min \_loss = \min_t[\sum_{s=1}^{n_b}(\mathbf{x}_s + \delta)\Delta W]_t - \sum_{s=1}^{n_b}\sum_{i=1}^{I} q_{s,i}[(\mathbf{x}_s + \delta)\Delta W]_i$. And similarly on benign data, we have $\Delta \max \_loss$ $\mathbf{x}_s$, $\Delta \max \_loss = \max_t(\sum_{s=1}^{n_c}\mathbf{x}_s \Delta W)_t - \sum_{s=1}^{n_c}\sum_{i=1}^{I} q_{s,i}(\mathbf{x}_s \Delta W)_i$.

A.3. Proof of General Robustness Condition

In this section, we will present general condition of robustness on trigger generation, formulate Δmin_loss on backdoored data and Δmax_loss on clean data, and then provide the detailed proofs for our Theorem 2 that are related to general robustness condition.

Our intuition is that we want the loss to increase more on backdoored data, and increase less on clean data. This means after applying defense, the global server loss in backdoored data will increase and the loss in clean data will change within a constant range. Accordingly, when evaluate on n_b backdoored data, we want the lower bound at least greater than 0, $\Delta min\_loss \geq 0$. When evaluate on n_c clean data, we want the upper bound $\Delta max\_loss \leq \zeta$, here ζ is a constant. In evaluation, denote global server has n_b backdoored data and n_c clean data for testing.

When evaluating on n_b backdoored data

$$\begin{aligned}\mathcal{L}'_g - \mathcal{L}_g \geq \min_t[&\sum_{s=1}^{n_b}(\mathbf{x}_s + \delta)(W' - W)]_t \\ &- \sum_{s=1}^{n_b}\sum_{i=1}^{I} q_{s,i}[(\mathbf{x}_s + \delta)(W' - W)]_i \\ &= \Delta \min \_loss \geq 0\end{aligned} \tag{27}$$

When evaluating on n_c clean data

$$\begin{aligned}\mathcal{L}'_g - \mathcal{L}_g \leq \max_t[&\sum_{s=1}^{n_c}\mathbf{x}_s(W' - W)]_t \\ &- \sum_{s=1}^{n_c}\sum_{i=1}^{I} q_{s,i}[\mathbf{x}_s(W' - W)]_i \\ &= \Delta \max \_loss \leq \zeta\end{aligned} \tag{28}$$

Since previous results we know $W'_{r+1} - W_{r+1}$ can be represented as

$$\begin{aligned}W'_{r+1} - W_{r+1} &= -\eta_r \sum_{j=1}^{n_1}[(\mathbf{x}_j + \delta + \epsilon)^T(p(\mathbf{x}_j + \delta + \epsilon) - Y_j)] \\ &= \eta_r \sum_{j=1}^{n_1}[(\mathbf{x}_j + \delta + \epsilon)^T(Y_j - p(\mathbf{x}_j + \delta + \epsilon))]\end{aligned} \tag{29}$$

Give that q_i is a one-hot vector, we denote the value of $\min_t$ as q_{t^*}, then substitute $(W' - W)$ in Δmin_loss we can get

$$\begin{aligned}\Delta \min \_loss = \min_t[&\sum_{s=1}^{n_b}(\mathbf{x}_s + \delta)(W' - W)]_t \\ &- \sum_{s=1}^{n_b}\sum_{i=1}^{I} q_{s,i}[(\mathbf{x}_s + \delta)(W' - W)]_i \\ &= \eta_r \sum_{s=1}^{n_b}\sum_{i=1}^{I} q_{t^*,i}[(\mathbf{x}_s + \delta)\sum_{j=1}^{n_1}[(\mathbf{x}_j + \delta + \epsilon)^T(Y_j \\ &- p(\mathbf{x}_j + \delta + \epsilon))]]_i \\ &- \eta_r \sum_{s=1}^{n_b}\sum_{i=1}^{I} q_{s,i}[(\mathbf{x}_s + \delta)\sum_{j=1}^{n_1}[(\mathbf{x}_j + \delta + \epsilon)^T(Y_j \\ &- p(\mathbf{x}_j + \delta + \epsilon))]]_i \\ &= \eta_r \sum_{s=1}^{n_b}\sum_{i=1}^{I}(q_{t^*} - q_{s,i})[(\mathbf{x}_s + \delta)\eta_r \sum_{j=1}^{n_1}[(\mathbf{x}_j \\ &+ \delta + \epsilon)^T(Y_j - p(\mathbf{x}_j + \delta + \epsilon))]]_i\end{aligned} \tag{30}$$

Let $z_s = \mathbf{x}_s + \delta + \epsilon$ and $z_j = \mathbf{x}_j + \delta + \epsilon$, then the $\Delta \min \_loss$ is

$$\begin{aligned}\Delta \min \_loss = \eta_r \sum_{s=1}^{n_b}\sum_{i=1}^{I}(q_{t^*} - q_{s,i})[(z_s \\ - \epsilon)\sum_{j=1}^{n_1}[z_j^T(Y_j - p(z_j))]]_i\end{aligned} \tag{31}$$

Let

$$\begin{aligned}f(\epsilon) &= \Delta \min \_loss \\ &= \eta_r \sum_{s=1}^{n_b}\sum_{i=1}^{I}(q_{t^*,i} - q_{s,i})\{(z_s - \epsilon)\sum_{j=1}^{n_1}[z_j^T(Y_j - p(z_j))]\}_i\end{aligned} \tag{32}$$

Compute the gradient of $f(\epsilon)$, we have

$$\frac{\nabla f(\epsilon)}{\nabla \epsilon} = -\eta_r \sum_{s=1}^{n_b}\sum_{i=1}^{I}(q_{t^*,i} - q_{s,i})[\sum_{j=1}^{n_1}[z_j^T(Y_j - p(z_j))]]_i \tag{33}$$

Let $||\epsilon||_\infty \leq \alpha$, and that $f(\epsilon)$ is a linear function, we know the minimal value of $f(\epsilon)$ is achieved when

$$\epsilon_k = \alpha \text{ sign} \left\{ \eta_r \sum_{s=1}^{n_b} \sum_{i=1}^{I} (q_{t^*,i} - q_{s,i}) \sum_{j=1}^{n_1} [z_j^T (Y_j - p(z_j))]]_{i,k} \right\} \tag{34}$$

For simplicity, denote b_k as

$$b_k = \text{sign} \left\{ \eta_r \sum_{s=1}^{n_b} \sum_{i=1}^{I} (q_{t^*,i} - q_{s,i}) \sum_{j=1}^{n_1} [z_j^T (Y_j - p(z_j))]]_{i,k} \right\} \tag{35}$$

and the vector as $\mathbf{b} = [b_1, ..., b_d]$. The minimal condition is thus $\epsilon = \alpha \mathbf{b}$.

Replace $\epsilon = \alpha \mathbf{b}$ into eq. (32), and in order to be consistent with previous section in main text Theorem 2, we use $\mathbf{q_j}$ to replace Y_j, we have

$$f(\epsilon) \geq \\ \eta_r \sum_{s=1}^{n_b} \sum_{i=1}^{I} (q_{t^*,i} - q_{s,i}) \{(\mathbf{z_s} - \alpha \mathbf{b}) \sum_{j=1}^{n_1} [\mathbf{z_j}^T (\mathbf{q_j} - p(\mathbf{z_j}))]\}_i \tag{36}$$

The sufficient condition of $f(\epsilon) \geq 0$ is thus

$$\eta_r \sum_{s=1}^{n_b} \sum_{i=1}^{I} (q_{t^*,i} - q_{s,i}) \{(\mathbf{z_s} - \alpha \mathbf{b}) \sum_{j=1}^{n_1} [\mathbf{z_j}^T (\mathbf{q_j} - p(\mathbf{z_j}))]\}_i \geq 0 \tag{37}$$

$$\begin{aligned} &\alpha \mathbf{b} \{ \eta_r \sum_{s=1}^{n_b} \sum_{i=1}^{I} (q_{t^*,i} - q_{s,i}) \{ \sum_{j=1}^{n_1} [\mathbf{z_j}^T (\mathbf{q_j} - p(\mathbf{z_j}))]\}_i \} \\ &\leq \eta_r \sum_{s=1}^{n_b} \sum_{i=1}^{I} (q_{t^*,i} - q_{s,i}) \{ \mathbf{z_s} \sum_{j=1}^{n_1} [\mathbf{z_j}^T (\mathbf{q_j} - p(\mathbf{z_j}))]\}_i \end{aligned} \tag{38}$$

Note that for any vector $\mathbf{x}$, we have $\text{sign}\,(\mathbf{x})\mathbf{x} \geq 0$. And we can divide the right hand side by the left hand side and finish the prove.

$$\alpha \leq \frac{\eta_r \sum_{s=1}^{n_b} \sum_{i=1}^{I} (q_{t^*,i} - q_{s,i}) \{ \mathbf{z_s} \sum_{j=1}^{n_1} [\mathbf{z_j}^T (\mathbf{q_j} - p(\mathbf{z_j}))]\}_i}{\mathbf{b} \{ \eta_r \sum_{s=1}^{n_b} \sum_{i=1}^{I} (q_{t^*,i} - q_{s,i}) \{ \sum_{j=1}^{n_1} [\mathbf{z_j}^T (\mathbf{q_j} - p(\mathbf{z_j}))]\}_i \}} \tag{39}$$

Each term in above can be computed, then we can always find a small enough error range ϵ where surely improve the loss function.

Similarly, for upper bound of $\Delta \max\_loss$, let

$$\begin{aligned} g(\epsilon) &= \Delta \max\_loss \\ &= \eta_r \sum_{s=1}^{n_c} \sum_{i=1}^{I} (q_{t',i} - q_{s,i}) \mathbf{x_s} \sum_{j=1}^{n_1} [\mathbf{z_j}^T (\mathbf{q_j} - p(\mathbf{z_j}))]_i \end{aligned} \tag{40}$$

Note that $g(\epsilon)$ is nothing but a constant with respect to ϵ. This means that the upper bound loss is up to some constant with respect to the recovered trigger z_j.

Note that $\Delta \min\_loss \geq 0$ indicates that the defense is provably effective than without defense. Since benign local clients training can increase global model backdoor loss, and they have positive effects on mitigating malicious poisoning effect. The second condition $\Delta \max\_loss \leq \eta_r \sum_{s=1}^{n_c} \sum_{i=1}^{I} (q_{t',i} - q_{s,i}) \mathbf{x_s} \sum_{j=1}^{n_1} [\mathbf{z_j}^T (\mathbf{q_j} - p(\mathbf{z_j}))]_i$ indicates that the defense is provably guarantee maintaining similar accuracy on clean data.

We now make several remarks about Corollary 1 and will verify them in our experiments section A.5: 1) We establish the connection between adversarial training and loss changes, and we develop upper and lower bounds quantifying the loss changes on backdoored and clean data, on both settings with and without the defense in place. For instance, benign local client training with perfect recovered trigger and assign correct label during adversarial training, it equivalent to benign client is doing exactly opposite training to attackers, this will reduce attacker's backdoor sample confidence, thus we study to validate benign client adversarial training is effective in reducing the attacker confidence during poison training. 2) The confidence threshold τ is a hyper-parameter which can be adjusted to control the attack success rate v.s. accuracy trade-off. For instance, in the rightmost part of Figure 1, the adversarial training already reduces attacker's confidence to a low level. However, without the threshold, the output with the highest probability will still be the target label. Thus we study to validate thresholding is critical during evaluation.

A.4. Methodology

In this section, we detail the design of FLIP, which consists of three main steps as illustrated in Figure 1. The procedure is summarized in Algorithm 1. (1) Trigger inversion. During local client training-time, benign local clients apply trigger inversion techniques to recover the triggers, stamp them on clean images and assign the correct ground truth label to get the augmented dataset. (2) Model hardening. Benign local clients combine the augmented data with the clean data to perform model hardening (adversarial training). The local clients submit updated local model weights to global server and global server will aggregate all the received weights. (3) Low-confidence sample rejection. During global inference, we apply a threshold to filter out samples with low prediction confidence.

Trigger inversion. Backdoor attacks injects hidden malicious behavior to deep learning systems such that any input with a stamped trigger can lead to such behaviors. Readers familiar with trigger inversions can skip this subject. Many existing backdoor defense techniques applying optimization method to invert the smallest input pattern that flip the classification results of the clean images to a target class. In Neural Cleanse [37], the optimization aims to derive a trigger for each class and observe if there is any trigger that is exceptionally small and hence likely injected instead of naturally occurring feature. In our paper, universal trigger generation aims to generate a trigger that can flip samples of all the classes (other than the target class) to the target class.

Class distance. Recent work quantifies model robustness by the class distance [34]. Given some images from source class s, we generate a trigger, composed of a mask m and a pattern δ, which can flip the labels of these images stamped with the trigger to the target class t. The stamping function is illustrated in Equation 41 and the optimization goal in Equation 42, where $\mathcal{L}(\cdot)$ is the cross entropy, M denotes the subject model, and $||\cdot||$ denotes the L^1, i.e. absolute value sum.

$$x'_{s\to t} = (1-m)\cdot x_s + m\cdot\delta \tag{41}$$

$$Loss = \mathcal{L}(M(x'_{s\to t}), y_t) + \alpha\cdot||m|| \tag{42}$$

The class distance $d_{s\to t}$ is measured as $||m||$. The intuition here is that if it's easy to generate a small trigger from source class to target class, the distance between two class is small. Otherwise, the class distance is large. Furthermore, the model is robust if all the class distances are large, or one can easily generate a small trigger between two classes.

Cached Warm-up. Adversarial training on samples with inverted triggers is a widely-used technique for model hardening. Observe that different label pairs have different distance capacities and enlarge label pairs distance by model hardening can improve model robustness and help mitigate backdoors [34]. While existing trigger inversion methods optimize all combination of label pairs without selection will lead to quadratic computation time ($O(n^2)$). In order to reduce the trigger optimization cost, we first generate universal triggers for each label as the target instead of generate triggers for all combination of label pairs, which is linear time complexity ($O(n)$). Then we utilize the universal trigger to approximate the distance from each source class to a target class, and thus we only need to optimize for each target class. During the optimization, FLIP start with a warm-up phase, which sort label pairs based on the distance, and each iteration we prioritize the promising pairs with small distance. Then the trigger optimization only needs to find out the backdoor that can flip most promising label pairs and update the label pairs distance matrix. In order to save computation, each benign local client will maintain a ranking matrix of promising pairs, which is so-called cached warm-up. When the client is selected again, promising pairs can be selected from the matrix and will be updated.

Algorithm 1 FLIP

```
1: Globals input: initial model parameters $w_0$, total training round $R_d$, random select $K$ local clients
2: Local client's input: local dataset $D : \{x, y\}$ and learning rate $\eta$
3: for each training round $r$ in $[1, R_d]$ do
4:     for each client $k$ in $[1, K]$ do
5:         $w^k_{r+1} \leftarrow$ Local_Update $(w_r, D_k)$    ▷ The aggregator sends $w_r$ to Client $k$ who invert triggers based on $w_r$ and its Data $D_k : D : \{x_k, y_k\}$ locally and sends $w^k_{r+1}$ back to the aggregator.
6:     end for
7:     $w_{r+1} \leftarrow \eta_r \sum_{k=1}^{N} w^k_{r+1}$    ▷ Global server aggregating all received weights from different clients.
8: end for
9: Global output: $w_{r+1}$    ▷ Global model after $R_d$ rounds.
10: function LOCAL_UPDATE($w_r, D_k$)
11:     if client $k$ never selected then
12:         for each label existing in client $k$ do
13:             $(d_{1,1}, \cdots, d_{s,t}) \leftarrow L^1$(source, target)    ▷ Store all pair-wise distances to Cache matrix
14:             promising_pairs $\leftarrow$ Cache($k$)    ▷ Select top few promising pairs from cache matrix
15:         end for
16:     else if client $k$ selected before then
17:         promising_pairs $\leftarrow$ Cache($k$)
18:     end if
19:     if promising_pairs exist in dataset then
20:         $x_{adv} \leftarrow$ Symmetric_Train $(w_r, x_k, y_k)$
21:     else
22:         $x_{adv} \leftarrow$ Asymmetric_Train $(w_r, x_k, y_k)$
23:     end if
24:     $w^k_{r+1} \leftarrow$ Adversarial_Train $(\{x_k, x_{adv}\}, \{y_k, y_{adv}\})$    ▷ Adversarial training on clean and augmented data, $y_{adv}$ is the ground truth label of $x_{adv}$.
25:     return $w^k_{r+1}$
26: end function
27: function GLOBAL_MODEL_INFERENCE($w^k_{r+1}, \tau, x, \delta$)    ▷ $\tau$ is the confidence threshold, $x$ is the test sample, $\delta$ is the ground truth trigger
28:     $R_{bd} = \sum \mathbf{1}\{M(x+\delta; w^k_{r+1}) < \tau\}$
29:     $R_{bn} = \sum \mathbf{1}\{M(x; w^k_{r+1}) < \tau\}$
30:     return $R_{bd}, R_{bn}$    ▷ return the number of rejected backdoor samples ($R_{bd}$) and benign samples($R_{bn}$) below $\tau$
31: end function
```

In Algorithm 1, each local client utilize their local samples as the source label and approximate the distance to the target label (measured by L^1, line 10, 11). Then based on the distance enlargement potentiality, each client cache up the ranking matrix of promising pairs (line 12). Given that each local clients are trained for a very few iterations in each round, while the warm-up phase can take up a large portion of training iterations. when the same client is selected again, the cache (store) the ranking matrix can be directly used to select promising label pairs (line 13, 14), which saves training efforts since this client won't need to warm up again. That is, cached warm-up. The cached ranking matrix will be updated and stored locally for every selected round in each client own device. It allows more iterations allocated for model hardening and subsequently, significantly boost model robustness against backdoor attacks.

(A)symmetric Hardening. Given a label pair of *label 1* and *label 2*, there are two directions for trigger inversion, from *1* to *2* and from *2* to *1*. A straightforward idea is to invert from both directions on the same time [34]. However, it is impossible due to the *non-i.i.d* nature of federated learning local client data, since each local client training data can be extremely unbalanced and there may be very few or even no samples for certain labels in a certain local client. During model hardening (adversarial training), a promising class pair will be selected for hardening in each iteration, according to the cached probability matrix mentioned above. We hence separate the model hardening into bidirectional or single directional based on data availability, which accordingly symmetric or asymmetric model hardening. That is, if there exist sufficient data for the two labels of a class pair, symmetric hardening is carried out by generating triggers for the two directions and stamped on the corresponding source labels simultaneously (Algorithm 1 line 15, 16). If there are samples only from one label of a pair, we then only harden the direction from this label to the target (Algorithm 1 line 17, 18). Recovered trigger example as shown in Appendix Figure 2.(c). After local benign clients finishing model hardening, they submit the updated model weights to global server (Algorithm 1 line 19, 20, 5), then global server will aggregate all the received local clients' model weights (Algorithm 1 line 6) and perform the global inference (Algorithm 1 line 7). Such (a)symmetric hardening design allows us to make full use of all the available data in each client.

In Algorithm 2, we present more details about the symmetric and asymmetric inversion. We first initialize the backdoor mask and trigger with the input values. The indicator vector p denotes the direction of symmetric hardening, i.e. 1 denotes label a to b, otherwise, 0 denotes label b to a (line 1-3), which only used in symmetric training. If the client has sufficient data (i.e. more than 5 images for a label) for the both labels of a label pair (a, b), symmetric hardening is carried out by generating triggers for the two directions ($a \rightarrow b$ and $b \rightarrow a$). During the optimization, generated triggers are stamped on the samples of the corresponding label (line 9-11). If the client only has data of one label of a label pair (a, b), we do the asymmetric hardening for one direction from the available label to the target (i.e., $a \rightarrow b$) and stamp the generated triggers on corresponding class samples (line 14, 15).

Low-confidence Sample Rejection. As the benign local clients hardening is contradict to malicious local clients poisoning, in other words, the injected backdoor features in aggregated global model will be mitigated by the benign clients' hardening. Intuitively, benign local clients hardening can *FLIP* adversaries' backdoor prediction confidence from high to low and won't influence much on benign samples. During global server inference, we apply a threshold τ to filter out samples with low prediction confidence after the softmax layer (Algorithm 1 line 22, 23), which significantly improves the model's robustness against backdoor attacks in federated learning. FLIP is compatible with any trigger inversion techniques. In the next section, we theoretically prove that, as long as the reversed trigger satisfies our given bound, then we can guarantee attack success rate will decrease and in the meantime the model can maintain similar accuracy on clean data.

Algorithm 2 Symmetric and Asymmetric Inversion

```
 1: Input: local model parameters $W_r$, local client data $\{\mathbf{x}_k, y\}$
 2: Initialization: $X_n \leftarrow$ a batch of $x \in \mathbf{x}_k$
 3: Initialization: Initialize model $M$ from model weights $W_r$,
    label $(a, b)$, $\mathbf{p}$ is an indicator vector denotes symmetric direction
 4: if $m_{init}$ is not None and $\delta_{init}$ is not None then
 5:     $m, \delta \leftarrow m_{init}, \delta_{init}$
 6: else
 7:     $m, \delta \leftarrow$ random init with shape of $x \in \mathbf{x}_k$
 8: end if
 9: function SYMMETRIC_TRAIN($W_r$, $\mathbf{x}_k$, $y$)
10:     for $step$ in $[0, max\_steps]$ do
11:         $X'_n = \mathbf{p} \cdot \big((1 - m[0]) \cdot X_n + m[0] \cdot \delta[0]\big)$
12:                 $+(1 - \mathbf{p}) \cdot \big((1 - m[1]) \cdot X_n + m[1] \cdot \delta[1]\big)$
13:     end for
14:     return $X'_n$
15: end function
16: function ASYMMETRIC_TRAIN($W_r$, $\mathbf{x}_k$, $y$)
17:     for $step$ in $[0, max\_steps]$ do
18:         $X'_n = (1 - m) \cdot X_n + m \cdot \delta$
19:     end for
20:     return $X'_n$
21: end function
```

A.5. More Experimental Details

A.5.1 Details on Experiment Setup

In this section, we illustrate more details about the experimental setups, neural network structures, parameters setups, etc. For more detailed hyperparameter settings and evaluations, please refer to our code repository, we will release our code upon the paper acceptance.

We train the FL system following our FLIP framework on three datasets: MNIST [15], Fashion-MNIST [41] and CIFAR-10 [14]. MNIST has a training set of 60,000 examples, and a test set of 10,000 examples and 10 classes. Fashion-MNIST consists of a training set of 60,000 examples and a test set of 10,000 examples. Each example is a 28x28 grayscale image, associated with a label from 10 classes. CIFAR-10 is an object recognition dataset with

32x32 colour images in 10 classes. It consists of 60,000 images and is divided into a training set (50000 images) and a test set (10000 images). We split the training data for FL clients in a *non-i.i.d.* manner, by a Dirichlet distribution [26] with hyperparameter α 0.5, following the same setting as [2, 43]. We train the FL global model until convergence and then apply various trigger inversion defense techniques, otherwise the main task accuracy is low and the backdoored model is hard to converge [43]. Note that the confidence threshold τ of FLIP discussed in Methodology section is only used in continuous backdoor attack setting to filter out low-confidence predictions. Based on our empirical study, we typically set $\tau = 0.3$ for simpler datasets, e.g. MNIST and Fashion-MNIST, while $\tau = 0.4$ for more complex datasets, e.g. CIFAR-10. We apply two convolutional layers and two fully connected layers in MNIST and Fashion-MNIST, and Resnet-18 in CIFAR-10 to train our model.

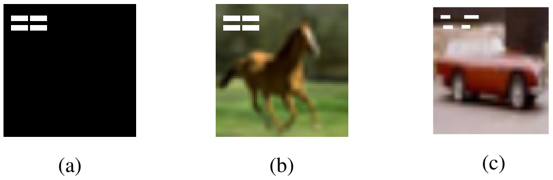
(a) (b) (c)

Figure 2. Trigger Examples, (a) is the ground truth trigger, (b) is malicious client poisoned data, (c) is after benign client trigger inversion, augmented data

Regarding the attack setting, there are 100 clients in total by default. In each round we randomly select 10 clients, including 4 adversaries and 6 benign clients. We do not have any restrictions on attackers as long as they follow the federated learning communication protocol. The attackers inject the pixel-pattern backdoor in images and swap the label of image source label to target label (by default, label ”2”). Figure 2 (b) shows a backdoored example. During testing phase, any inputs with such pattern will be classified as the target label. In single-shot attack, attackers can choose any round to participate. In continuous attack, attackers participate in every round after model convergence. Benign clients perform adversarial training continuously in both settings. We report the ACC and ASR after the attack happens at least 60 rounds, that is, attackers already achieve a high and stable attack success rate.

A.5.2 Evaluation on the Same Setting as Theoretical Analysis

In this section, we conduct an experiment that follows the same setting as our assumptions to validate our theoretical analysis. We conduct experiments on the multi-class logistic regression (one linear layer with softmax function and cross-entropy loss) as the setting in our theory section. We take MNIST as the example for analysis and it can be easily extended to other datasets. Regarding the FL system setting, there are one global server and two local clients consisting of one benign client and one malicious client. We train the FL global model until convergence and then apply the attack. The attackers inject the pixel-pattern backdoor in images and swap the label of the image source label to the target label. We also do not have any restrictions on attackers, as long as attackers follow the federated learning protocol.

Table 5. Logistic regression evaluation

Attack Type	Metric	No Defense	FLIP
Single-Shot	ACC	88.43	84.58
	ASR	64.48	5.28
Continuous	ACC	83.03	80.76
	ASR	63.78	4.90

Table 5 shows the result of single-shot and continuous attack ACC and ASR on the logistic regression, and Table 6 shows detailed number of clean samples and backdoored samples that are predicted correctly. If we use total samples to deduct the predicted correct samples, we can get the rejected samples, which are corresponding to R_{bd} and R_{bn} in theoretical analysis part. We can see both single-shot and continuous attack ASRs are reduced to around 5% and maintain the accuracy drop within an acceptable range. This observation is consistent with our observations on more complex settings above. In addition, we can compute the number of backdoored samples that are rejected (R_{bd}), and the number of benign samples that are rejected (R_{bn}) from Table 6. These sample numbers are corresponding to the number defined in Corollary 1.

Table 6. Logistic regression samples count

Attack Type	Samples count	Total samples	No defense	FLIP
Single-Shot	Clean	10000	8843	8458
	Poisoned	9020	5816	476
Continuous	Clean	10000	8303	8076
	Poisoned	9020	5753	442

A.5.3 Resilience to Adaptive Attacks.

As attackers may work out adaptive attacks to get over FLIP, in this section, we design a countermeasure for attackers and evaluate FLIP under the adaptive attack scenario. Detailed adaptive attack consists of the following steps: (1) attackers apply the same trigger inversion technique as benign clients to obtain the inverted triggers; (2) attackers stamp the inverted triggers to their local images and add them to the training phase for backdoor attacks; (3) attackers submit the updated model weights to global server. We conduct experiments on three datasets under continuous attack setting. Table 7 shows the result, observe that even under an adaptive attack setting, FLIP can still mitigate the backdoor attacks in both MNIST and Fashion-MNIST. In CIFAR-10, the accuracy drops and the adaptive attack is not effective. This indicates that even though the attackers are aware of our technique during poison training, under the FLIP framework, benign clients can still effectively reduce the attacker's poisoning confidence and keep the attack success rate in a low range.

Table 7. Adaptive attacks evaluation

Continuous	**ACC**	**ASR**
MNIST	96.82	0.61
Fashion-MNIST	73.42	18.94
CIFAR-10	60.08	14.02

A.5.4 Area Under the Curve (AUC).

In this section, we take MNIST as an example to show the AUC-ROC curves (Area Under the Curve Receiver Operating Characteristics), other datasets can be analyzed similarly. Figure 3 shows the AUC-ROC curve of our confidence-based sample rejection on MNIST. The curve is plotted with TPR (True Positive Rate) against the FPR (False Positive Rate) where TPR is on the y-axis and FPR is on the x-axis. In our evaluation, the AUC is 0.97. As stated by many existing works [24], an AUC of 0.5 (indicated by the orange dashed line) means the model is unable to discriminate positive and negative samples while an AUC higher than 0.9 is considered outstanding. Therefore our confidence-based rejection strategy is effective in distinguishing backdoored samples and benign samples.

A.5.5 Effect of Adversarial Training.

In this section, we aim to validate that adversarial training in benign local clients indeed can bring positive effects in reducing attackers' poisoning confidence. We conduct the experiments under continuous attacks. We use the same threshold τ and only remove adversarial training at benign local clients and keep all the other settings the same, e.g. confidence threshold.

Table 8. Effect of adversarial training

Continuous	**ACC**	**ASR**
MNIST	96.88	51.75
Fashion-MNIST	79.68	98.53
CIFAR-10	72.90	83.13

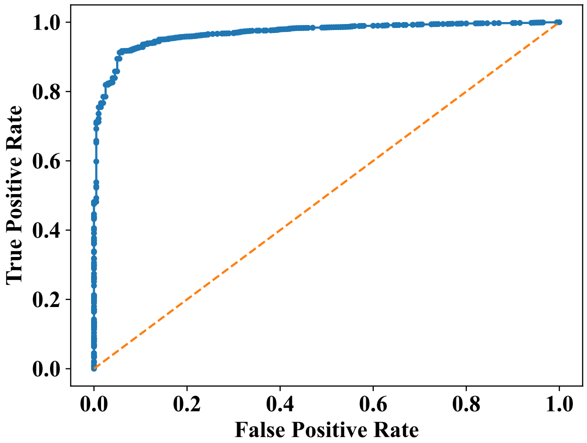

Figure 3. AUC-ROC Curves

In Table 8, we observe that without adversarial training, malicious can successfully inject backdoor patterns even with a high confidence threshold above τ. The underlying reason is that adversarial training in benign clients hardens the model against malicious samples and reduces the confidence of malicious samples. Interestingly, we notice MNIST ASR drops compare with no defenses, the reason could be MNIST dataset feature is simpler, thus with a lot of benign clients continuous training parallelly, it is easy to forget injected backdoor patterns quickly [38], so that attacker's poisoning confidence is reduced and parts samples are rejected. Hence, the results show that adversarial training is significantly effective in reducing the attacker's confidence in backdoor samples during backdoor training, which is consistent with our theoretical analysis.

A.5.6 Effect of Confidence Threshold.

In this section, we demonstrate that threshold is a critical component in FLIP and we evaluate our defense with and without thresholding, results can be found in Table 9. We conduct thresholding experiments under continuous backdoor attacks with three datasets. Each benign client performs trigger inversion and adversarial training as before, while in global inference-time, we set the confidence threshold τ to 0, which is no threshold, and keep all the other settings unchanged. We observe that without thresholding applied, though the ASRs are reduced to some extent, they are still much higher, compared with FLIP results in Table 3. The underlying reason is adversarial training does help in reducing the confidence of backdoored samples, however, without applying confidence threshold to reject the backdoored samples, ASR keeps high. We validate that the threshold is critical in FLIP and the observation is consistent with our results in Corollary 1.

Table 9. Effect of confidence threshold

Continuous	ACC	ASR
MNIST	97.20	22.35
Fashion-MNIST	78.76	30.67
CIFAR-10	75.31	52.47

A.5.7 Other Trigger Inversion Techniques Evaluation.

In general, FLIP is compatible with any trigger inversion technique. In this section, we use another widely-used technique ABS [20] as the trigger inversion component of our framework. Specifically, we replace the "Trigger inversion" part in Figure 1 with ABS, while keeping all other settings the same. We conduct experiments on both single-shot and continuous attack settings. Note that we only evaluate CIFAR-10, since the released version of ABS focuses on the complex dataset with three color channels instead of greyscale images. In each training round of local clients, we use ABS to invert 10 most likely triggers and perform the adversarial training.

Table 10. Other Trigger Inversion Techniques Evaluation

ABS	Single-shot		Continuous	
	ACC	ASR	ACC	ASR
CIFAR-10	74.14	8.00	74.90	22.38

Table 10 shows the defense technique evaluation result, which is consistent with results shown previously in Table 2 and 3. We observe that in continuous attack, FLIP equipped with ABS keeps higher clean accuracy 74% compared to 71% in Table 3 and they both reduce ASR to a low level, near 22%. However, in the single-shot attack, FLIP with ABS only reduces ASR to 8%. The underlying reason is that ABS inverts effective triggers within a small size range, while the method in our main text is more aggressive in hardening the model. The result demonstrates that FLIP is generally effective with various downstream trigger inversion techniques against backdoor attacks.

A.5.8 Impact of Trigger Size.

In this section, we study the different sizes of triggers effect and the evaluation results in Table 11. We define the initial trigger size as X, that is, 2*X denotes the trigger size is scaled up two times compared with the initial trigger. Take MNIST as an example, we observe that the single-shot ASR is low when trigger size (TS) is 1*X, the reason is each local trigger is too small to be recognized during the global model testing phase. We conduct an experiment consisting of different trigger sizes from 1*X, 2*X, 4*X, 6*X, to 8*X. The evaluation shows that our defense can significantly degrade ASR while maintaining comparable benign classification performance, no matter how triggers' sizes change.

Table 11. Trigger Size

TS	No Defense		FLIP	
	ACC	ASR	ACC	ASR
1* X	97.58	1.48	97.09	0.14
2* X	97.57	94.31	96.94	0.29
4* X	97.24	96.41	96.05	0.13
6* X	97.33	97.64	97.23	0.76
8* X	97.46	97.85	96.83	0.45

A.5.9 Impact of Confidence Thresholds

In this section, we show the trade-off between attack success rate and accuracy when we apply the confidence threshold. We conduct an extensive evaluation to study different threshold influences on ACC and ASR. We test our framework on MNIST dataset in the continuous attack setting with three different thresholds 0.0, 0.3, and 0.7. We found that with the increase of confidence threshold, ACC is 97.2%, 96.62%, and 88.86% accordingly, in the meantime, the ASR is 22.35%, 1.93%, and 0.91% accordingly. We observe that benign local model hardening has controllable negative effects on accuracy. Meanwhile, there is a trade-off between adversarial training accuracy and standard accuracy of a model [36]. If we aim for a much lower attack success rate, this will sacrifice part of clean accuracy. In other words, when we set a higher threshold, ASR indeed decreases, in the meantime, some low-confidence benign samples are also rejected, which causes the benign accuracy to reduce to some extent.

A.5.10 Discussion on Other Defenses.

In this section, we provide additional experimental results on the comparison between the Multi-KRUM [3] and our method. We take the CIFAR-10 dataset as an example, in the single-shot attack, Multi-KRUM can drop ASR from 80.46% to 4.18%, and our defense ASR is 7.83%. However, in the continuous attack, Multi-KRUM can only reduce ASR from 84.73% to 61.86%, and our defense ASR is 17.27%, the ACC is at a similar level. Our technique can achieve comparative performance with Multi-KRUM in the single-shot attack and outperforms Multi-KRUM in more complex attack scenarios of continuous attack. In addition, we also try to evaluate FLAME [28]. We contacted the authors of FLAME several times for their experiments and parameters setup but got no response until submission.

A.5.11 Justifications for SOTA Defenses not Working.

In this section, we provide concrete justifications on why SOTA defenses produce a nearly 100% attack success rate on continuous attacks setting. Continuous backdoor attacks denote that in each round the attackers will be selected and continuously participate in federated learning. We suspect there are three reasons that SOTA defenses are performing not well on continuous attacks. First, continuous backdoor attacks are more aggressive. In each round of selected participants, 40% of them are attackers and will participate in every round of model training. Second, as mentioned in [38], even under a very low attack frequency, the attacker still manages to gradually inject the backdoor as long as federated learning runs for long enough. Third, some of their assumptions, e.g. though FoolsGold [10] assumes that benign data are non-iid, meanwhile, it also assumes manipulated data are iid, this could cause FoolsGold to be only effective under certain simpler attack scenarios, e.g. single-shot attacks.

A.6. Conclusion

We presented a new provable defense framework for backdoor mitigation in Federated Learning (FLIP) and a novel trigger inversion technique under FL. The key insight is to combine trigger inversion techniques with FLIP, as long as the reversed trigger satisfies our given bound, then we can guarantee attack success rate will decrease and in the meantime the model can maintain similar accuracy on clean data. Our technique significantly outperforms prior work on the SOTA continuous FL backdoor attack. Our framework is general and can be instantiated with any trigger inversion technique. While applying various trigger inversion techniques, FLIP may have slight accuracy degradation, but it can significantly boost the robustness against backdoor attacks.