ERASER: ERASING TRIGGER-INDUCED EFFECTS FOR BACKDOOR DEFENSE IN FEDERATED LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Federated Learning (FL) is highly vulnerable to backdoor attacks, where malicious clients implant trigger-dependent behaviors into the global model while preserving main-task performance. Existing defenses rely on malicious-client identification or training-time intervention, and often suffer robustness degradation under heterogeneous data, stealthy attacks, and diverse federated settings. We propose Eraser, a post-training backdoor defense for FL that erases trigger-induced effects before inference, preventing triggers from activating backdoor behaviors in a poisoned global model. Since potential triggers are unknown and difficult to remove directly from input samples, Eraser instead reconstructs sanitized samples from robust latent semantics. Motivated by our observation that standalone benign local models exhibit inherent robustness against backdoor triggers, Eraser uses their Featurizers to suppress trigger-related cues in the latent space, while a collaboratively trained Generator reconstructs sanitized samples from these representations. To improve cross-client consistency, we introduce fixed *hyperspherical semantic anchors* to align local feature spaces and stabilize generator training. Eraser is model-agnostic, compatible with most federated algorithms, and requires neither trigger knowledge nor malicious-client identification. Experiments on diverse attacks and heterogeneous settings show that, compared with existing defenses, Eraser achieves the best robustness while preserving main-task accuracy.

1 INTRODUCTION

Federated Learning (FL) (McMahan et al., 2017; Baduwal et al., 2026; Zhang et al., 2026) enables multiple clients to collaboratively train a global model without directly sharing their raw data. By keeping data local while only exchanging model updates, FL offers an attractive balance between utility and privacy, and has been widely adopted in privacy-sensitive domains (Ismail et al., 2026; Bakas et al., 2026; Partohaghighi et al., 2026). Despite these advantages, the decentralized nature of FL also makes it particularly vulnerable to backdoor attacks (Zhang et al., 2022; Yang et al., 2025; Zhao et al., 2026). In such attacks, malicious clients poison their local training process to implant a hidden trigger-dependent behavior into the global model. When the trigger is present, the model is forced to output an attacker-specified prediction; otherwise, it behaves normally on clean samples. This makes backdoor attacks especially dangerous, as they preserve standard performance while remaining dormant until activated by specific triggers (Wang et al., 2026; Li et al., 2026).

To address backdoor attacks in FL, existing defenses can be broadly divided into four categories: Optimization-based defenses (Ozdayi et al., 2021; Xie et al., 2021; Huang et al., 2023b), which suppress malicious influence by modifying training or aggregation; Distance-difference defenses (Huang et al., 2024b; He et al., 2025; Wan et al., 2026), which identify suspicious clients from deviations in their updates; Statistics-distribution defenses (Xu et al., 2025a; Ding et al., 2025a; Yang et al., 2025), which detect malicious clients from statistical patterns; and Post-processing defenses (Wu et al., 2020; Ding et al., 2025b), which adapt centralized post-hoc backdoor mitigation methods to FL after training, but often rely on additional clean data or model-specific repair operations. Despite their effectiveness in some settings, these defenses face **two practical limitations**. First, as illustrated in Fig. 1 (left), advanced backdoor attacks can make malicious updates appear highly benign and stealthy (Zhang et al., 2022; Dai & Li, 2023; Lyu et al., 2023), weakening the effectiveness of existing defenses. Second, many defenses are developed under simplified FL settings and do not readily extend to practical heterogeneous scenarios. As shown in Fig. 1 (right) (a)–(c),

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

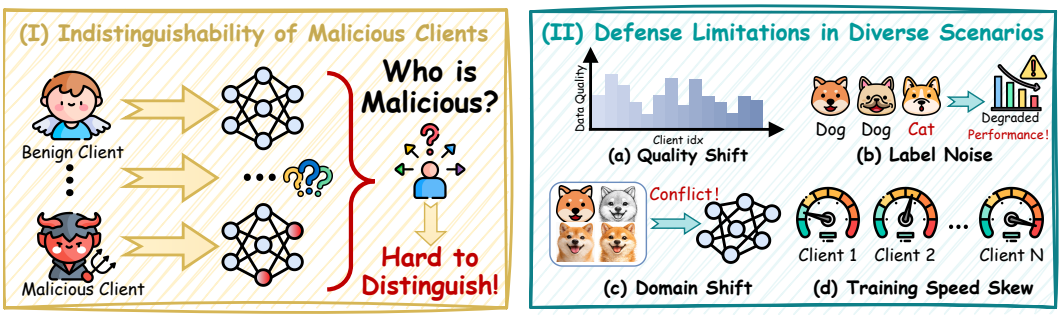

Figure 1: **Problem Illustration.** Existing backdoor defenses in FL suffer from two key challenges. (I) Malicious clients can exhibit updates highly similar to benign ones, making accurate identification difficult. (II) Most defenses are developed under simplified assumptions. As a result, they may not generalize well to heterogeneous deployments, which reduces their effectiveness in practice.

real-world FL often involves quality shift (Wu et al., 2024; 2025), label noise (Xu et al., 2022), and domain shift (Chen et al., 2024; Huang et al., 2023c), which amplify benign update diversity and make malicious updates harder to distinguish. Beyond these data-level challenges, Fig. 1 (right) (d) shows training speed skew (Xu et al., 2023; Wang et al., 2024), which can induce asynchronous training under non-blocking aggregation, where slow clients may submit stale updates from earlier rounds, a setting rarely considered by defenses designed for synchronous FL. Moreover, these scenarios are often handled by tailored FL algorithms that modify the training or aggregation, making direct integration of existing defenses difficult without substantial redesign. These limitations underscore the need for a defense that is both robust and applicable across diverse FL settings.

To address these limitations, we take a different perspective and propose Eraser, a post-training backdoor defense method for FL. Rather than identifying malicious clients or modifying the training and aggregation process, Eraser is applied at **inference time** to erase trigger-induced effects from input samples without altering the trained global model, making it independent of the underlying FL algorithm and applicable to heterogeneous settings. The key challenge is that potential triggers are unknown and difficult to remove directly from the input space. Eraser addresses this challenge by mapping each input into a robust latent semantic space and reconstructing a sanitized sample before it reaches the poisoned global model. Specifically, Eraser consists of a client-specific Featurizer and a collaboratively trained Generator. The Featurizer is motivated by our observation that standalone benign local models exhibit inherent robustness against backdoor triggers, and suppresses trigger-related cues while preserving task-relevant semantics. The Generator then reconstructs samples from these latent representations, preventing potential triggers from being reproduced and activating hidden backdoor behaviors. To improve cross-client consistency, we introduce fixed *hyperspherical semantic anchors* shared by the server, which align local feature spaces and stabilize collaborative Generator training. The main contributions are summarized as:

❶ **Problem Identification.** We identify that existing federated backdoor defenses are primarily designed for simplified federated settings, limiting their effectiveness and applicability under stealthier attacks and more heterogeneous federated settings.

❷ **Novel Solution.** To the best of our knowledge, Eraser is the first post-training sample-purification defense for FL, combining a client-specific Featurizer, a collaboratively trained Generator, and fixed hyperspherical semantic anchors for trigger erasure and feature alignment.

❸ **Experimental Validation.** We evaluate Eraser through extensive experiments on multiple datasets, diverse backdoor attacks, and heterogeneous federated settings, demonstrating the superiority of Eraser against state-of-the-art defense methods.

2 Related Work

2.1 Backdoor Attacks in Federated Learning

The distributed nature of FL makes it vulnerable to backdoor attacks, where malicious clients induce trigger-specific misclassification in the global model. Early attacks mainly focus on trigger design.

BadNet (Gu et al., 2019), Signal (Barni et al., 2019), and Blended (Chen et al., 2017) use pixel, sinusoidal, and blended triggers, respectively, while DBA (Xie et al., 2019) decomposes the trigger into multiple sub-triggers. Later attacks adopt more diverse triggers, including the spatial warping of WaNet (Nguyen & Tran, 2021), imperceptible perturbations of BppAttack (Wang et al., 2022b), and image-wide style transformations of Filter (Liu et al., 2019). However, these attacks are not defense-aware. Recent works therefore develop more adaptive attacks. MRA (Bagdasaryan et al., 2020) amplifies malicious updates to dominate aggregation, while A3FL (Zhang et al., 2023), Neurotoxin (Zhang et al., 2022), and Chameleon (Dai & Li, 2023) improve stealthiness by manipulating update magnitude, direction, or temporal behavior. CerP (Lyu et al., 2023) aligns malicious updates with benign distributions, while 3DFed (Li et al., 2023a) adapts attack behavior based on aggregation feedback. These advanced attacks make federated backdoor defense increasingly challenging.

2.2 Backdoor Defenses in Federated Learning

We broadly categorize existing defenses into four types. **Optimization-based defenses** (Ozdayi et al., 2021; Xie et al., 2021; Huang et al., 2023b) suppress malicious influence by modifying training or aggregation, but often rely on carefully designed objectives and hyperparameters. **Distance-difference defenses** (Blanchard et al., 2017; Fung et al., 2018; Huang et al., 2023a; Qin et al., 2024; Huang et al., 2024b; He et al., 2025; Wan et al., 2026) identify malicious clients from discrepancies among local updates or their deviations from the global update. **Statistics-distribution defenses** (Cao et al., 2020; Nguyen et al., 2022; Xu et al., 2025a;b; Ding et al., 2025a; Yang et al., 2025) detect abnormal clients from statistical patterns using trust scoring, robust aggregation, clustering, or representation modeling. **Post-processing defenses** (Wu et al., 2020; Ding et al., 2025b) adapt centralized post-hoc backdoor mitigation to FL by repairing or modifying the trained global model, typically relying on additional clean data or model-level operations. Overall, existing defenses rely on optimization designs, separability between benign and malicious updates, statistical assumptions, or model-level repair. These dependencies can limit their robustness and compatibility under heterogeneous FL settings. Moreover, many of these defenses intervene directly in optimization or aggregation, making them poorly compatible with tailored FL algorithms for heterogeneous scenarios, which weakens their robustness, practicality, and compatibility in real-world federated systems. Detailed descriptions of the evaluated defenses and attacks are provided in Appendix A.

3 Threat Model

Attack Model. The primary objective of the malicious clients is to implant a backdoor into the global model by uploading malicious updates during federated training. We assume that malicious clients possess strong capabilities to enable realistic and powerful backdoor attacks: they can arbitrarily modify their local training objectives, loss functions, optimization procedures, and attack strategies; they can freely collude with each other and fully share training data, triggers, and updates to evade defense mechanisms; and they face no restriction on their proportion in the system, with the only assumption being that FL still contains at least one benign client. Importantly, malicious clients remain adversarial throughout the entire process and may continue manipulating local optimization and uploaded components during subsequent post-training defense procedures. Similar to prior work (Fung et al., 2018; Huang et al., 2023a; Qin et al., 2024; Huang et al., 2024b; Xu et al., 2025a;b; Ding et al., 2025a; Yang et al., 2025; He et al., 2025; Yang et al., 2025), we do not consider attacks on the communication channel between the server and clients, such as man-in-the-middle attacks (Callegati et al., 2009) and membership inference attacks (Bai et al., 2024).

Defense Goal. Our defense objective is to ensure that normal participants can safely use the final global model, even if the federated training process has already been exposed to backdoor attacks. Accordingly, the effectiveness of the defense is assessed only from the perspective of benign clients, and malicious clients are excluded from evaluation. We measure the defense performance by adopting two key metrics. The first is the main-task accuracy $\mathcal{A}$, which represents the prediction accuracy on clean samples. The second is the backdoor failure rate $\mathcal{B}$, defined as $\mathcal{B} = 1 - \mathcal{ASR}$, where $\mathcal{ASR}$ denotes the attack success rate on the poisoned dataset. Specifically, $\mathcal{B}$ measures the proportion of backdoor samples that are not classified as the attacker's target label y_t. A higher value of $\mathcal{B}$ indicates that the defense suppresses the backdoor behavior. Our objective is to simultaneously achieve high $\mathcal{A}$ and high $\mathcal{B}$. A more detailed description of $\mathcal{A}$ and $\mathcal{B}$ can be found in Appendix B.



Figure 2: Visualization of one clean original image and its variants with different types of triggers.

Table 1: Motivation study on utility, robustness, and clean-triggered feature similarity on CIFAR-10.

Setting	Average $\mathcal{A}\uparrow$	Average $\mathcal{B}\uparrow$	Feature Similarity ↑							
			BadNet	DBA	Signal	Blended	WaNet	BppAttack	1977	Kelvin
Global Model	**86.05**	3.90	67.10	67.83	67.82	66.04	92.37	77.32	79.24	77.80
Standalone Model	73.22	**99.36**	**97.11**	**98.71**	**88.21**	**85.16**	**99.24**	**88.28**	**92.84**	**88.02**

4 METHODOLOGY

4.1 MOTIVATION

Existing federated backdoor defenses typically intervene during global model training by modifying update filtering, aggregation, or optimization, making them setting-specific and hard to generalize across heterogeneous FL scenarios. This motivates defenses that can be applied after training without altering the original FL procedure. Existing post-processing defenses either adapt centralized techniques such as trigger inversion (Wang et al., 2023) and model cleansing (Wang et al., 2019), or repair the trained federated model through pruning or neuron-level weight manipulation (Wu et al., 2020; Ding et al., 2025b). However, these methods operate directly on model parameters and often rely on clean or auxiliary data, which can be limited and heterogeneous in FL.

Motivated by these limitations, we avoid both modifying federated training and repairing the poisoned model, and instead intervene before the backdoor is activated. Since backdoors rely on trigger-induced cues, suppressing such cues before prediction can prevent backdoor activation without modifying the global model. However, directly removing triggers at the pixel level is unreliable because triggers vary widely in pattern, transparency, shape, and location, as shown in Fig. 2. We therefore turn to semantic reconstruction: a Featurizer extracts task-relevant semantics while suppressing trigger-related cues, a Generator reconstructs a sanitized sample from these features, and the global model performs the final prediction. Crucially, the Generator never accesses the raw input and reconstructs the sample solely from the Featurizer output. If the Featurizer maps clean and triggered versions of the same sample to similar semantic representations, the Generator lacks sufficient trigger-specific cues to infer and reconstruct an effective trigger.

To identify a suitable Featurizer, we compare those from the *Global Model* and the *Standalone Model*. The former is obtained by FedAVG under backdoor attacks, while the latter is trained on clean local data without malicious updates or trigger-target supervision. Besides $\mathcal{A}$ and $\mathcal{B}$, we measure the cosine similarity between clean and triggered features of the same sample. As shown in Table 1, the Global Model achieves higher $\mathcal{A}$ but lower $\mathcal{B}$ and feature similarity, indicating trigger-sensitive representations. In contrast, the Standalone Model preserves trigger-insensitive semantics, yielding higher $\mathcal{B}$ and feature similarity despite lower $\mathcal{A}$. Details are provided in Appendix C.

Based on this observation, we propose Eraser, a post-training sample-purification defense that erases trigger-induced effects through semantic reconstruction without modifying the FL procedure or global model. Eraser combines a Featurizer from standalone local training with a lightweight collaboratively trained Generator that maps robust semantic features to the sample space. By reconstructing inputs from trigger-insensitive semantics, Eraser preserves task-relevant cues while suppressing trigger effects before they reach the poisoned global model. Fig. 3 illustrates the framework.

4.2 OVERVIEW OF ERASER

Following the typical FL framework (Huang et al., 2024b; Wan et al., 2026; Yang et al., 2025), we consider a set of clients $\mathcal{S} = \{1, \ldots, N\}$, and denote by $\mathcal{G}$ the global model obtained after federated training. We emphasize that $\mathcal{G}$ may already contain hidden backdoor behaviors, and malicious clients also behave adversarially during Eraser training. The objective of Eraser is not to identify malicious clients, but to learn a reconstruction-based sanitization transform that erases potential trigger effects from sample x before it is input to $\mathcal{G}$. The full procedure contains four stages.

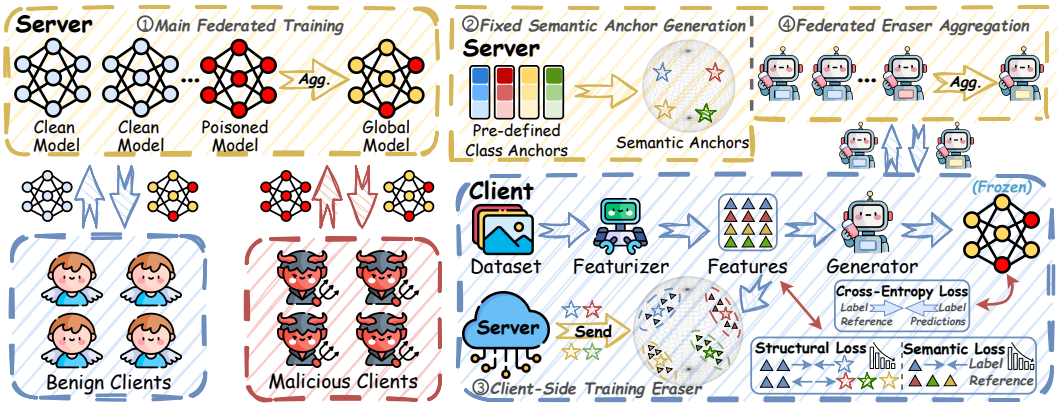

Figure 3: Overview of Eraser. ① Benign and malicious clients jointly train the poisoned global model. ② The server generates fixed semantic anchors. ③ Each client (including malicious clients) locally trains Eraser, which consists of a Featurizer and a Generator. ④ The server aggregates Generators from all clients. Eraser training runs for E rounds by repeating Steps ③ and ④.

① **Main Federated Training.** The server and clients first conduct standard federated training, such as FedAVG (McMahan et al., 2017). At communication round t, the server distributes the current global model $\mathcal{G}^{t-1}$ to the selected clients. Each client i performs local optimization on its dataset D_i and uploads the resulting update to the server, which aggregates all received updates to obtain the next global model $\mathcal{G}^t$. Although $\mathcal{G}$ benefits from collaborative learning and achieves strong main-task utility, it may also inherit hidden backdoor behaviors from malicious updates.

② **Fixed Semantic Anchor Generation.** Heterogeneous client data can make locally trained Featurizers produce inconsistent feature spaces, hindering collaborative Generator training. To provide a shared class-level reference without public data, Eraser introduces fixed *Hyperspherical Semantic Anchors*. Unlike prior anchor-based FL methods (Dong et al., 2022; Zhou et al., 2025) for global model training, Eraser generates anchors once and keeps them fixed throughout. These anchors serve as task-agnostic coordinates that associate each class index with a common feature direction. The server first samples a class-wise anchor matrix from a Gaussian distribution:

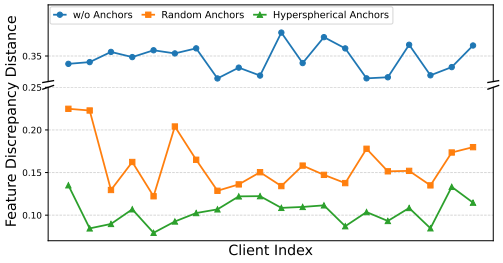

Figure 4: Client-wise feature discrepancy under different anchor designs.

$$\tilde{A} = [\tilde{a}_1, \ldots, \tilde{a}_C]^\top \in \mathbb{R}^{C\times d}, \qquad \tilde{a}_c \sim \mathcal{N}(0, I_d), \tag{1}$$

where C is the number of classes and d is the feature dimension. The anchors are then projected onto the unit hypersphere:

$$A = [a_1, \ldots, a_C]^\top \in \mathbb{R}^{C\times d}, \qquad a_c = \frac{\tilde{a}_c}{\|\tilde{a}_c\|_2}. \tag{2}$$

This projection removes random magnitude bias and makes feature alignment depend on angular separation, yielding more balanced class-wise references. Even if a client lacks some classes, their anchors still act as negative references in the alignment objective. Fig. 4 shows that hyperspherical anchors reduce client-wise feature discrepancy compared with no anchors and unnormalized random anchors, confirming their benefit for cross-client consistency. Details are provided in Appendix D.

③ **Client-Side Eraser Training.** Given the shared *Hyperspherical Semantic Anchors*, the client-side objective is to learn a feature space in which trigger-irrelevant semantic structure is preserved, while the feature that activates hidden backdoor behaviors is suppressed. To this end, each client optimizes its local Eraser, which consists of a client-specific Featurizer F_i and a Generator R_i, on private data under the guidance of the fixed anchor space introduced above. Specifically, client i employs its local Featurizer F_i to map an input sample x into a latent feature:

$$z = F_i(x). \tag{3}$$

Instead of operating in the raw input space, Eraser maps each sample into a semantic space through a local Featurizer, leveraging robust client-specific features while avoiding unknown trigger diversity. The Featurizer is **trained locally from scratch** and does **not participate in federated aggregation**, preventing inherited backdoor features from the potentially poisoned global model. The resulting features mainly preserve class-relevant semantics while suppressing trigger-related cues, and are encouraged to be both *globally aligned across clients* and *locally discriminative within each client*.

To enforce global alignment, we introduce an anchor-guided objective that pulls each feature toward the anchor of its ground-truth class and pushes it away from the anchors of other classes. For a sample $(x, y) \in D_i$, this objective is defined as:

$$\mathcal{L}_{\text{STR}} = -\log \frac{\exp(\text{sim}(F_i(x), a_y)/\tau)}{\sum_{c=1}^{C} \exp(\text{sim}(F_i(x), a_c)/\tau)}, \tag{4}$$

where $\text{sim}(\cdot, \cdot)$ denotes cosine similarity and τ is the temperature parameter. Aligning latent features to shared anchors alleviates cross-client semantic inconsistency under heterogeneous data and provides a more stable feature space for federated Generator training.

However, anchor supervision alone is insufficient. Under heterogeneous data distributions, features may still lack sufficient intra-class compactness and inter-class separability even when they are aligned with their class anchors. To further regularize the local semantic structure, we introduce an auxiliary semantic supervision term. Although the Featurizer contains no classifier, a lightweight linear head is attached only during training for label supervision. The loss is defined as:

$$\mathcal{L}_{\text{SEM}} = \mathcal{L}_{\text{CE}}(H_i(F_i(x)), y), \tag{5}$$

where $H_i(\cdot)$ denotes the auxiliary linear head and $\mathcal{L}_{\text{CE}}$ is the cross-entropy loss. While $\mathcal{L}_{\text{STR}}$ aligns features to a shared global reference, $\mathcal{L}_{\text{SEM}}$ improves local semantic discrimination. Combining the two terms, the local Featurizer F_i is optimized by:

$$\mathcal{L}_{E_i} = \lambda \mathcal{L}_{\text{STR}} + \mathcal{L}_{\text{SEM}}, \tag{6}$$

where λ balances global alignment and local discrimination structure. Given the latent feature z, client i inputs it into its Generator R_i and obtains a reconstructed sample:

$$\tilde{x} = R_i(z). \tag{7}$$

Unlike standard reconstruction learning (Demoment, 2002; Li et al., 2025; Ahmed et al., 2025), Eraser does not aim to reproduce the original input at the pixel level, but instead aims to preserve only the class-discriminative cues required by the global model. Following this principle, the frozen global model $\mathcal{G}$ supervises the Generator R_i through its prediction on $\tilde{x}$, with gradients flowing through $\mathcal{G}$ to update only R_i. The Generator is optimized by:

$$\mathcal{L}_{R_i} = \mathcal{L}_{\text{CE}}\big(\mathcal{G}(R_i(z)), y\big). \tag{8}$$

Notably, we do not assume malicious clients behave honestly during Eraser training. They may manipulate local optimization and upload adversarial Generator updates to poison the aggregated Generator; we adopt this threat model consistently in all subsequent evaluations, with attack procedures detailed in Appendix A.2. Even if the Generator is poisoned, effective backdoor activation remains difficult because it only receives trigger-suppressed latent features from the local Featurizer. Without reliable trigger-specific cues, the poisoned Generator cannot selectively reconstruct effective triggers without also harming clean samples and main-task accuracy.

④ **Federated Generator Aggregation.** Eraser keeps the Featurizer local to preserve the robustness inherited from standalone training, and introduces collaboration only at the reconstruction stage. After local optimization, each selected client i uploads its Generator R_i to the server, which updates the shared Generator by:

$$R^{(e+1)} = \sum_{i \in \mathcal{S}_e} \frac{|D_i|}{\sum_{j \in \mathcal{S}_e} |D_j|} R_i^{(e+1)}, \tag{9}$$

where $\mathcal{S}_e$ denotes the clients participating in Eraser training round e.

Inference Phase. At inference time, each client combines its local Featurizer with the final shared Generator before the global model. Given a sample x, prediction is made by:

$$\hat{y} = \mathcal{G}(R(F_i(x))). \tag{10}$$

Through this pipeline, Eraser erases trigger-induced effects before they reach $\mathcal{G}$, thereby reducing trigger-induced misclassification while preserving the main-task utility of the global model.

4.3 Discussion

Privacy Preservation. Eraser does not require clients to share private samples, features, or labels with the server. The server only receives Generator parameters through federated aggregation, consistent with standard FL communication. Similar to existing defenses (Qin et al., 2024; Huang et al., 2024b; Xu et al., 2025a;b; Ding et al., 2025a; Yang et al., 2025; He et al., 2025), Eraser does not consider parameter leakage attacks (Wang et al., 2022a; Yang et al., 2023; Hu et al., 2023) that recover client data from uploaded parameters, as they fall outside the scope of federated backdoor defense.

Robustness to Residual Trigger Cues. Eraser does not assume that the Featurizer provides a formal guarantee of producing trigger-free features. Instead, it aims to make trigger-related cues insufficient for reconstructing an effective trigger. The local Featurizer avoids aggregation with malicious updates and preserves the robustness inherited from standalone training, while the anchor-guided and semantic losses bias the latent space toward class-level semantics. Since the Generator reconstructs samples solely from Featurizer-produced latent features, it can recover trigger-dependent patterns only if these features contain sufficient trigger-related cues. When such cues are weak or absent, the Generator lacks the necessary input signal to reconstruct an effective trigger. Moreover, if the Generator were to map generic latent perturbations to trigger-like patterns, clean samples with similar feature variations would also be reconstructed with such patterns, substantially degrading main-task accuracy and violating the stealthiness requirement of backdoor attacks (Gu et al., 2019; Zhang et al., 2022; Dai & Li, 2023; Zhang et al., 2023; Lyu et al., 2023; Li et al., 2023a).

5 Experiments

5.1 Experimental Setup

We consider an FL system with 20 clients, of which 8 are malicious. All clients participate in every communication round, and malicious clients are allowed to participate in the global model and the Eraser training phase. SGD is used as the local optimizer, with an initial learning rate of $\eta = 0.03$, a batch size of 64, and one local epoch. We train the global model for $T = 300$ rounds and Eraser for $E = 50$ rounds. **The architecture details of Eraser are given in Appendix E.** Unless otherwise specified, $\tau = 0.1$, $\lambda = 1.0$, and $d = 512$. We conduct all experiments on an NVIDIA 4090 GPU. **Code is available at:** Eraser. *Consistent with all evaluated defense methods, we do not consider device-specific latency or edge-device deployment evaluation, focusing on FL backdoor defense.*

Datasets and Models. We evaluate Eraser on CIFAR-10 (Krizhevsky et al., 2009), CIFAR-100 (Krizhevsky et al., 2009), and PACS (Li et al., 2017). Additional results on Fashion-MNIST (Xiao et al., 2017), CINIC-10 (Darlow et al., 2018), and Tiny-ImageNet (Le et al., 2015) are provided in Appendix F. To simulate non-IID distributions, we use the Dirichlet distribution with sampling parameter α set to 0.5. By default, ResNet-18 (He et al., 2016) is used as the global model.

Evaluated Defense Methods. We compare FedAVG [AISTATS'17] (McMahan et al., 2017) and four categories of backdoor defenses. **Optimization-based defenses**: RLR [AAAI'21] (Ozdayi et al., 2021), CRFL [ICML'21] (Xie et al., 2021), and LockDown [NeurIPS'23] (Huang et al., 2023b). **Distance-difference defenses**: Multi-Krum [NeurIPS'17] (Blanchard et al., 2017), FoolsGold [arXiv'18] (Fung et al., 2018), Multi-Metrics [ICCV'23] (Huang et al., 2023a), SnowBall [AAAI'24] (Qin et al., 2024), FDCR [NeurIPS'24] (Huang et al., 2024b), SPMC [ICML'25] (He et al., 2025), and MARS [NeurIPS'25] (Wan et al., 2026). **Statistics-distribution defenses**: FLTrust [NDSS'21] (Cao et al., 2020), RFA [TSP'22] (Pillutla et al., 2022), FLAME [USENIX Security'22] (Nguyen et al., 2022), AlignIns [CVPR'25] (Xu et al., 2025a), MASA [WACV'25] (Xu et al., 2025b), FedDLAD [IJCAI'25] (Ding et al., 2025a), and FilterFL [CCS'25] (Yang et al., 2025). **Post-processing defenses**: FLAIN [ICMR'25] (Ding et al., 2025b). **Appendix N distinguishes Eraser from prior centralized post-training defenses and compares their performance in FL.**

Evaluated Attack Methods. We evaluate Eraser against four widely used backdoor attacks: BadNet (Gu et al., 2019) as the default attack, DBA (Xie et al., 2019), Signal (Barni et al., 2019), and Blended (Chen et al., 2017). We further examine its robustness against a broader set of triggers and attacks, including WaNet (Nguyen & Tran, 2021), BppAttack (Wang et al., 2022b), 1977 (Liu et al., 2019), Kelvin (Liu et al., 2019), MRA (Bagdasaryan et al., 2020), A3FL (Zhang et al., 2023), Neurotoxin (Zhang et al., 2022), Chameleon (Dai & Li, 2023), CerP (Lyu et al., 2023), and 3DFed (Li et al., 2023a), with the corresponding results reported in Appendix G. To simulate effective backdoor

Table 2: Comparison with state-of-the-art defense methods on CIFAR-10 and CIFAR-100. The best results are shown in **bold**, and the second-best results are underlined.

Baselines	BadNet			DBA			Signal			Blended		
	$\mathcal{A}\uparrow$	$\mathcal{B}\uparrow$	$\mathcal{V}\uparrow$	$\mathcal{A}\uparrow$	$\mathcal{B}\uparrow$	$\mathcal{V}\uparrow$	$\mathcal{A}\uparrow$	$\mathcal{B}\uparrow$	$\mathcal{V}\uparrow$	$\mathcal{A}\uparrow$	$\mathcal{B}\uparrow$	$\mathcal{V}\uparrow$
Dataset: CIFAR-10												
RLR	87.07	0.00	43.54	86.63	7.33	46.98	86.74	0.01	43.38	86.84	0.00	43.42
CRFL	69.61	0.09	34.85	68.02	48.65	58.34	65.96	11.07	38.52	69.17	0.28	34.73
LockDown	49.20	83.84	66.52	47.97	95.18	71.58	52.57	0.00	26.29	53.03	0.00	26.52
Multi-Krum	82.09	0.00	41.05	83.99	34.47	59.23	85.45	0.29	42.87	85.65	0.07	42.86
FoolsGold	83.35	28.10	55.73	83.16	80.23	81.70	82.62	40.80	61.71	83.15	31.39	57.27
Multi-Metrics	77.53	10.19	43.86	77.28	55.77	66.53	80.28	3.75	42.02	76.77	0.08	38.43
SnowBall	26.88	0.00	13.44	13.75	0.00	6.88	50.41	0.37	25.39	48.69	0.49	24.59
FDCR	86.42	0.00	43.21	86.24	13.46	49.85	86.50	0.07	43.29	86.37	0.00	43.19
SPMC	86.73	0.00	43.37	86.44	7.54	46.99	86.57	0.03	43.30	86.79	0.00	43.40
MARS	55.76	96.82	76.29	69.51	98.30	83.91	33.10	0.66	16.88	10.31	0.00	5.16
FLTrust	84.89	0.00	42.45	84.62	7.62	46.12	86.22	2.90	44.56	83.72	0.01	41.87
RFA	84.47	0.00	42.24	83.30	40.81	62.06	83.97	0.30	42.14	84.57	0.04	42.31
FLAME	84.04	0.00	42.02	83.97	13.13	48.55	84.17	0.01	42.09	84.45	0.00	42.23
AlignIns	85.18	96.75	90.97	85.17	98.34	91.76	85.22	99.44	92.33	85.59	99.66	92.63
MASA	82.85	36.94	59.90	82.76	83.62	83.19	82.22	43.30	62.76	82.62	40.88	61.75
FedDLAD	57.74	4.36	31.05	58.63	47.15	52.89	56.57	96.50	76.54	57.01	97.96	77.49
FilterFL	86.65	0.00	43.33	86.50	6.32	46.41	86.54	0.00	43.27	86.80	0.00	43.40
FedAVG	86.23	0.00	43.12	85.57	0.00	42.79	85.51	0.00	42.76	86.06	0.00	43.03
+FLAIN	73.39	61.04	67.22	72.84	60.46	66.65	74.12	1.88	38.00	73.56	0.14	36.85
+Eraser	**87.68**	**98.76**	**93.22**	**87.67**	**98.37**	**93.02**	**87.89**	**99.77**	**93.83**	**87.95**	**99.90**	**93.93**
Dataset: CIFAR-100												
RLR	37.42	0.01	18.72	37.25	1.70	19.48	37.34	1.26	19.30	37.45	0.43	18.94
CRFL	27.31	0.30	13.81	27.71	12.59	20.15	25.29	0.17	12.73	25.29	0.04	12.67
LockDown	1.39	92.03	46.71	1.53	97.68	49.61	0.88	0.10	0.49	1.00	0.08	0.54
Multi-Krum	34.61	0.14	17.38	34.60	1.94	18.27	34.81	1.27	18.04	33.00	0.13	16.57
FoolsGold	1.51	98.75	50.13	1.55	98.86	50.21	0.74	0.00	0.37	0.74	0.00	0.37
Multi-Metrics	32.20	4.21	18.21	31.30	59.87	45.59	28.17	2.53	15.35	32.01	0.03	16.02
SnowBall	1.26	0.00	0.63	0.74	0.00	0.37	0.74	0.00	0.37	2.10	0.00	1.05
FDCR	38.36	0.02	19.19	35.75	0.81	18.28	36.49	0.70	18.60	37.72	0.33	19.03
SPMC	37.90	0.01	18.96	38.13	2.26	20.20	37.45	1.63	19.54	38.08	0.64	19.36
MARS	32.28	93.51	62.90	32.25	99.36	65.81	20.90	0.01	10.46	32.25	80.82	56.54
FLTrust	33.87	0.19	17.03	35.24	12.68	23.96	32.03	11.57	21.80	31.50	9.42	20.46
RFA	34.75	0.06	17.41	36.41	3.82	20.12	34.18	0.42	17.30	34.30	0.13	17.22
FLAME	32.77	0.02	16.40	33.90	1.32	17.61	32.47	0.25	16.36	31.57	0.03	15.80
AlignIns	1.52	0.00	0.76	1.47	0.00	0.74	1.51	0.00	0.76	1.62	0.00	0.81
MASA	1.47	0.38	0.92	1.45	0.36	0.91	1.45	0.19	0.82	1.46	0.20	0.83
FedDLAD	1.07	20.17	10.62	0.96	0.00	0.48	1.13	0.00	0.57	0.19	0.00	0.10
FilterFL	36.96	0.04	18.50	36.88	2.53	19.71	36.99	0.94	18.97	36.42	0.61	18.52
FedAVG	37.97	0.03	19.00	37.73	2.03	19.88	37.52	1.83	19.68	37.93	0.76	19.35
+FLAIN	17.35	0.70	9.03	15.82	0.42	8.12	17.10	0.36	8.73	17.64	0.08	8.86
+Eraser	**38.85**	**99.65**	**69.25**	**38.93**	**99.41**	**69.17**	**38.19**	**96.68**	**67.44**	**38.80**	**96.85**	**67.83**

attacks, each malicious client poisons $p = 50\%$ of its local data and sets the target label to $y_t = 1$. Unless otherwise specified, we set the malicious-client ratio to 40%, and we launch persistent attacks during the final 50% of rounds in both the main FL training and subsequent Eraser training. Appendix H evaluates diverse attack configurations. **To ensure reproducibility, we consider only defense and attack methods with publicly available code.**

Evaluation Metrics. Following prior work (Huang et al., 2023b; 2024a;b; Qin et al., 2024; He et al., 2025; Wan et al., 2026), we adopt the main-task accuracy $\mathcal{A}$ and the backdoor failure rate $\mathcal{B}$. To summarize their overall trade-off, we further introduce the metric $\mathcal{V}$, defined as:

$$\mathcal{V} = \frac{1}{2}(\mathcal{A} + \mathcal{B}). \tag{11}$$

We report the mean performance over the last five communication rounds as the final result.

5.2 Experimental Results

Comparison to State-of-the-Arts. Table 2 compares Eraser with state-of-the-art defenses on CIFAR-10 and CIFAR-100 under four representative backdoor attacks. Existing defenses often struggle to balance utility and robustness, as update-level filtering is sensitive to benign heterogeneity, while training-time interventions may disrupt optimization. On CIFAR-10, Eraser achieves the best performance, with $\mathcal{A}$ of 87.67%$\sim$87.95%, $\mathcal{B}$ of 98.37%$\sim$99.90%, and $\mathcal{V}$ of 93.02%$\sim$93.93%. Compared with AlignIns, Eraser improves $\mathcal{V}$ by 1.26%$\sim$2.25%, while several defenses remain highly attack-sensitive.On CIFAR-100, Eraser maintains $\mathcal{A}$ of 38.19%$\sim$38.93% and $\mathcal{V}$ of 67.44%$\sim$69.25%, improving $\mathcal{V}$ over FedAVG by 47.76%$\sim$50.25%. Notably, FLAIN partially mitigates BadNet and DBA on CIFAR-10 but sacrifices main-task accuracy and nearly fails on Signal and Blended; on CIFAR-100, it reduces $\mathcal{A}$ to 15.82%$\sim$17.64% while keeping $\mathcal{B}$ below 0.70%. Overall, Eraser provides a more stable utility–robustness trade-off across attacks and datasets.

Comparison on Heterogeneous Scenarios. We evaluate defenses across four heterogeneous scenarios together with their corresponding tailored methods: quality shift (FedISM (Wu et al., 2024)), label noise (FedCORR (Xu et al., 2022)), domain shift (FedHEAL (Chen et al., 2024)), and training speed skew (FADAS (Wang et al., 2024)). To assess compatibility, we combine several defenses that

Table 3: Comparison of defenses, tailored methods, and their combinations under heterogeneous scenarios. – denotes failure to converge.

Baselines	Quality Shift *(CIFAR-10)*			Label Noise *(CIFAR-10)*			Domain Shift *(PACS)*			Training Speed Skew *(CIFAR-10)*		
	$\mathcal{A}\uparrow$	$\mathcal{B}\uparrow$	$\mathcal{V}\uparrow$	$\mathcal{A}\uparrow$	$\mathcal{B}\uparrow$	$\mathcal{V}\uparrow$	$\mathcal{A}\uparrow$	$\mathcal{B}\uparrow$	$\mathcal{V}\uparrow$	$\mathcal{A}\uparrow$	$\mathcal{B}\uparrow$	$\mathcal{V}\uparrow$
LockDown	11.74	0.00	5.87	13.03	0.00	6.16	19.00	1.48	10.24	38.12	0.00	19.06
FoolsGold	63.14	18.91	41.03	21.39	0.41	10.90	66.29	74.02	70.15	60.01	11.35	35.68
Multi-Metrics	58.28	0.33	29.30	58.83	0.01	29.42	81.24	0.00	40.62	30.78	0.00	15.39
FDCR	80.36	0.07	40.22	64.27	0.00	32.14	84.83	0.08	42.46	82.37	0.03	41.20
SPMC	81.72	0.09	40.91	54.93	1.78	28.36	84.83	0.00	42.42	80.15	0.02	40.09
MARS	67.12	0.00	33.56	52.46	0.00	26.23	14.18	0.00	7.09	72.20	52.50	62.35
AlignIns	66.41	91.18	78.80	42.90	73.63	58.27	59.24	64.85	62.05	81.05	30.55	55.80
FedDLAD	21.95	2.24	12.09	15.59	5.16	10.37	40.70	44.26	42.48	18.38	0.05	9.21
FilterFL	80.41	0.03	40.22	64.85	0.00	32.43	-	-	-	81.90	0.04	40.97
FedAVG	81.01	0.04	40.53	65.29	0.00	32.65	84.76	0.00	42.38	82.53	0.17	41.35
+FLAIN	62.15	50.31	56.23	24.63	10.01	17.32	41.67	5.96	23.82	71.56	58.45	65.01
+Eraser	83.23	97.95	90.59	51.73	92.90	72.32	85.45	**98.78**	92.12	83.17	97.10	90.14
Tailored Method	84.34	0.05	42.20	78.32	0.00	39.16	87.04	0.24	43.64	85.01	0.08	42.55
+FDCR	82.92	0.00	41.46	77.94	0.00	38.97	86.96	0.00	43.48	84.73	0.70	42.72
+MARS	75.58	0.58	38.08	65.35	0.00	32.68	85.33	0.00	42.67	77.96	68.09	73.03
+AlignIns	79.73	0.14	39.93	59.16	76.62	67.89	85.79	0.05	42.92	83.44	31.04	57.24
+FilterFL	83.25	0.04	41.65	78.46	0.00	39.23	-	-	-	83.39	1.44	42.42
+FLAIN	62.74	51.41	57.08	64.81	52.76	58.79	42.08	4.56	23.32	72.15	63.48	67.82
+Eraser	**84.90**	**98.10**	**91.50**	**79.57**	**96.39**	**87.98**	**88.46**	98.47	**93.47**	**85.62**	**97.40**	**91.51**

Table 4: Runtime overhead compared with FedAVG under CIFAR-10.

Baselines	LockDown	FoolsGold	Multi-Metrics	SnowBall	FDCR	SPMC	MARS	AlignIns	MASA	FedDLAD	FilterFL	FedAVG + FLAIN	FedAVG + ***Eraser***
Runtime × FedAVG	1.13	1.15	1.78	3.74	1.68	1.45	1.66	**1.09**	5.90	1.58	1.37	1.25	1.28

Table 5: Ablation study of the key components in Eraser on CIFAR-10.

Setting	$\mathcal{A}\uparrow$	$\mathcal{B}\uparrow$	$\mathcal{V}\uparrow$
FedAVG + ***Eraser***	**87.68**	**98.76**	**93.22**
w/o HSA	86.43	98.57	92.50
w/o $\mathcal{L}_{STR}$	85.65	98.44	92.05
w/o $\mathcal{L}_{SEM}$	85.06	98.42	91.74

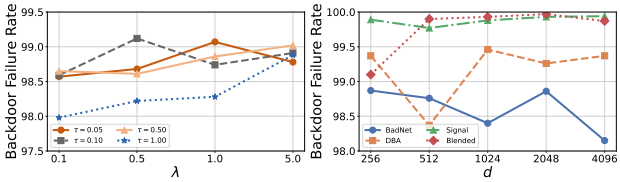

Figure 5: Sensitivity to λ, τ, and d on CIFAR-10.

can identify malicious clients with tailored methods by excluding suspected malicious updates before tailored aggregation. Table 3 shows that existing defenses and their adapted combinations still fail to maintain robustness across heterogeneous scenarios. FLAIN also degrades substantially because its low-activation neuron identification relies on an auxiliary dataset. Under quality shift, label noise, and domain shift, the resulting activation patterns may become less representative, weakening its flipping decisions. By contrast, Eraser consistently achieves strong robustness, with $\mathcal{B}$ ranging from 92.90% to 98.78%. Under label noise, FedAVG+ ***Eraser*** mainly suffers a drop in $\mathcal{A}$ due to corrupted labels, whereas FedCORR+ ***Eraser*** mitigates this degradation through FedCORR's label correction. These results show that Eraser generalizes well to heterogeneous scenarios and remains highly compatible with tailored methods. Detailed scenario descriptions can be found in Appendix I.

Overhead Comparison. Table 4 reports the total runtime overhead relative to FedAVG. Eraser incurs acceptable overhead, with a total runtime of $1.28\times$ FedAVG. For communication, Eraser only transmits the 0.25 MB Generator, which is negligible compared with the 44.63 MB ResNet-18 communicated in FedAVG, resulting in only $1.0009\times$ the total communication cost.

Ablation Study and Sensitivity. We conduct an ablation study to evaluate the contributions of *Hyperspherical Semantic Anchors (HSA)*, $\mathcal{L}_{\text{STR}}$, and $\mathcal{L}_{\text{SEM}}$. As shown in Table 5, each component improves the overall performance. Fig. 5 analyzes the sensitivity of Eraser to λ, τ, and the feature dimension d. Across different settings, $\mathcal{B}$ remains consistently high, indicating that Eraser's defense effect is insensitive to hyperparameters, while increasing d without restoring trigger-related cues.

Additional Results. We provide additional results in the appendix on client scales, newly joined clients, $dir(0.1)$, compatibility, ViT, reconstruction visualizations, pre-processing and centralized defenses, adaptive attack, label-only manipulation attack, anchor designs, text/speech backdoor extensions, pseudocode, complexity, inference overhead, and convergence analysis (Appendices J–T).

6 Conclusion

In this paper, we propose Eraser, a post-training backdoor defense designed for Federated Learning. Existing backdoor defenses either rely on malicious-client identification or directly intervene in training and aggregation, thereby limiting their reliability against attacks and their compatibility with heterogeneous scenarios. To overcome these limitations, Eraser moves the defense stage to inference time and erases trigger effects through sample restructuring, rather than client filtering. Extensive experiments on multiple datasets and attack settings verify the effectiveness of Eraser. We hope this work offers a practical perspective on federated backdoor defense.

REFERENCES

Md Toukir Ahmed, Ocean Monjur, Alin Khaliduzzaman, and Mohammed Kamruzzaman. A comprehensive review of deep learning-based hyperspectral image reconstruction for agri-food quality appraisal. *Artificial Intelligence Review*, 58(4):96, 2025.

Madan Baduwal, Priyanka Paudel, and Vini Chaudhary. Federated learning: A survey of core challenges, current methods, and opportunities. *Computers*, 15(3):155, 2026.

Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In *International conference on artificial intelligence and statistics*, pp. 2938–2948. PMLR, 2020.

Li Bai, Haibo Hu, Qingqing Ye, Haoyang Li, Leixia Wang, and Jianliang Xu. Membership inference attacks and defenses in federated learning: A survey. *ACM Computing Surveys*, 57(4):1–35, 2024.

Spyridon Bakas, Xiaoxiao Li, Prashant Shah, and Holger R Roth. Federated learning in healthcare: From research to real-world deployment. *Annual Review of Biomedical Engineering*, 28, 2026.

Mauro Barni, Kassem Kallas, and Benedetta Tondi. A new backdoor attack in cnns by training set corruption without label poisoning. In *2019 IEEE International Conference on Image Processing (ICIP)*, pp. 101–105. IEEE, 2019.

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.

Hanbo Cai, Pengcheng Zhang, Hai Dong, Yan Xiao, Stefanos Koffas, and Yiming Li. Toward stealthy backdoor attacks against speech recognition via elements of sound. *IEEE Transactions on Information Forensics and Security*, 19:5852–5866, 2024.

Franco Callegati, Walter Cerroni, and Marco Ramilli. Man-in-the-middle attack to the https protocol. *IEEE Security & Privacy*, 7(1):78–81, 2009.

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

Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017.

Yuhang Chen, Wenke Huang, and Mang Ye. Fair federated learning under domain skew with local consistency and domain diversity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12077–12086, 2024.

Yanbo Dai and Songze Li. Chameleon: Adapting to peer images for planting durable backdoors in federated learning. In *International Conference on Machine Learning*, pp. 6712–6725. PMLR, 2023.

Luke N Darlow, Elliot J Crowley, Antreas Antoniou, and Amos J Storkey. Cinic-10 is not imagenet or cifar-10. *arXiv preprint arXiv:1810.03505*, 2018.

Guy Demoment. Image reconstruction and restoration: Overview of common estimation structures and problems. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(12):2024–2036, 2002.

Binbin Ding, Penghui Yang, and Sheng-Jun Huang. Feddlad: A federated learning dual-layer anomaly detection framework for enhancing resilience against backdoor attacks. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI-25*, pp. 5021–5029, 2025a.

Binbin Ding, Penghui Yang, and Sheng-Jun Huang. Flain: Mitigating backdoor attacks in federated learning via flipping weight updates of low-activation input neurons. In *Proceedings of the 2025 International Conference on Multimedia Retrieval*, pp. 219–227, 2025b.

Xin Dong, Sai Qian Zhang, Ang Li, and HT Kung. Spherefed: Hyperspherical federated learning. In *European Conference on Computer Vision*, pp. 165–184. Springer, 2022.

Kealan Dunnett, Reza Arablouei, Volkan Dedeoglu, Dimity Miller, and Raja Jurdak. Backdoor mitigation via invertible pruning masks. *Advances in Neural Information Processing Systems*, 38: 100948–100975, 2026.

Clement Fung, Chris JM Yoon, and Ivan Beschastnikh. Mitigating sybils in federated learning poisoning. *arXiv preprint arXiv:1808.04866*, 2018.

Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 7:47230–47244, 2019.

Kaiqi Guan, Wenke Huang, Xianda Guo, Yueyang Yuan, Bin Yang, and Mang Ye. Rethinking fair federated learning from parameter and client view. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Wenwen He, Wenke Huang, Bin Yang, Shukan Liu, and Mang Ye. Spmc: Self-purifying federated backdoor defense via margin contribution. In *Forty-second International Conference on Machine Learning*, 2025.

Jiahui Hu, Zhibo Wang, Yongsheng Shen, Bohan Lin, Peng Sun, Xiaoyi Pang, Jian Liu, and Kui Ren. Shield against gradient leakage attacks: Adaptive privacy-preserving federated learning. *IEEE/ACM Transactions on Networking*, 32(2):1407–1422, 2023.

Siquan Huang, Yijiang Li, Chong Chen, Leyu Shi, and Ying Gao. Multi-metrics adaptively identifies backdoors in federated learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4652–4662, 2023a.

Tiansheng Huang, Sihao Hu, Ka-Ho Chow, Fatih Ilhan, Selim Tekin, and Ling Liu. Lockdown: backdoor defense for federated learning with isolated subspace training. *Advances in Neural Information Processing Systems*, 36:10876–10896, 2023b.

Wenke Huang, Mang Ye, Zekun Shi, He Li, and Bo Du. Rethinking federated learning with domain shift: A prototype view. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16312–16322. IEEE, 2023c.

Wenke Huang, Mang Ye, Zekun Shi, Bo Du, and Dacheng Tao. Fisher calibration for backdoor-robust heterogeneous federated learning. In *European Conference on Computer Vision*, pp. 247–265. Springer, 2024a.

Wenke Huang, Mang Ye, Zekun Shi, Guancheng Wan, He Li, and Bo Du. Parameter disparities dissection for backdoor defense in heterogeneous federated learning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024b.

Mohanad Ismail, Sherif Fathallah, Ehab M Attia, Mehdi Narimani, and Ryan Ahmed. Self-supervised masked autoencoding meets federated learning for electric vehicle battery state-of-health estimation. *IEEE Transactions on Transportation Electrification*, 2026.

Nazmul Karim, Abdullah Al Arafat, Umar Khalid, Zhishan Guo, and Nazanin Rahnavard. Augmented neural fine-tuning for efficient backdoor purification. In *European Conference on Computer Vision*, pp. 401–418. Springer, 2024.

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

Yann Le, Xuan Yang, et al. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

Congyi Li, Peizhuo Lv, Jinwen He, Kai Chen, and Hong Zhu. Heflguard: Backdoor detection in homomorphic encryption-based federated learning. *IEEE Transactions on Information Forensics and Security*, 2026.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pp. 5542–5550, 2017.

Haoyang Li, Qingqing Ye, Haibo Hu, Jin Li, Leixia Wang, Chengfang Fang, and Jie Shi. 3dfed: Adaptive and extensible framework for covert backdoor attack in federated learning. In *2023 IEEE symposium on security and privacy (SP)*, pp. 1893–1907. IEEE, 2023a.

Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10713–10722, 2021.

Xin Li, Yulin Ren, Xin Jin, Cuiling Lan, Xingrui Wang, Wenjun Zeng, Xinchao Wang, and Zhibo Chen. Diffusion models for image restoration and enhancement: A comprehensive survey. *International Journal of Computer Vision*, 133(11):8078–8108, 2025.

Yige Li, Xixiang Lyu, Xingjun Ma, Nodens Koren, Lingjuan Lyu, Bo Li, and Yu-Gang Jiang. Reconstructive neuron pruning for backdoor defense. In *International Conference on Machine Learning*, pp. 19837–19854. PMLR, 2023b.

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*, pp. 1265–1282, 2019.

Xiaoting Lyu, Yufei Han, Wei Wang, Jingkai Liu, Bin Wang, Jiqiang Liu, and Xiangliang Zhang. Poisoning with cerberus: Stealthy and colluded backdoor attack against federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 9020–9028, 2023.

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*, pp. 1273–1282. PMLR, 2017.

Anh Nguyen and Anh Tran. Wanet–imperceptible warping-based backdoor attack. *arXiv preprint arXiv:2102.10369*, 2021.

Thien Duc Nguyen, Phillip Rieger, Roberta De Viti, Huili Chen, Björn B Brandenburg, Hossein Yalame, Helen Möllering, Hossein Fereidooni, Samuel Marchal, Markus Miettinen, et al. {FLAME}: Taming backdoors in federated learning. In *31st USENIX Security Symposium (USENIX Security 22)*, pp. 1415–1432, 2022.

Mustafa Safa Ozdayi, Murat Kantarcioglu, and Yulia R Gel. Defending against backdoors in federated learning with robust learning rate. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 9268–9276, 2021.

Mohammad Partohaghighi, Roummel Marcia, Bruce J West, and YangQuan Chen. Roughness-informed federated learning for energy consumption modeling of battery electric vehicles. *IEEE Transactions on Transportation Electrification*, 2026.

Krishna Pillutla, Sham M Kakade, and Zaid Harchaoui. Robust aggregation for federated learning. *IEEE Transactions on Signal Processing*, 70:1142–1154, 2022.

Zhen Qin, Feiyi Chen, Chen Zhi, Xueqiang Yan, and Shuiguang Deng. Resisting backdoor attacks in federated learning via bidirectional elections and individual perspective. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 14677–14685, 2024.

Changlong Shi, He Zhao, Bingjie Zhang, Mingyuan Zhou, Dandan Guo, and Yi Chang. Fedawa: adaptive optimization of aggregation weights in federated learning using client vectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 30651–30660, 2025.

Wei Wan, Ning Yuxuan, Zhicong Huang, Cheng Hong, Shengshan Hu, Ziqi Zhou, Yechao Zhang, Tianqing Zhu, Wanlei Zhou, and Leo Yu Zhang. Mars: A malignity-aware backdoor defense in federated learning. *Advances in Neural Information Processing Systems*, 38:154985–155017, 2026.

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)*, pp. 707–723. IEEE, 2019.

Junxiao Wang, Song Guo, Xin Xie, and Heng Qi. Protect privacy from gradient leakage attack in federated learning. In *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*, pp. 580–589. IEEE, 2022a.

Kang Wang, Liangliang Wang, Zhiquan Liu, Yiyuan Luo, Kai Zhang, and Weiwei Li. Shift: Enhancing federated learning robustness through client-side backdoor detection. *Information Fusion*, pp. 104144, 2026.

Yujia Wang, Shiqiang Wang, Songtao Lu, and Jinghui Chen. Fadas: Towards federated adaptive asynchronous optimization. *arXiv preprint arXiv:2407.18365*, 2024.

Zhenting Wang, Juan Zhai, and Shiqing Ma. Bppattack: Stealthy and efficient trojan attacks against deep neural networks via image quantization and contrastive adversarial learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15074–15084, 2022b.

Zhenting Wang, Kai Mei, Juan Zhai, and Shiqing Ma. Unicorn: A unified backdoor trigger inversion framework. *arXiv preprint arXiv:2304.02786*, 2023.

Chen Wu, Xian Yang, Sencun Zhu, and Prasenjit Mitra. Mitigating backdoor attacks in federated learning. *arXiv preprint arXiv:2011.01767*, 2020.

Nannan Wu, Zhuo Kuang, Zengqiang Yan, and Li Yu. From optimization to generalization: fair federated learning against quality shift via inter-client sharpness matching. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pp. 5199–5207, 2024.

Nannan Wu, Zhuo Kuang, Zengqiang Yan, Ping Wang, and Li Yu. Fair federated medical image classification against quality shift via inter-client progressive state matching. *arXiv preprint arXiv:2503.09587*, 2025.

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

Chulin Xie, Keli Huang, Pin-Yu Chen, and Bo Li. Dba: Distributed backdoor attacks against federated learning. In *International conference on learning representations*, 2019.

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

Chenhao Xu, Youyang Qu, Yong Xiang, and Longxiang Gao. Asynchronous federated learning on heterogeneous devices: A survey. *Computer Science Review*, 50:100595, 2023.

Jiahao Xu, Zikai Zhang, and Rui Hu. Detecting backdoor attacks in federated learning via direction alignment inspection. *arXiv preprint arXiv:2503.07978*, 2025a.

Jiahao Xu, Zikai Zhang, and Rui Hu. Identify backdoored model in federated learning via individual unlearning. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 7960–7969. IEEE, 2025b.

Jingyi Xu, Zihan Chen, Tony QS Quek, and Kai Fong Ernest Chong. Fedcorr: Multi-stage federated learning for label noise correction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10184–10193, 2022.

Haomiao Yang, Mengyu Ge, Dongyun Xue, Kunlan Xiang, Hongwei Li, and Rongxing Lu. Gradient leakage attacks in federated learning: Research frontiers, taxonomy, and future directions. *IEEE Network*, 38(2):247–254, 2023.

Yanxin Yang, Ming Hu, Xiaofei Xie, Yue Cao, Pengyu Zhang, Yihao Huang, and Mingsong Chen. Filterfl: Knowledge filtering-based data-free backdoor defense for federated learning. In *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security*, pp. 3147–3161, 2025.

Hangfan Zhang, Jinyuan Jia, Jinghui Chen, Lu Lin, and Dinghao Wu. A3fl: Adversarially adaptive backdoor attacks to federated learning. *Advances in neural information processing systems*, 36: 61213–61233, 2023.

Zhengming Zhang, Ashwinee Panda, Linyue Song, Yaoqing Yang, Michael Mahoney, Prateek Mittal, Ramchandran Kannan, and Joseph Gonzalez. Neurotoxin: Durable backdoors in federated learning. In *International Conference on Machine Learning*, pp. 26429–26446. PMLR, 2022.

Zikai Zhang, Suman Rath, Jiahao Xu, and Tingsong Xiao. Federated learning for smart grid: A survey on applications and potential vulnerabilities. *ACM transactions on cyber-physical systems*, 10(1):1–26, 2026.

Haodong Zhao, Jinming Hu, and Gongshen Liu. Revisiting backdoor threat in federated instruction tuning from a signal aggregation perspective. In *ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2286–2290. IEEE, 2026.

Runkai Zheng, Rongjun Tang, Jianze Li, and Li Liu. Data-free backdoor removal based on channel lipschitzness. In *European Conference on Computer Vision*, pp. 175–191. Springer, 2022.

Yanbing Zhou, Xiangmou Qu, Chenlong You, Jiyang Zhou, Jingyue Tang, Xin Zheng, Chunmao Cai, and Yingbo Wu. Fedsa: A unified representation learning via semantic anchors for prototype-based federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 23009–23017, 2025.

Appendix Organization.

The appendix is organized as follows.

- Appendix A: Evaluated Defenses and Attacks.
- Appendix B: Evaluation Metrics.
- Appendix C: Details of Featurizer Comparison.
- Appendix D: Details of Client-wise Feature Discrepancy.
- Appendix E: Architecture Details of Eraser.
- Appendix F: Performance on More Datasets.
- Appendix G: Performance on Diverse Triggers and Attacks.
- Appendix H: Performance on Different Attack Configurations.
- Appendix I: Description of Heterogeneous Scenarios.
- Appendix J: Performance on Larger Scales, New Clients, and Severe Heterogeneity.
- Appendix K: Performance on Compatibility with Different FL Algorithms.
- Appendix L: Performance on Vision Transformer.
- Appendix M: Visualization of Reconstructed Samples.
- Appendix N: Performance on Pre-processing and Centralized Defenses.
- Appendix O: Performance on Adaptive Attack and Label-Only Manipulation Attack.
- Appendix Q: Performance on Text/Speech Backdoor Extensions.
- Appendix P: Analysis of Different Anchor Designs.
- Appendix R: Algorithm Description.
- Appendix S: Computational Complexity Analysis.
- Appendix T: Convergence Analysis.

A Evaluated Defenses and Attacks

A.1 Defenses

FedAVG (McMahan et al., 2017). FedAVG is the standard aggregation algorithm in federated learning. In each communication round, the server collects local model updates from selected clients and directly averages them to update the global model. Since FedAVG does not include malicious-client detection or robust aggregation, it serves as the non-defense baseline.

RLR (Ozdayi et al., 2021). RLR is an optimization-based defense that adjusts the server-side learning rate according to the sign consistency of client updates. For each parameter, it checks whether the update signs from different clients are sufficiently consistent. If the signs are inconsistent, the server assigns a negative or reduced learning rate to suppress potentially malicious directions during aggregation.

CRFL (Xie et al., 2021). CRFL is a certified robustness-based defense for federated backdoor attacks. It applies model clipping and perturbation during training to bound the influence of malicious updates, and further uses model smoothing to certify that the prediction of the global model remains stable under bounded backdoor perturbations.

LockDown (Huang et al., 2023b). LockDown is an optimization-based defense that mitigates backdoor attacks through isolated subspace training. It restricts local training to selected parameter subspaces and suppresses parameters that are less important for benign clients, thereby reducing the influence of backdoor-related parameters during aggregation.

Multi-Krum (Blanchard et al., 2017). Multi-Krum is a distance-based robust aggregation algorithm. It evaluates each local update by calculating the sum of squared Euclidean distances between the update and its closest $n-f-2$ neighboring updates, where n and f denote the number of participants and the number of attackers, respectively. Updates with smaller scores are considered more reliable and are selected for aggregation.

FoolsGold (Fung et al., 2018). FoolsGold is a similarity-based defense designed to mitigate sybil-style poisoning attacks. It measures the pairwise cosine similarity between client updates, especially

their historical update directions. Clients with highly similar updates are considered more likely to be colluding attackers and are assigned smaller aggregation weights.

Multi-Metrics (Huang et al., 2023a). Multi-Metrics is a distance-difference defense that combines multiple statistical metrics to identify malicious updates. It computes Manhattan distance, Euclidean distance, and cosine similarity between each local update and the global model update. These metrics are then used to form an anomaly score, and updates with abnormal scores are filtered before aggregation.

SnowBall (Qin et al., 2024). SnowBall is a filtering-based defense that selects benign updates through bidirectional elections. In the bottom-up election, each update votes for several similar peer updates, and the most trusted candidates are selected. In the top-down election, these selected updates further expand the benign set by choosing additional reliable updates for aggregation.

FDCR (Huang et al., 2024b). FDCR is a parameter-disparity-based defense for heterogeneous federated learning. It uses Fisher information to estimate the importance of model parameters for local distributions, and then detects malicious updates by analyzing discrepancies on important parameters. After detection, FDCR rescales parameter updates to reduce the influence of suspicious or less important components.

SPMC (He et al., 2025). SPMC is a self-purifying defense based on margin contribution. It estimates the marginal contribution of each client according to the consistency of collaborative training behavior. Clients whose contributions deviate from the collaborative norm are down-weighted, while suspicious gradients are further adjusted through a self-purification process to mitigate local poisoning effects.

MARS (Wan et al., 2026). MARS is a malignity-aware defense that detects backdoor models from the perspective of neuron-level maliciousness. It defines backdoor energy to measure the malicious extent of neurons and extracts prominent backdoor-energy values to form concentrated backdoor energy. Then, it uses Wasserstein-distance-based clustering to distinguish backdoored models from benign ones.

FLTrust (Cao et al., 2020). FLTrust is a trust-bootstrapping defense that relies on a small clean root dataset held by the server. The server trains a reference update on this root dataset and uses its direction as a trusted benchmark. Client updates that are better aligned with the server update receive higher trust scores, while their magnitudes are normalized before weighted aggregation.

RFA (Pillutla et al., 2022). RFA is a robust aggregation method based on the geometric median. Instead of directly averaging all local updates, it computes a geometric-median update that is less sensitive to outliers. Since malicious updates are expected to deviate from the majority of benign updates, RFA reduces their influence during aggregation.

FLAME (Nguyen et al., 2022). FLAME is a statistics-distribution defense that combines clustering, clipping, and noise injection. It first clusters client models to separate suspicious updates from benign ones. Then, it clips the remaining updates and injects a calibrated amount of noise to eliminate backdoor effects while preserving main-task performance.

AlignIns (Xu et al., 2025a). AlignIns is a direction-alignment-based defense that inspects local updates at different granularity levels. It first measures temporal direction alignment between each local update and the global model, and then evaluates fine-grained sign alignment on important parameters using a principal sign reference. Updates with abnormal alignment patterns are filtered out, and the remaining updates are clipped before aggregation.

MASA (Xu et al., 2025b). MASA identifies malicious local models through individual machine unlearning. It observes that backdoored models incur larger empirical loss during unlearning on clean inputs, and uses this abnormal unlearning behavior to filter malicious models. To improve robustness under non-IID data, MASA further applies pre-unlearning model fusion to reduce divergence across clients.

FedDLAD (Ding et al., 2025a). FedDLAD is a dual-layer anomaly detection framework for federated backdoor defense. It first uses a connectivity-based outlier factor module to divide clients into trusted and untrusted groups. Then, an interquartile-range module removes residual malicious clients from the trusted group, while a pardon module reintegrates benign clients that were mistakenly excluded.

FilterFL (Yang et al., 2025). FilterFL is a data-free backdoor defense based on knowledge filtering. It uses two conditional GANs to extract incremental knowledge learned by the current global model and further filter out its backdoor components. If an uploaded model confidently classifies the generated backdoor-knowledge sample into the corresponding target, that model is excluded from aggregation.

FLAIN (Ding et al., 2025b). FLAIN is a post-processing backdoor defense that operates on the trained global model. It is motivated by the observation that backdoor attacks may rely on neurons that remain inactive on clean inputs but respond strongly to malicious inputs. Using an auxiliary dataset, FLAIN identifies low-activation input neurons and iteratively flips their associated weight updates to suppress backdoor-related behavior.

A.2 ATTACKS

To further evaluate Eraser under stronger attack settings, we adapt advanced FL backdoor attacks to the Eraser training stage. These attacks are originally designed to poison the global model during the main federated training process. In our evaluation, malicious clients continue to participate in the subsequent Eraser training and modify their local objectives to preserve trigger-induced effects after reconstruction. Let G denote the poisoned global model, F_i the local Featurizer, and R_i the Generator of client i. For a malicious client with trigger δ and target label y_t, the defense-aware attack objective generally encourages:

$$G(R_i(F_i(x \oplus \delta))) \to y_t, \tag{12}$$

while maintaining normal behavior on clean samples. Thus, malicious clients attempt to upload adversarial Generator updates that make reconstructed triggered samples still activate the backdoor.

MRA (Bagdasaryan et al., 2020). MRA (Model Replacement Attack) amplifies malicious updates so that a small number of attackers can dominate the aggregation result and replace the global model with a backdoored one. To adapt MRA to Eraser, malicious clients apply the same replacement principle to Generator training. Specifically, after optimizing the local Generator with the defense-aware backdoor objective, the malicious Generator update is amplified before being uploaded to the server. This encourages the aggregated Generator to preserve trigger-induced effects after reconstruction.

A3FL (Zhang et al., 2023). A3FL optimizes adaptive triggers by considering the global training dynamics, making the trigger less likely to be forgotten during subsequent FL training. To attack Eraser, we optimize the trigger through the full inference pipeline $G \circ R_i \circ F_i$ rather than only through the poisoned global model. In this way, the trigger is updated to remain effective even after being processed by the Featurizer and reconstructed by the Generator.

Neurotoxin (Zhang et al., 2022). Neurotoxin improves backdoor durability by attacking parameters that change less during training, since these parameters are less likely to be overwritten by subsequent benign updates. To adapt Neurotoxin to Eraser, malicious clients restrict their Generator update to parameters with small historical changes during Eraser training. This aims to implant more persistent trigger-preserving behavior into the aggregated Generator.

Chameleon (Dai & Li, 2023). Chameleon uses contrastive learning to enhance the durability of FL backdoors by strengthening the relationship between triggered samples and the target class. In the Eraser stage, we adapt this contrastive objective to reconstructed samples. Specifically, malicious clients encourage reconstructed triggered samples to be closer to the target-class representations and farther from their original-class representations. This makes the Generator more likely to preserve trigger-induced target behavior after reconstruction.

CerP (Lyu et al., 2023). CerP jointly tunes the backdoor trigger and constrains poisoned model changes to make malicious updates appear similar to benign ones, thereby improving stealthiness against filtering-based defenses. In the Eraser stage, we transfer this idea by jointly optimizing the trigger and the malicious Generator update. Meanwhile, we constrain the Generator update to stay close to benign-looking updates, so that malicious clients attempt to preserve the backdoor effect while reducing the discrepancy of their uploaded Generator parameters.

3DFed (Li et al., 2023a). 3DFed is an adaptive and extensible attack framework that uses constrained backdoor loss, noise masks, and decoy models to disguise poisoned updates. To adapt 3DFed to Eraser, we apply these evasion modules to Generator training. The constrained loss is used to preserve the target prediction after reconstruction, the noise mask is used to hide malicious

changes in the Generator update, and the decoy update is used to make the uploaded Generator appear less suspicious. We further evaluate 3DFed under different learning rates and local training epochs to test whether Eraser remains robust when the attacker strengthens local optimization.

For fair comparison, all evaluated defense and attack methods are implemented following their original designs, and their hyperparameters are set according to the default or recommended settings reported in the corresponding papers.

B Evaluation Metrics

We exclude malicious clients from evaluation to define a cleaner and more meaningful evaluation target. In federated backdoor defense, malicious clients are the source of trigger-target correlations used to poison the global model. Their data and labels may be attacker-controlled, and thus should not be treated as reliable clean semantics whose utility must be preserved. If they were included in clean evaluation, the metric would mix two conflicting objectives: a successful defense should suppress the injected trigger behavior, while the evaluation would also require the model to preserve the attacker-imposed trigger-target mapping. This would make the result misleading rather than fair. Therefore, malicious clients still participate in federated training, but the evaluation is conducted on benign clients, whose data represent legitimate semantics to be preserved. In the following, $\mathcal{D}_{\text{test}}$ denotes the clean test set from benign clients.

Main-task accuracy $\mathcal{A}$ measures the classification accuracy on clean test samples:

$$\mathcal{A} = \frac{1}{|\mathcal{D}_{\text{test}}|} \sum_{(x,y)\in\mathcal{D}_{\text{test}}} \mathbf{1}\left[\mathcal{G}(x) = y\right], \tag{13}$$

where $\mathbf{1}[\cdot]$ is the indicator function.

Backdoor failure rate $\mathcal{B}$ measures the proportion of triggered test samples that are not classified into the attacker-specified target label. Given a trigger δ and target label y_t, the attack success rate is:

$$\mathcal{ASR} = \frac{1}{|\mathcal{D}_{\text{test}}^{\neg y_t}|} \sum_{(x,y)\in\mathcal{D}_{\text{test}}^{\neg y_t}} \mathbf{1}\left[\mathcal{G}(x \oplus \delta) = y_t\right], \tag{14}$$

where $\mathcal{D}_{\text{test}}^{\neg y_t} = \{(x, y) \in \mathcal{D}_{\text{test}} \mid y \neq y_t\}$. The backdoor failure rate is:

$$\mathcal{B} = 1 - \mathcal{ASR}. \tag{15}$$

C Details of Featurizer Comparison

The Global Model is trained using standard FedAVG under default settings. The Standalone Model is trained independently on each benign client using only its local clean data, without federated aggregation or malicious interference. For fairness, both models are trained for 300 rounds.

D Details of Client-wise Feature Discrepancy

Each client independently trains a local model for 100 rounds using its own local data. For each client i and class c, we compute the class-wise mean feature representation:

$$\bar{\mathbf{z}}_{i,c} = \frac{1}{|\mathcal{D}_{i,c}|} \sum_{(x,y)\in\mathcal{D}_{i,c}} F_i(x), \tag{16}$$

where $\mathcal{D}_{i,c}$ denotes the subset of local samples from class c on client i. We measure the discrepancy between two clients i and j for class c using the cosine distance:

$$dis_{i,j}^{c} = 1 - \frac{\bar{\mathbf{z}}_{i,c}^{\top}\bar{\mathbf{z}}_{j,c}}{\|\bar{\mathbf{z}}_{i,c}\|_2\|\bar{\mathbf{z}}_{j,c}\|_2}. \tag{17}$$

The final feature discrepancy is obtained by averaging $dis_{i,j}^{c}$ over all valid client pairs and classes.

Table 6: Architecture of the Generator in Eraser.

Layer	Operation
Input	Latent feature vector
Linear	Fully connected projection
Reshape	Reshape to feature maps
BatchNorm	Batch normalization
Upsample	Spatial upsampling with scale factor 2
Conv + BN + LeakyReLU	3×3 convolution, batch normalization, LeakyReLU
Upsample	Spatial upsampling with scale factor 2
Conv + BN + LeakyReLU	3×3 convolution, batch normalization, LeakyReLU
Conv	3×3 convolution to image channels
Output	Reconstructed image

E Architecture Details of Eraser

We provide the architecture details of Eraser in this section. The Featurizer uses the same architecture as the global model's featurizer. By default, the global model is ResNet-18, and the Featurizer corresponds to its feature extractor. The Generator is implemented as a lightweight upsampling network that first projects the latent feature into feature maps and then progressively upsamples them back to the image space; the architecture is summarized in Table 6.

F Performance on More Datasets

We evaluate Eraser on Fashion-MNIST, CINIC-10, and Tiny-ImageNet to examine its effectiveness across datasets. For Fashion-MNIST, we use a lightweight model consisting of two convolutional layers followed by two fully connected layers. The global model is trained for $T = 100$ rounds, and Eraser is trained for $E = 50$ rounds. For CINIC-10, we follow the same experimental setting as CIFAR-10. For Tiny-ImageNet, which contains more classes and higher-resolution images, we train the global model for $T = 600$ rounds and Eraser for $E = 150$ rounds. Other settings follow the default experimental setup. As shown in Table 7, Eraser consistently achieves strong backdoor suppression while maintaining competitive main-task accuracy across datasets of varying complexity. On Fashion-MNIST, Eraser keeps $\mathcal{B}$ close to 100% across all attacks, while $\mathcal{A}$ is around 86%, indicating that the reconstruction mechanism remains effective even with a lightweight model. On CINIC-10, Eraser improves $\mathcal{B}$ from nearly 0% under FedAVG to over 97% across attacks, while preserving $\mathcal{A}$ around 72%, demonstrating a clear robustness–utility advantage on more diverse natural images. On Tiny-ImageNet, although the larger label space and higher visual complexity make defense more challenging, Eraser still achieves nearly 100% $\mathcal{B}$ while keeping $\mathcal{A}$ around 22%, comparable to FedAVG. These results show that Eraser generalizes well and remains effective as dataset complexity increases.

G Performance on Diverse Triggers and Attacks

We evaluate Eraser on CIFAR-10 under diverse trigger patterns and stronger backdoor attacks. Table 8 shows that Eraser is robust to different trigger forms. Across WaNet, BppAttack, 1977, and Kelvin, Eraser consistently achieves high $\mathcal{B}$ above 99% and $\mathcal{V}$ around 93%, while improving $\mathcal{A}$ to about 87%. In contrast, many existing defenses are sensitive to trigger patterns. For example, FDCR, SPMC, and FilterFL fail to suppress most triggers, with $\mathcal{B}$ often close to zero. MARS performs well on WaNet and BppAttack but collapses on 1977 and Kelvin, indicating clear trigger-specific instability. Although AlignIns also achieves strong robustness, Eraser provides a better overall trade-off by maintaining higher $\mathcal{A}$ and the best $\mathcal{V}$ across all triggers.

Table 9 demonstrates the effectiveness of Eraser against stronger attack strategies. While FedAVG almost completely fails under MRA, CerP, 3DFed, Neurotoxin, and Chameleon, Eraser consistently keeps $\mathcal{B}$ between 98% and 100% with $\mathcal{A}$ around 87%. Existing defenses show unstable performance across attacks: MARS is effective on MRA but fails under Neurotoxin and Chameleon; MASA performs well on Neurotoxin and Chameleon but is weak under MRA, CerP, and 3DFed. In comparison, Eraser achieves the best $\mathcal{V}$ on all six attacks, indicating that semantic reconstruction provides a more stable defense than attack-specific detection or filtering.

Table 7: Comparison with the defenses on Fashion-MNIST, CINIC-10, and Tiny-ImageNet.

Baselines	BadNet			DBA			Signal			Blended		
	$\mathcal{A}\uparrow$	$\mathcal{B}\uparrow$	$\mathcal{V}\uparrow$	$\mathcal{A}\uparrow$	$\mathcal{B}\uparrow$	$\mathcal{V}\uparrow$	$\mathcal{A}\uparrow$	$\mathcal{B}\uparrow$	$\mathcal{V}\uparrow$	$\mathcal{A}\uparrow$	$\mathcal{B}\uparrow$	$\mathcal{V}\uparrow$
Dataset: Fashion-MNIST												
RLR	77.23	0.04	38.64	77.20	0.39	38.80	77.50	0.00	38.75	77.68	0.00	38.84
CRFL	86.10	0.00	43.05	85.25	0.35	42.80	86.61	0.00	43.31	86.57	0.00	43.29
LockDown	52.29	83.39	67.84	54.38	79.46	66.92	53.40	81.28	67.34	52.29	80.17	66.23
Multi-Krum	85.19	99.62	92.41	85.09	99.52	92.31	85.19	0.00	42.60	85.16	0.00	42.58
FoolsGold	67.83	96.41	82.12	68.02	94.83	81.43	67.86	52.15	60.01	67.85	44.13	55.99
Multi-Metrics	85.01	0.00	42.51	81.60	99.47	90.54	83.59	0.00	41.80	82.94	0.00	41.47
SnowBall	79.74	0.00	39.87	81.83	1.15	41.49	78.47	0.28	39.38	80.19	1.52	40.86
FDCR	86.52	0.01	43.27	86.53	0.13	43.33	86.34	0.00	43.17	85.71	0.00	42.86
SPMC	86.55	0.00	43.28	85.60	0.19	42.90	86.52	0.00	43.26	86.72	0.00	43.36
MARS	86.42	66.92	76.67	**86.80**	98.02	92.41	**86.78**	0.00	43.39	86.31	0.00	43.16
FLTrust	85.34	55.82	70.58	83.52	97.99	90.76	85.58	0.00	42.79	85.45	0.00	42.73
RFA	85.38	0.00	42.69	85.61	0.18	42.90	85.32	0.00	42.66	85.36	0.00	42.68
FLAME	85.98	99.72	92.85	85.52	99.11	92.32	85.93	98.75	92.34	85.97	98.95	92.46
AlignIns	67.55	99.04	83.30	67.76	99.16	83.46	67.36	20.02	43.69	67.34	10.94	39.14
MASA	68.74	97.33	83.04	68.54	96.31	82.43	68.64	79.64	74.14	68.70	82.40	75.55
FedDLAD	57.93	95.09	76.51	58.50	96.65	77.58	58.26	46.40	52.33	58.31	28.21	43.26
FilterFL	86.57	0.01	43.29	86.12	0.25	43.19	86.40	0.00	43.20	86.38	0.00	43.19
FedAVG	86.03	0.00	43.02	86.07	0.20	43.14	86.33	0.00	43.17	86.62	0.00	43.31
+FLAIN	81.75	93.42	87.59	82.11	91.86	86.99	80.94	0.26	40.60	81.30	0.17	40.74
+Eraser	**86.65**	**99.73**	**93.19**	86.60	**99.57**	**93.09**	86.70	**99.55**	**93.13**	**86.81**	**99.57**	**93.19**
Dataset: CINIC-10												
RLR	69.31	0.00	34.66	69.25	0.46	34.86	69.96	0.00	34.98	70.12	0.00	35.06
CRFL	62.89	10.86	36.88	64.51	87.72	76.12	65.25	8.94	37.10	63.74	6.97	35.36
LockDown	16.78	0.00	8.39	16.15	0.00	8.08	19.74	0.00	9.87	19.03	0.00	9.52
Multi-Krum	51.90	0.14	26.02	59.70	14.17	36.94	55.14	9.16	32.15	41.12	55.09	48.11
FoolsGold	66.58	49.22	57.90	66.69	87.86	77.28	65.91	58.38	62.15	66.02	58.54	62.28
Multi-Metrics	48.21	97.42	72.82	54.41	97.15	75.78	53.59	17.10	35.35	41.36	94.50	67.93
SnowBall	12.21	0.00	6.11	11.57	0.00	5.79	26.56	82.81	54.69	32.04	68.22	50.13
FDCR	69.49	0.02	34.76	69.56	0.77	35.17	69.52	0.01	34.77	69.98	0.01	35.00
SPMC	69.80	0.01	34.91	70.09	0.65	35.37	70.29	0.01	35.15	70.51	0.00	35.26
MARS	55.44	7.77	31.61	64.68	41.50	53.09	65.14	0.12	32.63	56.65	1.94	29.30
FLTrust	61.69	0.08	30.89	69.80	10.44	40.12	62.00	2.22	32.11	61.36	0.25	30.81
RFA	55.91	0.04	27.98	64.71	1.41	33.06	47.64	7.02	27.33	43.00	27.20	35.10
FLAME	62.46	0.02	31.24	63.76	7.06	35.41	63.90	0.21	32.06	64.28	0.09	32.19
AlignIns	67.35	94.77	81.06	67.33	96.18	81.76	67.55	97.92	82.74	67.19	96.23	81.71
MASA	65.53	40.60	53.07	65.79	85.60	75.70	64.72	38.41	51.57	64.43	37.93	51.18
FedDLAD	44.29	1.58	22.94	49.54	45.26	47.40	54.57	92.10	73.34	54.86	84.82	69.84
FilterFL	69.84	0.00	34.92	70.20	0.51	35.36	70.28	0.01	35.15	69.76	0.00	34.88
FedAVG	69.58	0.01	34.80	69.81	0.47	35.14	70.09	0.01	35.05	70.41	0.00	35.21
+FLAIN	57.30	46.72	52.01	58.72	44.41	56.57	64.38	0.04	32.21	60.76	0.41	30.59
+Eraser	**72.93**	**97.89**	**85.41**	**72.70**	**98.23**	**85.47**	**72.33**	**99.50**	**85.92**	**72.45**	**99.74**	**86.10**
Dataset: Tiny-ImageNet												
RLR	21.89	0.00	10.95	22.07	1.20	11.63	22.25	0.21	11.23	21.71	0.05	10.88
CRFL	14.12	0.06	7.09	12.42	8.76	10.59	13.32	0.01	6.67	12.53	0.02	6.27
LockDown	0.95	99.88	50.41	0.88	99.94	50.41	1.20	8.17	4.68	1.12	2.01	1.57
Multi-Krum	18.15	0.13	9.14	6.35	30.76	18.55	14.82	0.15	7.49	15.40	1.08	8.24
FoolsGold	1.30	96.92	49.11	1.17	99.85	50.51	0.92	0.00	0.46	0.95	0.00	0.47
Multi-Metrics	18.45	1.03	9.74	18.24	13.93	16.08	17.24	0.01	8.62	18.16	0.10	9.13
SnowBall	0.61	0.00	0.30	0.61	0.07	0.34	0.69	**100.00**	50.35	0.48	**100.00**	50.24
FDCR	22.06	0.01	11.03	22.72	1.17	11.95	22.08	0.09	11.09	21.92	0.03	10.97
SPMC	21.94	0.01	10.97	21.91	0.75	11.33	21.95	0.11	11.03	21.18	0.07	10.62
MARS	15.11	99.85	57.48	16.47	99.86	58.16	16.66	**100.00**	57.40	14.16	**100.00**	57.08
FLTrust	21.51	0.64	11.08	21.81	26.56	24.18	20.97	39.14	30.05	21.66	17.23	19.44
RFA	20.60	0.09	10.34	21.01	3.56	12.29	18.23	0.02	9.13	18.44	0.05	9.24
FLAME	16.43	0.01	8.22	16.18	0.74	8.46	16.39	0.02	8.20	16.67	0.01	8.34
AlignIns	1.65	**100.00**	50.82	1.61	**100.00**	50.80	1.71	**100.00**	50.86	1.41	**100.00**	50.71
MASA	1.61	**100.00**	50.80	1.50	99.99	50.74	0.90	0.00	0.45	0.94	0.00	0.47
FedDLAD	0.61	0.00	0.30	0.73	64.96	32.85	0.70	**100.00**	50.35	0.68	**100.00**	50.34
FilterFL	-	-	-	-	-	-	-	-	-	-	-	-
FedAVG	22.04	0.01	11.03	22.11	1.03	11.57	21.77	0.11	10.94	21.86	0.04	10.95
+FLAIN	1.74	99.31	50.53	1.03	99.85	50.44	0.47	99.81	50.14	1.63	99.99	50.81
+Eraser	**22.10**	99.82	**60.96**	**22.79**	99.94	**61.36**	**22.31**	99.72	**61.01**	**22.12**	99.88	**61.00**

Table 8: Comparison with the defenses under different triggers on CIFAR-10.

Baselines	WaNet			BppAttack			1977			Kelvin		
	$\mathcal{A}\uparrow$	$\mathcal{B}\uparrow$	$\mathcal{V}\uparrow$	$\mathcal{A}\uparrow$	$\mathcal{B}\uparrow$	$\mathcal{V}\uparrow$	$\mathcal{A}\uparrow$	$\mathcal{B}\uparrow$	$\mathcal{V}\uparrow$	$\mathcal{A}\uparrow$	$\mathcal{B}\uparrow$	$\mathcal{V}\uparrow$
FoolsGold	86.27	97.07	91.67	85.66	62.27	73.97	85.57	78.07	81.82	85.60	61.10	73.35
FDCR	85.28	32.55	58.92	85.60	1.38	43.49	84.56	0.41	42.49	80.89	0.15	40.52
SPMC	84.98	25.02	55.00	84.88	3.83	44.36	86.99	3.47	45.23	87.07	2.81	44.94
MARS	57.70	98.74	78.22	40.25	98.94	69.60	44.05	1.29	22.67	10.41	0.00	5.21
AlignIns	85.29	99.46	92.38	85.36	99.09	92.23	85.13	99.08	92.11	85.10	99.11	92.11
MASA	86.62	97.73	92.18	85.85	69.29	77.57	85.76	78.71	82.24	85.72	65.94	75.83
FedDLAD	63.92	98.92	81.42	67.78	95.62	81.70	67.57	96.59	82.08	67.87	94.40	81.14
FilterFL	84.53	21.57	53.05	84.65	1.87	43.26	86.58	5.83	46.21	84.40	1.04	42.72
FedAVG	84.66	22.89	53.78	84.87	2.63	43.75	86.92	3.24	45.08	86.81	2.46	44.64
+Eraser	**87.88**	**99.50**	**93.69**	**87.93**	**99.43**	**93.68**	**88.00**	**99.28**	**93.64**	**87.84**	**99.15**	**93.50**

Tables 10 and 11 analyze 3DFed under different attack configurations. When the learning rate varies from 0.01 to 0.10, Eraser maintains high robustness with $\mathcal{B}$ around 98% and achieves the best $\mathcal{V}$ in all settings. Several baselines degrade sharply as the learning rate changes. For instance, MARS is effective only at a learning rate of 0.01 but fails when the learning rate increases, while FoolsGold and MASA show rapidly decreasing robustness. When the number of local epochs varies from 5

Table 9: Comparison with the defenses under different backdoor attacks on CIFAR-10.

Baselines	MRA			A3FL			Neurotoxin			Chameleon			CerP			3DFed		
	$\mathcal{A}\uparrow$	$\mathcal{B}\uparrow$	$\mathcal{V}\uparrow$	$\mathcal{A}\uparrow$	$\mathcal{B}\uparrow$	$\mathcal{V}\uparrow$	$\mathcal{A}\uparrow$	$\mathcal{B}\uparrow$	$\mathcal{V}\uparrow$	$\mathcal{A}\uparrow$	$\mathcal{B}\uparrow$	$\mathcal{V}\uparrow$	$\mathcal{A}\uparrow$	$\mathcal{B}\uparrow$	$\mathcal{V}\uparrow$	$\mathcal{A}\uparrow$	$\mathcal{B}\uparrow$	$\mathcal{V}\uparrow$
FoolsGold	82.45	0.17	41.31	84.15	85.87	76.43	85.69	97.68	91.69	78.76	97.82	88.29	83.83	35.37	59.60	83.42	28.74	56.08
FDCR	86.51	0.00	43.25	86.64	67.41	77.02	85.81	0.96	43.38	79.82	0.00	39.91	86.56	0.00	43.28	86.40	0.00	43.20
SPMC	86.40	0.00	43.20	86.69	67.36	77.02	86.74	0.67	43.70	83.92	0.01	41.97	87.07	0.00	43.53	86.57	0.00	43.29
MARS	71.36	**98.66**	85.01	59.28	90.78	75.03	11.26	0.00	5.63	10.41	0.00	5.21	49.30	78.63	63.97	77.41	73.83	75.62
AlignIns	85.14	96.72	90.30	85.55	88.87	87.21	86.96	97.71	92.33	85.35	98.09	91.72	85.05	89.34	87.19	85.90	98.35	92.13
MASA	81.98	0.02	41.00	83.91	68.28	76.10	83.42	97.37	90.40	76.64	**99.86**	88.25	83.38	9.62	46.50	82.97	33.53	59.93
FedDLAD	54.52	10.77	32.64	57.65	68.57	63.11	67.47	99.07	83.27	71.17	80.14	75.66	57.76	16.97	37.37	57.96	4.42	31.19
FilterFL	86.46	0.00	43.23	86.67	67.62	77.15	86.37	0.53	43.45	84.39	0.00	42.19	87.15	0.00	43.58	85.70	0.00	42.85
FedAVG	86.54	0.00	43.27	86.97	67.38	77.17	86.59	0.55	43.57	84.86	0.01	42.43	87.34	0.00	43.67	85.26	0.00	42.63
+Eraser	**87.08**	98.06	**92.57**	**87.28**	**99.65**	**93.46**	**87.29**	**99.85**	**93.57**	**87.31**	98.19	**92.80**	**87.73**	**98.31**	**93.02**	**87.64**	**98.49**	**93.07**

Table 10: Comparison with the defenses under 3DFed with different learning rates on CIFAR-10.

Baselines	0.01			0.05			0.07			0.10		
	$\mathcal{A}\uparrow$	$\mathcal{B}\uparrow$	$\mathcal{V}\uparrow$	$\mathcal{A}\uparrow$	$\mathcal{B}\uparrow$	$\mathcal{V}\uparrow$	$\mathcal{A}\uparrow$	$\mathcal{B}\uparrow$	$\mathcal{V}\uparrow$	$\mathcal{A}\uparrow$	$\mathcal{B}\uparrow$	$\mathcal{V}\uparrow$
FoolsGold	86.58	58.52	72.55	85.27	11.49	48.38	84.37	4.56	44.47	81.66	1.02	41.34
FDCR	86.96	0.00	43.48	86.63	0.00	43.32	86.08	0.00	43.04	**85.99**	0.00	43.00
SPMC	85.35	0.00	42.68	85.31	0.00	42.66	85.29	0.00	42.65	84.53	0.00	42.27
MARS	77.35	**99.62**	88.49	77.29	0.00	38.65	77.44	0.00	38.72	76.94	0.00	38.47
AlignIns	85.38	98.16	91.77	85.36	98.15	91.76	85.39	98.16	91.78	85.38	98.13	91.76
MASA	86.50	60.09	73.30	85.07	14.06	49.57	84.04	5.47	44.76	81.15	1.56	41.36
FedDLAD	69.16	5.67	37.42	68.96	11.35	40.16	70.04	15.10	42.57	68.00	14.82	41.41
FilterFL	85.63	0.00	42.82	85.17	0.00	42.59	50.01	0.00	25.01	22.40	0.00	11.20
FedAVG	86.92	0.00	43.46	86.37	0.00	43.19	85.87	0.00	42.94	81.12	0.00	40.56
+Eraser	**87.45**	98.25	**92.85**	**88.05**	**98.33**	**93.19**	**87.70**	**98.49**	**93.10**	84.94	**98.82**	**91.88**

Table 11: Comparison with the defenses under 3DFed with different local epochs on CIFAR-10.

Baselines	5			10			15			20		
	$\mathcal{A}\uparrow$	$\mathcal{B}\uparrow$	$\mathcal{V}\uparrow$	$\mathcal{A}\uparrow$	$\mathcal{B}\uparrow$	$\mathcal{V}\uparrow$	$\mathcal{A}\uparrow$	$\mathcal{B}\uparrow$	$\mathcal{V}\uparrow$	$\mathcal{A}\uparrow$	$\mathcal{B}\uparrow$	$\mathcal{V}\uparrow$
FoolsGold	86.21	20.17	53.19	86.01	19.29	52.65	85.95	19.07	52.51	85.63	16.59	51.11
FDCR	86.28	0.00	43.14	85.23	0.00	42.62	85.90	0.00	42.95	85.32	0.00	42.66
SPMC	86.47	0.00	43.24	86.34	0.00	43.17	85.48	0.00	42.74	85.52	0.00	42.76
MARS	70.00	0.04	35.02	29.00	76.56	52.78	73.84	0.00	36.92	61.30	0.00	30.65
AlignIns	87.00	97.74	92.37	86.73	97.50	92.12	86.70	97.48	92.09	86.80	97.46	92.13
MASA	85.88	21.15	53.52	85.52	17.66	51.59	85.64	15.08	50.36	84.04	14.51	49.28
FedDLAD	67.52	17.98	42.75	67.87	19.83	43.85	67.82	19.70	43.76	67.45	20.53	43.99
FilterFL	68.37	0.00	34.19	86.82	0.00	43.41	86.35	0.00	43.18	86.93	0.00	43.47
FedAVG	86.63	0.00	43.32	86.66	0.00	43.33	86.81	0.00	43.41	85.80	0.00	42.90
+Eraser	**87.27**	**98.91**	**93.09**	**87.10**	**97.86**	**92.48**	**87.22**	**97.58**	**92.40**	**87.57**	**97.64**	**92.61**

to 20, Eraser remains stable, keeping $\mathcal{B}$ around 98% and $\mathcal{V}$ above 92%. These results show that Eraser is not only robust to diverse triggers and advanced attacks, but also stable when the attack is strengthened through different local optimization settings.

H Performance on Different Attack Configurations

Comparison on Different Malicious-client Ratios. We evaluate defenses under different malicious-client ratios, including 0%, 20%, 60%, and 80%. Table 12 shows that increasing the malicious-client ratio significantly weakens most existing defenses. FoolsGold and MASA provide moderate robustness when the malicious-client ratio is low, but their $\mathcal{B}$ drops sharply as the ratio increases. FDCR, SPMC, and FilterFL fail under most attacked settings, with $\mathcal{B}$ close to zero. AlignIns is effective when the malicious-client ratio is 20%, but its robustness degrades substantially under higher malicious-client ratios. FedDLAD performs well on some attacks, such as Signal and Blended, but suffers from a large drop in $\mathcal{A}$. By contrast, Eraser remains stable across all malicious-client ratios and attacks. Even when the malicious-client ratio reaches 80%, Eraser still maintains $\mathcal{B}$ above 98% and achieves the best $\mathcal{V}$ in all cases. These results show that Eraser is robust even when malicious clients dominate the federation.

Comparison on Different Attack Start Ratios. We evaluate attacks that start at different training progress ratios, including 10%, 30%, 70%, and 90%. Table 13 shows that many defenses are sensitive to when the attack begins. For early attacks, FedAVG, FDCR, SPMC, and FilterFL usually fail to recover robustness, while FoolsGold and MASA provide limited protection. When attacks start later, some defenses achieve higher $\mathcal{B}$ because the global model has already learned stronger clean semantics before being poisoned. However, their performance remains unstable across attack types. For example, MARS performs well in some late-start settings but collapses in several early-start settings, and FedDLAD improves robustness on Signal and Blended but keeps much lower $\mathcal{A}$.

Table 12: Comparison with the defenses under different malicious-client ratios on CIFAR-10.

Baselines	0%			20%			60%			80%		
	$\mathcal{A}\uparrow$	$\mathcal{B}\uparrow$	$\mathcal{V}\uparrow$	$\mathcal{A}\uparrow$	$\mathcal{B}\uparrow$	$\mathcal{V}\uparrow$	$\mathcal{A}\uparrow$	$\mathcal{B}\uparrow$	$\mathcal{V}\uparrow$	$\mathcal{A}\uparrow$	$\mathcal{B}\uparrow$	$\mathcal{V}\uparrow$
Attack: BadNet												
FoolsGold	84.33	96.49	90.41	83.66	63.29	73.47	82.48	10.80	46.64	83.22	2.89	43.05
FDCR	86.30	98.57	92.43	85.88	0.00	42.94	85.36	0.00	42.68	86.30	0.00	43.15
SPMC	85.94	97.80	91.87	86.33	0.00	43.16	86.64	0.00	43.32	86.40	0.00	43.20
MARS	80.54	98.34	89.44	67.00	93.47	80.23	60.65	41.89	51.27	45.08	0.00	22.54
AlignIns	83.68	96.91	90.30	84.24	96.59	90.41	81.26	3.27	42.26	82.74	0.85	41.79
MASA	84.24	96.58	90.41	83.80	72.64	78.22	82.37	14.43	48.40	83.61	3.98	43.79
FedDLAD	56.83	97.88	77.36	57.86	95.51	76.68	53.95	0.00	26.97	51.62	0.00	25.81
FilterFL	85.78	97.87	91.83	86.14	0.00	43.07	85.58	0.00	42.79	86.32	0.00	43.16
FedAVG	85.88	97.91	91.90	86.15	0.00	43.07	86.00	0.00	43.43	86.30	0.00	43.15
+Eraser	**86.69**	**98.93**	**92.81**	**87.30**	**98.97**	**93.14**	**86.85**	**98.85**	**92.85**	**88.61**	**99.76**	**94.19**
Attack: DBA												
FoolsGold	84.33	98.02	91.18	83.62	93.41	88.52	81.94	48.87	65.41	83.37	26.33	54.85
FDCR	86.15	99.03	92.59	86.24	6.02	46.13	85.96	0.11	43.03	86.05	0.00	43.03
SPMC	86.26	98.79	92.52	86.15	5.79	45.97	86.25	0.08	43.16	86.07	0.00	43.04
MARS	70.30	98.95	84.62	17.50	0.00	8.75	15.59	0.00	7.80	46.61	10.11	28.36
AlignIns	83.65	98.21	90.93	84.33	98.07	91.20	81.52	24.65	53.09	82.82	17.54	50.18
MASA	84.29	98.04	91.16	83.79	94.65	89.22	82.15	54.85	68.50	83.25	28.36	55.81
FedDLAD	57.03	99.23	78.13	57.92	98.03	77.98	53.58	2.30	27.94	50.43	0.09	25.26
FilterFL	86.12	98.53	92.32	85.82	5.34	45.58	86.00	0.04	43.02	86.20	0.00	43.10
FedAVG	86.04	98.89	92.46	85.92	5.35	45.64	86.03	0.04	43.04	86.72	0.00	43.36
+Eraser	**86.35**	**99.39**	**92.87**	**87.11**	**98.44**	**92.78**	**87.25**	**98.81**	**93.03**	**86.90**	**98.64**	**92.77**
Attack: Signal												
FoolsGold	84.36	99.95	92.15	83.77	88.52	86.15	81.82	10.55	46.18	82.42	2.39	42.40
FDCR	86.04	99.88	92.96	85.80	0.25	43.02	85.04	0.00	42.52	86.56	0.00	43.28
SPMC	85.91	99.79	92.85	86.05	0.25	43.15	81.18	0.00	40.59	87.14	0.00	43.57
MARS	77.32	**100.00**	88.66	9.32	0.00	4.66	14.84	0.00	7.42	37.70	0.00	18.85
AlignIns	83.67	99.92	91.79	83.14	99.91	91.53	79.95	1.61	40.78	81.36	0.80	41.08
MASA	84.20	99.93	92.06	83.68	90.24	86.96	82.12	10.88	46.50	83.02	2.56	42.79
FedDLAD	56.81	**100.00**	78.40	58.15	**100.00**	79.07	54.44	84.95	69.70	55.80	74.19	64.99
FilterFL	86.18	99.77	92.97	85.81	0.60	43.20	86.10	0.00	43.05	85.79	0.00	42.89
FedAVG	85.89	99.76	92.83	85.99	0.18	43.09	79.98	0.00	39.99	86.49	0.00	43.24
+Eraser	**86.41**	99.85	**93.13**	**87.16**	99.85	**93.50**	**86.42**	**99.08**	**92.75**	**88.52**	**99.16**	**93.84**
Attack: Blended												
FoolsGold	84.36	99.98	92.17	83.96	88.03	86.00	82.25	4.88	43.56	82.78	0.57	41.68
FDCR	85.99	99.99	92.99	86.11	0.00	43.06	85.74	0.00	42.87	86.11	0.00	43.05
SPMC	85.89	99.93	92.91	86.30	0.00	43.15	83.95	0.00	41.97	86.96	0.00	43.48
MARS	75.97	**100.00**	87.98	16.59	0.00	58.29	19.97	0.00	59.99	32.49	0.00	16.25
AlignIns	83.67	99.98	91.82	83.38	99.99	91.69	80.29	0.41	40.35	81.84	0.21	41.03
MASA	84.23	99.97	92.10	83.88	92.16	88.02	82.79	6.36	44.58	83.04	0.99	42.01
FedDLAD	56.81	**100.00**	78.40	58.75	**100.00**	79.37	54.90	86.66	70.78	56.48	76.17	66.32
FilterFL	86.30	99.93	93.12	86.37	0.01	43.19	86.43	0.00	43.21	86.71	0.00	43.35
FedAVG	86.01	99.93	92.97	86.24	0.00	43.12	84.71	0.00	41.35	86.93	0.00	43.47
+Eraser	**86.85**	99.85	**93.35**	**86.79**	99.67	**93.23**	**86.57**	**98.44**	**92.51**	**88.08**	**99.11**	**93.60**

Eraser consistently achieves high robustness regardless of the attack start ratio. Across all attacks and start ratios, Eraser keeps $\mathcal{B}$ around 98%–99% and obtains the best $\mathcal{V}$ in every setting. These results indicate that Eraser does not rely on a specific poisoning schedule and can mitigate both early and late backdoor injections.

Comparison on Different Poison Ratios. Table 14 reports the results under different attack poison ratios. As the poison ratio changes, the strength and stealthiness of malicious updates also change, which makes detection-based defenses unstable. FDCR, SPMC, FilterFL, and FedAVG still fail to suppress most attacks. FoolsGold and MASA are effective only in limited cases, while MARS shows large fluctuations across poison ratios and attack types. AlignIns maintains strong robustness, but its $\mathcal{A}$ and $\mathcal{V}$ are consistently lower than those of Eraser.

Eraser achieves the best overall trade-off under all poison ratios. Across BadNet, DBA, Signal, and Blended, Eraser maintains $\mathcal{B}$ above 98% in nearly all settings and keeps $\mathcal{V}$ above 92%. These results demonstrate that Eraser remains robust when the attacker changes the amount of poisoned local data.

Overall, the results in Tables 12, 13, and 14 show that Eraser is stable under different attack configurations. Unlike malicious-client detection defenses, Eraser does not depend on the attack ratio, attack timing, or poison ratio to separate benign and malicious updates before aggregation. By removing trigger-induced effects through semantic reconstruction, Eraser provides consistent robustness across diverse attack settings.

Table 13: Comparison with the defenses under different attack start ratios on CIFAR-10.

Baselines	10% $\mathcal{A}\uparrow$	10% $\mathcal{B}\uparrow$	10% $\mathcal{V}\uparrow$	30% $\mathcal{A}\uparrow$	30% $\mathcal{B}\uparrow$	30% $\mathcal{V}\uparrow$	70% $\mathcal{A}\uparrow$	70% $\mathcal{B}\uparrow$	70% $\mathcal{V}\uparrow$	90% $\mathcal{A}\uparrow$	90% $\mathcal{B}\uparrow$	90% $\mathcal{V}\uparrow$
Attack: BadNet												
FoolsGold	83.91	4.85	44.38	83.52	13.72	48.62	83.33	54.35	68.84	84.01	88.19	86.10
FDCR	85.58	0.00	42.79	86.41	0.00	43.21	86.35	0.00	43.17	86.18	0.00	43.09
SPMC	85.92	0.00	42.96	86.51	0.00	43.26	86.93	0.00	43.46	86.75	0.00	43.37
MARS	42.55	89.42	65.99	72.39	8.89	40.64	52.09	97.98	75.04	71.97	97.82	84.90
AlignIns	85.17	96.79	90.98	85.20	96.73	90.97	84.94	96.73	90.84	84.45	96.66	90.56
MASA	83.15	7.23	45.19	82.87	20.20	51.54	82.99	61.37	72.18	83.75	89.33	86.54
FedDLAD	56.23	0.11	28.17	57.14	0.75	28.94	58.27	25.21	41.74	57.19	80.28	68.73
FilterFL	86.10	0.00	43.05	86.53	0.00	43.27	86.80	0.00	43.40	86.92	0.00	43.46
FedAVG	86.02	0.00	43.01	86.64	0.00	43.32	86.77	0.00	43.38	86.81	0.00	43.41
+Eraser	**87.41**	**98.52**	**92.96**	**87.74**	**98.66**	**93.20**	**87.67**	**98.46**	**93.07**	**87.55**	**98.70**	**93.13**
Attack: DBA												
FoolsGold	83.53	54.30	68.92	83.18	67.44	75.31	83.30	91.26	87.28	84.12	96.66	90.39
FDCR	85.51	32.25	58.88	86.32	18.11	52.22	86.38	12.38	49.38	86.41	10.44	48.43
SPMC	85.49	18.33	51.91	86.64	7.22	46.93	86.80	6.76	46.78	86.23	7.15	46.69
MARS	55.75	82.91	69.33	68.02	**98.58**	83.30	67.27	96.95	82.11	76.00	98.02	87.01
AlignIns	85.27	**98.46**	91.87	85.18	98.45	91.82	84.93	**98.23**	91.58	84.50	98.22	91.36
MASA	82.74	57.74	70.24	82.68	71.51	77.10	83.04	92.19	87.62	84.05	96.79	90.42
FedDLAD	57.13	9.46	33.30	57.99	23.78	40.89	58.86	77.03	67.95	57.64	95.33	76.49
FilterFL	85.55	16.36	50.96	86.67	6.64	46.66	86.68	5.58	46.13	86.68	4.90	45.79
FedAVG	85.69	18.95	52.32	86.21	9.35	47.78	86.59	5.65	46.12	86.40	6.04	46.22
+Eraser	**87.53**	98.36	**92.95**	**87.44**	98.53	**92.99**	**87.28**	98.21	**92.75**	**86.85**	**98.38**	**92.62**
Attack: Signal												
FoolsGold	82.70	12.99	47.84	82.61	24.04	53.33	83.03	70.37	76.70	83.91	97.55	90.73
FDCR	85.82	0.00	42.91	86.29	0.04	43.17	86.32	0.11	43.22	85.99	0.08	43.03
SPMC	85.97	0.00	42.99	86.25	0.00	43.12	86.88	0.03	43.45	86.82	0.26	43.54
MARS	14.17	0.00	7.09	12.93	0.00	6.47	72.67	99.91	86.29	23.22	98.17	60.69
AlignIns	84.72	**99.97**	92.35	85.05	**99.97**	92.51	85.81	**99.97**	92.89	85.15	99.92	92.54
MASA	82.09	13.63	47.86	82.02	25.77	53.90	82.66	71.65	77.15	83.87	97.41	90.64
FedDLAD	55.62	77.02	66.32	56.17	91.15	73.66	56.46	99.21	77.84	55.71	**100.00**	77.85
FilterFL	85.84	0.00	42.92	86.26	0.01	43.14	86.80	0.08	43.44	86.41	0.22	43.32
FedAVG	86.21	0.00	43.10	86.50	0.00	43.25	86.96	0.07	43.51	86.61	0.25	43.43
+Eraser	**86.81**	99.52	**93.16**	**87.32**	99.71	**93.52**	**87.89**	99.76	**93.82**	**87.64**	99.46	**93.55**
Attack: Blended												
FoolsGold	83.17	7.29	45.23	83.12	15.50	49.31	83.32	67.04	75.18	83.96	98.49	91.23
FDCR	86.07	0.00	43.03	86.13	0.00	43.06	86.62	0.01	43.31	86.29	0.02	43.15
SPMC	86.16	0.00	43.08	86.48	0.00	43.24	86.73	0.00	43.37	86.90	0.03	43.47
MARS	21.03	0.00	10.51	14.68	7.82	11.25	52.67	62.28	57.47	73.90	98.91	86.40
AlignIns	84.42	**100.00**	92.21	85.15	**100.00**	92.58	86.29	97.98	92.13	85.29	99.98	92.64
MASA	82.28	10.31	46.29	82.40	23.06	52.73	83.20	74.96	79.08	83.93	98.49	91.21
FedDLAD	56.83	82.71	69.77	57.09	93.63	75.36	56.93	**99.78**	78.36	56.27	**100.00**	78.13
FilterFL	86.11	0.00	43.06	86.61	0.00	43.31	86.85	0.00	43.42	86.64	0.02	43.33
FedAVG	86.06	0.00	43.03	86.30	0.00	43.15	86.53	0.00	43.27	86.69	0.03	43.36
+Eraser	**86.90**	99.85	**93.37**	**87.29**	99.79	**93.54**	**86.94**	98.91	**92.93**	**87.45**	99.23	**93.34**

I Description of Heterogeneous Scenarios

Quality Shift. Quality shift models heterogeneity caused by unequal input quality across clients. For each client, we randomly select a fraction of local samples from $U(0, 0.6)$ and corrupt them with Gaussian noise with zero mean and standard deviation 0.1. As a result, different clients observe data with varying degrees of visual degradation, leading to greater divergence among benign local updates. FedISM (Wu et al., 2024) is used as the tailored method for this scenario, as it mitigates fairness issues under client-specific quality shifts by matching inter-client sharpness for fair generalization.

Label Noise. Label noise models heterogeneity caused by unreliable local annotations. We randomly select 80% of clients as noisy clients. For each noisy client, the label corruption ratio is sampled from $U(0.5, 1)$, and the labels of the selected samples are randomly changed to incorrect classes. This setting creates severe client-dependent label corruption, where different clients may optimize toward inconsistent or even misleading supervision signals. Notably, when label noise is combined with backdoor attacks, this corruption is applied before malicious clients poison their local data, so the target labels are introduced after the label-noise process. FedCORR (Xu et al., 2022) is used as the tailored method because it identifies and corrects noisy labels during federated training.

Domain Shift. Domain shift models heterogeneity caused by distribution discrepancies across domains. Clients are partitioned according to their domains, with five clients in each domain. To ensure that backdoor attacks are distributed across domains, two clients in each domain are randomly selected as malicious clients. Under this setting, benign clients from different domains naturally

Table 14: Comparison with the defenses under different attack poison ratios on CIFAR-10.

Baselines	10% $\mathcal{A}\uparrow$	10% $\mathcal{B}\uparrow$	10% $\mathcal{V}\uparrow$	30% $\mathcal{A}\uparrow$	30% $\mathcal{B}\uparrow$	30% $\mathcal{V}\uparrow$	70% $\mathcal{A}\uparrow$	70% $\mathcal{B}\uparrow$	70% $\mathcal{V}\uparrow$	90% $\mathcal{A}\uparrow$	90% $\mathcal{B}\uparrow$	90% $\mathcal{V}\uparrow$
Attack: BadNet												
FoolsGold	84.43	48.11	66.27	83.99	35.63	59.81	82.62	23.91	53.27	81.19	20.84	51.02
FDCR	86.70	0.02	43.36	86.43	0.00	43.22	86.02	0.00	43.01	85.02	0.00	42.51
SPMC	86.60	0.02	43.31	86.94	0.00	43.47	86.12	0.00	43.06	85.46	0.00	42.73
MARS	65.28	19.52	42.40	20.88	**100.00**	60.44	70.90	97.96	84.43	77.93	98.12	88.03
AlignIns	85.54	87.39	86.46	85.07	96.71	90.89	85.16	96.75	90.96	85.15	96.75	90.95
MASA	84.57	46.51	65.54	82.51	31.10	56.81	81.00	25.16	53.08	83.69	30.09	56.89
FedDLAD	57.95	74.57	66.26	57.65	28.52	43.09	56.24	0.14	28.19	46.33	0.00	23.17
FilterFL	86.98	0.02	43.50	87.02	0.00	43.51	86.10	0.00	43.05	85.22	0.00	42.61
FedAVG	86.74	0.02	43.38	86.85	0.00	43.43	86.22	0.00	43.11	84.90	0.00	42.45
+Eraser	**87.03**	**98.85**	**92.94**	**87.66**	98.21	**92.94**	**87.29**	**98.42**	**92.86**	**87.46**	**98.17**	**92.82**
Attack: DBA												
FoolsGold	84.35	87.98	86.17	83.82	83.29	83.56	82.36	79.43	80.90	81.03	79.19	80.11
FDCR	86.76	21.52	54.14	86.76	19.81	53.29	85.64	17.25	51.45	85.22	13.04	49.13
SPMC	86.68	20.64	53.66	86.74	12.71	49.73	86.18	10.10	48.14	85.65	6.00	45.83
MARS	10.38	0.00	5.19	34.31	97.94	66.13	77.63	**98.69**	88.16	77.92	98.19	88.06
AlignIns	85.68	96.20	90.94	85.16	98.37	91.77	85.12	98.34	91.73	85.13	**98.37**	91.75
MASA	84.50	87.84	86.17	83.55	79.95	81.75	82.50	81.88	82.19	81.14	82.27	81.71
FedDLAD	58.24	93.79	76.02	58.45	76.17	67.31	57.52	13.89	35.71	46.83	0.48	23.66
FilterFL	86.83	21.27	54.05	86.72	10.96	48.84	86.26	6.40	46.33	85.03	6.92	45.98
FedAVG	86.81	21.56	54.19	86.10	15.57	50.84	86.22	8.01	47.12	85.38	6.71	46.05
+Eraser	**87.42**	**98.30**	**92.86**	**87.08**	**98.40**	**92.74**	**87.45**	98.44	**92.95**	**87.33**	**98.37**	**92.85**
Attack: Signal												
FoolsGold	84.20	80.10	82.15	83.67	55.12	69.39	82.02	35.68	58.85	79.84	41.75	60.79
FDCR	86.75	0.34	43.55	86.26	0.12	43.19	85.48	0.03	42.75	85.36	0.32	42.84
SPMC	86.93	0.28	43.61	86.71	0.07	43.39	86.54	0.01	43.27	86.25	0.26	43.26
MARS	78.20	50.15	64.18	46.12	**100.00**	73.06	70.84	**100.00**	85.42	77.84	**100.00**	88.92
AlignIns	84.37	96.71	90.54	85.15	97.97	91.56	85.38	97.97	91.67	85.79	97.99	91.89
MASA	84.43	79.47	81.95	83.22	37.73	60.47	81.90	46.08	63.99	80.58	52.41	66.49
FedDLAD	52.72	**100.00**	76.36	54.13	99.92	77.02	57.85	74.66	66.25	50.50	23.18	36.84
FilterFL	86.60	0.16	43.38	86.86	0.07	43.46	86.38	0.03	43.20	86.11	0.26	43.18
FedAVG	86.98	0.33	43.65	86.72	0.04	43.38	86.68	0.01	43.35	86.14	0.36	43.25
+Eraser	**87.11**	99.11	**93.11**	**86.93**	98.30	**92.62**	**87.61**	99.83	**93.72**	**86.80**	99.16	**92.98**
Attack: Blended												
FoolsGold	84.26	75.64	79.95	83.76	40.46	62.11	82.79	35.68	59.23	81.48	52.33	66.90
FDCR	86.90	0.01	43.45	86.69	0.00	43.35	86.27	0.00	43.14	86.18	0.01	43.10
SPMC	87.00	0.01	43.51	87.04	0.00	43.52	86.54	0.00	43.27	86.25	0.02	43.14
MARS	37.71	93.42	65.57	39.07	14.21	26.64	61.95	**100.00**	80.97	75.61	**100.00**	87.80
AlignIns	84.12	97.20	90.66	85.17	98.00	91.58	85.80	98.98	92.39	85.77	98.98	92.38
MASA	84.50	76.05	80.27	83.60	25.29	54.45	82.64	45.77	64.20	81.82	68.55	75.18
FedDLAD	52.38	**100.00**	76.19	54.03	**99.96**	76.99	58.67	82.53	70.60	53.23	42.65	47.94
FilterFL	87.00	0.01	43.50	87.07	0.00	43.54	86.89	0.00	43.44	86.40	0.01	43.20
FedAVG	87.03	0.01	43.52	86.72	0.00	43.36	86.54	0.00	43.27	86.26	0.03	43.15
+Eraser	**87.04**	98.98	**93.01**	**87.34**	99.71	**93.53**	**87.21**	99.84	**93.53**	**87.01**	99.92	**93.46**

produce diverse updates, making it more difficult to distinguish malicious updates from domain-induced variations. FedHEAL (Chen et al., 2024) is used as the tailored method for this scenario, as it improves fairness under domain heterogeneity by mitigating parameter-update conflicts and aggregation bias across clients with diverse domains.

Training Speed Skew. Training speed skew models system heterogeneity in asynchronous FL. Clients are divided into fast, medium, and slow groups, and each group is assigned a different range of training time, such as 1–2, 3–5, and 5–8 time units. During asynchronous training, the server does not wait for all sampled clients to finish local training. Instead, the server updates the global model once it receives the first ten completed client updates. Clients that train more slowly may finish after the global model has already been updated for several rounds, and their delayed updates, computed from older global models, may still be received and aggregated by the server. This setting introduces both update staleness and participation imbalance, since faster clients contribute more frequently while slower clients tend to provide stale information. FADAS (Wang et al., 2024) is used as the tailored method for this scenario, as it improves asynchronous federated adaptive optimization with delay-aware learning adjustment for straggler clients.

J Performance on Larger Scales, New Clients, and Severe Heterogeneity.

We further evaluate Eraser under challenging FL settings, including larger client populations, newly joined clients, and severe non-IID data with Dirichlet parameter $dir(0.1)$.

Table 15: Comparison with the defenses with 100 clients on CIFAR-10.

Baselines	BadNet			DBA			Signal			Blended		
	$\mathcal{A}\uparrow$	$\mathcal{B}\uparrow$	$\mathcal{V}\uparrow$	$\mathcal{A}\uparrow$	$\mathcal{B}\uparrow$	$\mathcal{V}\uparrow$	$\mathcal{A}\uparrow$	$\mathcal{B}\uparrow$	$\mathcal{V}\uparrow$	$\mathcal{A}\uparrow$	$\mathcal{B}\uparrow$	$\mathcal{V}\uparrow$
FoolsGold	81.31	11.01	46.16	81.05	41.39	61.22	80.88	65.74	73.31	80.80	54.18	67.49
FDCR	**84.96**	0.00	42.48	84.36	0.00	42.18	83.82	0.14	41.98	83.95	0.01	41.98
SPMC	84.59	0.00	42.30	83.64	0.00	41.82	84.15	0.11	42.13	84.71	0.00	42.36
MARS	-	-	-	-	-	-	-	-	-	-	-	-
AlignIns	81.31	97.01	89.16	81.16	**98.39**	89.78	81.48	**98.98**	90.23	81.43	98.99	90.21
MASA	81.18	13.10	47.14	81.08	48.97	65.03	81.07	74.75	77.91	80.93	65.59	73.26
FedDLAD	69.58	5.99	37.79	68.25	79.13	73.69	66.04	98.31	82.18	66.32	99.09	82.71
FilterFL	84.67	0.00	42.34	**85.27**	0.00	42.64	83.49	0.05	41.77	84.27	0.00	42.13
FedAVG	84.63	0.00	42.32	83.40	0.00	41.70	83.27	0.23	41.75	84.70	0.00	42.35
+Eraser	84.95	**98.94**	**91.94**	85.13	98.31	**91.72**	**84.74**	98.24	**91.49**	**85.01**	**99.52**	**92.26**

Table 16: Performance of Eraser with different client scales under BadNet on CIFAR-10.

Client Scale	Avg. Samples / Client	$\mathcal{A}\uparrow$	$\mathcal{B}\uparrow$	$\mathcal{V}\uparrow$
100	$\sim$449	**84.95**	**98.94**	**91.94**
200	$\sim$224	77.35	96.93	87.14
300	$\sim$149	74.18	96.64	85.41

Table 17: Performance on 10 newly joined clients under BadNet on CIFAR-10.

Baseline	$\mathcal{A}\uparrow$	$\mathcal{B}\uparrow$	$\mathcal{V}\uparrow$
FedAVG + ***Eraser***	87.49	98.81	93.15

Table 18: Comparison with the defenses with $dir(0.1)$ on CIFAR-10.

Baselines	BadNet			DBA			Signal			Blended		
	$\mathcal{A}\uparrow$	$\mathcal{B}\uparrow$	$\mathcal{V}\uparrow$	$\mathcal{A}\uparrow$	$\mathcal{B}\uparrow$	$\mathcal{V}\uparrow$	$\mathcal{A}\uparrow$	$\mathcal{B}\uparrow$	$\mathcal{V}\uparrow$	$\mathcal{A}\uparrow$	$\mathcal{B}\uparrow$	$\mathcal{V}\uparrow$
FoolsGold	78.20	19.45	48.82	78.00	68.60	73.30	77.78	59.88	68.83	78.34	50.78	64.56
FDCR	74.72	0.00	37.36	73.91	0.99	37.45	72.06	0.02	36.04	72.54	0.00	36.27
SPMC	77.53	0.00	38.77	77.35	0.02	38.69	78.72	0.00	39.36	78.47	0.00	39.24
MARS	-	-	-	-	-	-	-	-	-	-	-	-
AlignIns	80.04	94.62	87.33	80.11	97.67	88.89	79.90	98.94	89.42	79.88	98.98	89.43
MASA	74.38	15.84	45.11	74.41	64.39	69.40	76.97	65.68	71.33	76.21	57.81	67.01
FedDLAD	60.58	0.94	30.76	60.82	23.55	42.19	62.16	88.07	75.12	61.92	89.54	75.73
FilterFL	75.74	0.00	37.87	76.21	0.31	38.26	76.15	0.00	38.08	76.67	0.00	38.34
FedAVG	78.64	0.00	39.32	78.37	0.39	39.38	78.42	0.00	39.21	78.21	0.00	39.10
+Eraser	**82.89**	**98.04**	**90.47**	**82.21**	**97.91**	**90.06**	**81.64**	**99.58**	**90.61**	**82.61**	**99.49**	**91.05**

Performance with 100 clients. Table 15 shows that Eraser remains effective when the number of clients increases to 100, with only 20% participating in each round. Eraser achieves an average $\mathcal{A}$ of 84.96%, $\mathcal{B}$ of 98.76%, and $\mathcal{V}$ of 91.85%. Increasing the client population reduces the number of local samples per client and enlarges update diversity, making malicious-client identification more difficult. Consequently, FedAVG, FDCR, SPMC, and FilterFL fail to suppress most attacks, while AlignIns achieves strong robustness at the cost of lower main-task accuracy. In contrast, Eraser maintains a better utility–robustness trade-off. This scale is also consistent with common FL backdoor-defense settings, where the number of clients typically ranges from 10 to 100 (Huang et al., 2023b;a; Qin et al., 2024; Huang et al., 2024b; He et al., 2025; Wan et al., 2026; Xu et al., 2025a;b; Ding et al., 2025a; Yang et al., 2025).

Performance at larger client scales. We further increase the client population to 200 and 300 while keeping the total training data unchanged. As shown in Table 16, the average number of samples per client decreases from approximately 449 to 149 as the client population grows. Although $\mathcal{A}$ decreases because each client has less local data for learning an effective Featurizer, Eraser maintains high robustness, with $\mathcal{B}$ above 96.6% even with 300 clients. These results indicate that Eraser remains effective as the federation scales to substantially more clients.

Performance with newly joined clients. We also examine whether Eraser can accommodate clients that join after the global model and shared Generator have been trained. A newly joined client only optimizes its local Featurizer using its private data and the fixed semantic anchors, while both the global model and shared Generator remain frozen. Thus, neither needs to be retrained. We introduce 10 new clients that do not participate in the original federated training or collaborative Eraser training. As shown in Table 17, Eraser achieves $\mathcal{A} = 87.49\%$, $\mathcal{B} = 98.81\%$, and $\mathcal{V} = 93.15\%$, demonstrating its effectiveness for newly joined clients.

Performance under severe non-IID data. Table 18 evaluates Eraser under the more challenging $dir(0.1)$ setting. Eraser achieves the best overall performance across all four attacks, with an average $\mathcal{A}$ of 82.34%, $\mathcal{B}$ of 98.76%, and $\mathcal{V}$ of 90.55%. Severe data heterogeneity increases the diversity of benign updates and makes malicious-update identification more error-prone. Consequently, FDCR, SPMC, FilterFL, and FedAVG fail to recover robustness, while FoolsGold, MASA, and FedDLAD remain unstable across attacks. Although AlignIns maintains strong robustness, its lower $\mathcal{A}$ and $\mathcal{V}$ indicate a larger utility loss. In contrast, Eraser relies on post-training semantic reconstruction rather than update separability, enabling a more stable utility–robustness trade-off under severe non-IID data.

Table 19: Compatibility with different FL algorithms on CIFAR-10.

Baselines	BadNet			DBA			Signal			Blended		
	$\mathcal{A}\uparrow$	$\mathcal{B}\uparrow$	$\mathcal{V}\uparrow$	$\mathcal{A}\uparrow$	$\mathcal{B}\uparrow$	$\mathcal{V}\uparrow$	$\mathcal{A}\uparrow$	$\mathcal{B}\uparrow$	$\mathcal{V}\uparrow$	$\mathcal{A}\uparrow$	$\mathcal{B}\uparrow$	$\mathcal{V}\uparrow$
FedPW	88.50	0.00	44.25	**88.97**	10.65	49.81	88.82	1.22	45.02	88.24	0.65	44.45
+AlignIns	83.75	97.88	90.82	83.07	99.25	91.16	82.78	**99.99**	91.39	82.93	**100.00**	91.47
+MARS	81.91	2.60	42.26	83.14	22.60	52.87	82.74	0.90	41.82	82.95	0.42	41.69
+FilterFL	82.30	0.00	41.15	81.42	15.65	48.54	82.92	2.41	42.67	82.24	0.38	41.31
+FDCR	80.06	0.02	40.04	81.97	21.16	51.57	80.47	0.55	40.51	79.28	1.05	40.17
+Eraser	**88.90**	**98.02**	**93.46**	88.89	**99.35**	**94.12**	**88.89**	99.02	**93.96**	**89.47**	99.94	**94.71**
FedAWA	88.70	0.00	44.35	**88.28**	5.39	46.84	88.39	0.00	44.20	88.89	0.00	44.45
+AlignIns	86.98	97.67	92.33	87.20	98.91	93.06	86.89	99.71	93.30	87.03	**99.92**	93.48
+MARS	87.03	0.00	43.52	86.79	2.20	44.50	86.23	0.02	43.13	86.46	0.00	43.23
+FilterFL	86.88	0.00	43.44	86.60	6.83	46.72	86.67	0.01	43.34	86.91	0.00	43.46
+FDCR	86.63	0.00	43.32	86.33	13.77	50.05	86.19	0.03	43.11	86.41	0.00	43.21
+Eraser	**89.45**	**98.11**	**93.78**	87.64	**99.53**	**93.59**	**89.34**	**99.78**	**94.56**	**89.25**	99.86	**94.56**
MOON	**89.31**	0.00	44.66	88.68	5.28	46.98	89.29	0.01	44.65	88.71	0.00	44.36
+AlignIns	86.44	97.69	92.07	86.63	98.49	92.56	86.42	99.69	93.06	86.30	**99.96**	93.13
+MARS	86.60	0.00	43.30	86.72	4.22	45.47	85.74	0.01	42.88	85.30	0.01	42.66
+FilterFL	85.25	0.00	42.63	85.11	3.77	44.44	86.21	0.00	43.11	86.90	0.00	43.45
+FDCR	85.42	0.00	42.71	85.70	8.43	47.07	85.75	0.00	42.88	85.86	0.00	42.93
+Eraser	89.14	**98.01**	**93.58**	**89.81**	**99.45**	**94.63**	**89.77**	**99.92**	**94.85**	**89.10**	99.88	**94.49**

K Performance on Compatibility with Different FL Algorithms

We evaluate compatibility across three FL algorithms: FedPW (Guan et al.), FedAWA (Shi et al., 2025), and MOON (Li et al., 2021). To assess compatibility, we combine several defenses that can identify malicious clients with these FL algorithms by excluding suspected malicious updates before aggregation. Specifically, we consider AlignIns, MARS, FilterFL, and FDCR as pre-aggregation filtering defenses. Table 19 shows that FedPW, FedAWA, and MOON alone fail to defend against backdoor attacks, with $\mathcal{B}$ close to zero under most attacks. Their adapted combinations with existing defenses also show unstable performance. AlignIns achieves strong robustness in most cases, but it noticeably reduces $\mathcal{A}$. In contrast, MARS, FilterFL, and FDCR fail to consistently identify malicious clients across different FL algorithms. By contrast, Eraser remains effective across all FL algorithms and attacks. FedPW***+Eraser***, FedAWA***+Eraser***, and MOON***+Eraser*** achieve consistently high $\mathcal{B}$, ranging from 98.01% to 99.94%, while maintaining $\mathcal{A}$ around 89%. They also obtain the best $\mathcal{V}$ in all settings, with average $\mathcal{V}$ of 94.06%, 94.12%, and 94.39% under FedPW, FedAWA, and MOON, respectively. These results show that Eraser is highly compatible with different FL algorithms. Since Eraser removes trigger-induced effects through post-training semantic reconstruction, it does not rely on the separability of benign and malicious updates before aggregation, making it more stable than combinations based on malicious-client identification.

Table 20: Comparison of FedAVG and Eraser with ViT on CIFAR-10.

Baselines	BadNet			DBA			Signal			Blended		
	$\mathcal{A}\uparrow$	$\mathcal{B}\uparrow$	$\mathcal{V}\uparrow$	$\mathcal{A}\uparrow$	$\mathcal{B}\uparrow$	$\mathcal{V}\uparrow$	$\mathcal{A}\uparrow$	$\mathcal{B}\uparrow$	$\mathcal{V}\uparrow$	$\mathcal{A}\uparrow$	$\mathcal{B}\uparrow$	$\mathcal{V}\uparrow$
FedAVG	96.57	0.50	48.54	96.63	0.12	48.38	**96.91**	0.00	48.46	96.68	0.00	48.34
+Eraser	**96.69**	**98.85**	**97.77**	**96.74**	**99.20**	**97.97**	96.63	**98.41**	**97.52**	**96.70**	**98.43**	**97.57**

L Performance on Vision Transformer

We evaluate Eraser with a Vision Transformer (ViT) backbone on CIFAR-10 to examine whether Eraser remains effective beyond convolutional architectures. Table 20 compares FedAVG and FedAVG***+Eraser*** under BadNet, DBA, Signal, and Blended attacks. The results show that FedAVG

with ViT achieves high clean accuracy, but still fails to defend against backdoor attacks. Across the four attacks, its $\mathcal{B}$ remains close to zero, indicating that the model is highly vulnerable once trigger-target correlations are injected. By contrast, FedAVG+***Eraser*** consistently suppresses backdoor activation while preserving clean accuracy. It achieves $\mathcal{B}$ from 98.41% to 99.20% and maintains $\mathcal{A}$ around 96.7% across all attacks. Moreover, Eraser obtains $\mathcal{V}$ above 97.5% in every setting, showing a strong trade-off between clean accuracy and robustness. These results indicate that Eraser is not limited to CNN-based models. By reconstructing samples from trigger-insensitive semantic features, Eraser remains effective with ViT and provides stable robustness across different model architectures.

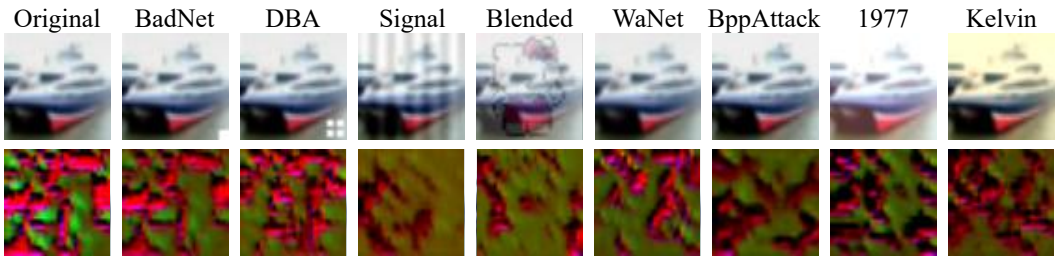

Figure 6: Visualization of input images and their reconstructed counterparts under different trigger types. The first row shows the original clean image and triggered images, while the second row shows the corresponding samples reconstructed by Eraser.

Table 21: Trigger Projection Residual (TPR) of reconstructed samples under different trigger types. Lower values indicate weaker trigger-specific residual signals after reconstruction.

Metric	BadNet	DBA	Signal	Blended	WaNet	BppAttack	1977	Kelvin
TPR ↓	0.0031	0.0016	0.0215	0.0187	0.0199	0.0532	0.0530	0.0540

M VISUALIZATION OF RECONSTRUCTED SAMPLES

Fig. 6 visualizes the reconstruction results of Eraser under different trigger types. The first row shows the original clean image and its triggered variants generated by different backdoor attacks. The second row presents the corresponding images reconstructed by Eraser. As observed, although the triggered inputs preserve clear object semantics and attack-specific trigger patterns, the reconstructed images no longer retain visually recognizable trigger structures. This indicates that Eraser weakens trigger-related cues during reconstruction, thereby preventing the poisoned global model from being activated by the original backdoor patterns.

To further quantify this observation, we introduce the Trigger Projection Residual (TPR), which measures how much of the original trigger-specific signal is preserved in the reconstruction residual. Given a clean sample x and its triggered counterpart $x \oplus \delta$, where δ denotes the trigger pattern, we define the original trigger-induced difference as $\Delta_x = (x \oplus \delta) - x$ and the reconstruction residual as $\Delta_{\text{rec}} = R(F(x \oplus \delta)) - R(F(x))$. We then compute the TPR only on benign clients as:

$$\text{TPR} = \frac{|\langle \Delta_{\text{rec}}, \Delta_x \rangle|}{\|\Delta_x\|_2^2 + 10^{-8}}. \tag{18}$$

TPR measures how much of the reconstruction residual remains aligned with the original trigger-induced direction. A lower TPR indicates that the reconstructed sample retains less trigger-specific signal. As shown in Table 21, Eraser achieves consistently low TPR values across all evaluated trigger types. For localized triggers such as BadNet and DBA, the TPR values are only 0.0031 and 0.0016, respectively, indicating that almost no trigger-aligned signal remains after reconstruction. For more complex triggers such as Signal, Blended, and WaNet, the TPR values remain below 0.022, showing that Eraser also suppresses distributed trigger patterns. Even for visually more pervasive or style-based triggers such as BppAttack, 1977, and Kelvin, the TPR values are around $0.053 \sim 0.054$, meaning that less than 5.4% of the original trigger-aligned signal is retained. On average, the TPR over all trigger types is 0.0281. These results quantitatively support the visual evidence in Fig. 6: Eraser does not merely alter the input appearance, but effectively erases the trigger-induced effect component that could activate the poisoned global model.



Figure 7: Original image and its processed variants under several pre-processing input operations.

Table 22: Comparison with pre-processing and centralized defenses on CIFAR-10.

Baselines	BadNet			DBA			Signal			Blended		
	$\mathcal{A}$	$\mathcal{B}$	$\mathcal{V}$	$\mathcal{A}$	$\mathcal{B}$	$\mathcal{V}$	$\mathcal{A}$	$\mathcal{B}$	$\mathcal{V}$	$\mathcal{A}$	$\mathcal{B}$	$\mathcal{V}$
FedAVG	86.23	0.00	43.12	85.57	0.00	42.78	85.51	0.00	42.76	86.06	0.00	43.03
+Rotate	54.77	62.01	58.39	54.59	62.87	58.73	57.91	3.40	30.65	56.59	9.25	32.92
+Flip	84.07	97.25	90.66	84.05	96.47	90.26	85.23	14.56	49.90	83.35	1.58	42.46
+Crop	65.87	65.97	65.92	63.60	61.22	62.41	65.71	23.40	44.55	65.63	47.12	56.38
+Gaussian	49.38	34.94	42.16	47.10	43.80	45.45	53.41	0.00	26.71	52.28	0.00	26.14
+Compression	38.48	24.77	31.63	38.41	44.98	41.70	37.72	0.02	18.87	39.55	0.97	20.26
+Neural Cleanse	71.95	68.31	70.13	76.05	42.22	59.13	82.61	3.83	43.22	78.46	17.13	47.79
+RNP	77.21	73.05	75.13	74.68	63.91	69.30	76.44	18.37	47.41	72.93	41.82	57.38
+NFT	80.49	67.73	74.11	78.55	70.84	74.69	81.12	32.47	56.80	77.86	36.95	57.41
+IMS	51.39	9.19	30.29	56.28	14.63	35.45	48.71	1.84	25.28	53.96	7.57	30.77
+CLP	73.23	0.00	36.62	69.84	6.92	38.38	75.17	2.31	38.74	71.46	13.88	42.67
+Eraser	**87.68**	**98.76**	**93.22**	**87.67**	**98.37**	**93.02**	**87.89**	**99.77**	**93.83**	**87.95**	**99.90**	**93.93**
Baselines	WaNet			BppAttack			1977			Kelvin		
	$\mathcal{A}$	$\mathcal{B}$	$\mathcal{V}$	$\mathcal{A}$	$\mathcal{B}$	$\mathcal{V}$	$\mathcal{A}$	$\mathcal{B}$	$\mathcal{V}$	$\mathcal{A}$	$\mathcal{B}$	$\mathcal{V}$
FedAVG	84.66	22.89	53.77	84.87	2.63	43.75	86.92	3.24	45.08	86.81	2.46	44.63
+Rotate	46.14	92.00	69.07	59.79	0.53	30.16	53.07	0.61	26.84	53.77	0.50	27.14
+Flip	82.06	86.37	84.22	83.12	2.33	42.73	84.56	2.48	43.52	84.85	1.74	43.29
+Crop	60.50	80.28	70.39	65.14	3.24	34.19	65.80	5.62	35.71	64.48	1.68	33.08
+Gaussian	38.80	**99.74**	69.27	61.21	0.84	31.03	51.52	1.25	26.39	52.90	2.24	27.57
+Compression	31.09	53.67	42.38	11.70	2.03	6.87	10.93	1.43	6.18	12.30	1.92	7.11
+Neural Cleanse	74.76	52.33	63.55	75.36	85.56	80.46	74.31	73.44	73.88	70.14	76.33	73.23
+RNP	75.62	68.44	72.03	73.81	52.17	62.99	77.36	44.93	61.14	71.94	38.66	55.30
+NFT	79.18	62.53	70.86	81.04	49.76	65.40	76.92	56.31	66.62	80.37	45.22	62.80
+IMS	54.72	17.86	36.29	49.31	3.44	26.38	55.83	11.29	33.56	47.96	2.18	25.07
+CLP	70.53	27.41	48.97	74.26	5.93	40.09	68.91	19.64	44.27	73.88	8.27	41.07
+Eraser	**87.88**	99.50	**93.69**	**87.93**	**99.43**	**93.68**	**88.00**	**99.28**	**93.64**	**87.84**	**99.15**	**93.50**

N Performance on Pre-processing and Centralized Defenses.

Since Eraser is a post-training defense, we compare it with several post-training baselines applied to trained FedAVG poisoned models. For each attack, we first train a poisoned global model using FedAVG and then apply different defense operations before inference, without modifying the original FL training process. For input-level defenses, these operations are applied as pre-processing transformations before the samples are evaluated by the trained poisoned model, as shown in Fig. 7. Their goal is to disrupt or remove the trigger pattern at inference time. We consider random rotation with angles uniformly sampled from $[-45°, 45°]$, horizontal flipping, random cropping with padding 4, Gaussian noise injection with zero mean and standard deviation 0.20, and JPEG compression followed by decompression with quality 30.

We also include representative centralized post-training defenses, including Neural Cleanse (Wang et al., 2019), RNP (Li et al., 2023b), NFT (Karim et al., 2024), IMS (Dunnett et al., 2026), and CLP (Zheng et al., 2022), with their default settings. Unlike the above input-level transformations, these methods directly purify the trained poisoned model. Neural Cleanse, RNP, NFT, and IMS require benign task-relevant data for trigger inspection, fine-tuning, or pruning, whereas CLP performs data-free channel pruning based on channel Lipschitzness. To adapt them to the FL setting without assuming an additional server-side IID dataset, we distribute the poisoned global model to benign clients and let each client independently perform purification using only its local non-IID data; CLP is applied in its original data-free manner.

As shown in Table 22, simple image transformations provide inconsistent backdoor mitigation. Although some transformations suppress specific triggers, their effectiveness is highly attack-dependent and often accompanied by substantial utility degradation. For example, Flip achieves high $\mathcal{B}$ on BadNet, DBA, and WaNet, but is largely ineffective against Signal, Blended, BppAttack, 1977, and Kelvin. Rotate, Crop, Gaussian noise, and Compression show similar instability because perturbing low-level patterns may also damage task-relevant semantics. This is especially evident for Gaussian noise and Compression, whose average $\mathcal{A}$ drops to 50.83% and 27.52%, respectively.

Centralized post-training defenses also exhibit limited effectiveness when transferred to FL. Neural Cleanse, RNP, and NFT can mitigate certain attacks, but their robustness varies considerably across trigger types when purification relies on heterogeneous local data. Their $\mathcal{B}$ values drop substantially on several attacks, suggesting that limited non-IID data may be insufficient to reliably expose and repair backdoor-related behaviors. IMS is less stable, with an average $\mathcal{B}$ of only 8.50%. Similarly, although CLP is data-free, its channel-pruning strategy achieves an average $\mathcal{B}$ of only 10.55%. Overall, directly transferring centralized post-training defenses to FL remains challenging because their data assumptions and model-repair mechanisms are not designed for decentralized and statistically heterogeneous data.

We further evaluate these centralized post-training defenses under more severe heterogeneity with $dir(0.1)$. As shown in Table 23, their performance degrades further. Neural Cleanse and NFT achieve only 32.50% and 27.92% $\mathcal{B}$, while RNP, IMS, and CLP perform even worse and incur substantial drops in $\mathcal{A}$. In contrast, Eraser achieves 82.89% $\mathcal{A}$, 98.04% $\mathcal{B}$, and 90.47% $\mathcal{V}$, further demonstrating its robustness under severe non-IID data.

Table 23: Comparison with centralized post-training backdoor purification methods under BadNet on CIFAR-10 with $dir(0.1)$.

Baselines	$\mathcal{A}$	$\mathcal{B}$	$\mathcal{V}$
FedAVG	78.64	0.00	39.32
+Neural Cleanse	62.83	32.50	47.67
+RNP	57.98	11.44	34.71
+NFT	70.63	27.92	49.28
+IMS	44.97	1.80	23.39
+CLP	68.40	1.07	34.74
+Eraser	**82.89**	**98.04**	**90.47**

As shown in Table 23, centralized post-training defenses degrade substantially under stronger heterogeneity $dir(0.1)$. Neural Cleanse and NFT achieve only 32.50% and 27.92% $\mathcal{B}$, while RNP, IMS, and CLP perform even worse and substantially reduce $\mathcal{A}$. In contrast, Eraser achieves 82.89% $\mathcal{A}$, 98.04% $\mathcal{B}$, and 90.47% $\mathcal{V}$, demonstrating much stronger robustness to severe non-IID data.

Overall, Eraser differs from centralized post-training purification in both setting and mechanism. Centralized methods mainly repair a single poisoned model through trigger inversion, fine-tuning, or pruning, often relying on benign auxiliary data. In contrast, Eraser is designed for decentralized non-IID FL and performs sample-level purification at inference without modifying the poisoned global model or requiring a server-side clean dataset.

Table 24: Comparison under adaptive Blended attacks with different blending ratios on CIFAR-10.

Baselines	$\chi = 1.0$			$\chi = 0.8$			$\chi = 0.6$			$\chi = 0.4$		
	$\mathcal{A}\uparrow$	$\mathcal{B}\uparrow$	$\mathcal{V}\uparrow$	$\mathcal{A}\uparrow$	$\mathcal{B}\uparrow$	$\mathcal{V}\uparrow$	$\mathcal{A}\uparrow$	$\mathcal{B}\uparrow$	$\mathcal{V}\uparrow$	$\mathcal{A}\uparrow$	$\mathcal{B}\uparrow$	$\mathcal{V}\uparrow$
FedAVG	85.83	0.00	42.92	85.70	0.00	42.85	85.87	0.00	42.94	85.98	0.00	42.99
+Eraser	**86.74**	**93.09**	**89.92**	**86.53**	**94.49**	**90.51**	**86.76**	**97.14**	**91.95**	**87.12**	**99.86**	**93.49**

O Performance on Adaptive Attack and Label-Only Manipulation Attack

We consider an adaptive attacker that is aware of the semantic reconstruction mechanism of Eraser. Instead of changing the trigger type, the attacker strengthens the Blended attack by increasing its blending ratio. In the default Blended attack, the triggered sample is generated by mixing a clean sample with a trigger pattern. Here, we gradually increase the blending ratio to evaluate whether Eraser remains robust when the trigger becomes more dominant in the input. Given a clean sample x and a Blended trigger pattern δ, we denote the triggered sample as $x \oplus \delta_\chi$, where χ is the blending ratio. Specifically, the trigger operation is defined as:

$$x \oplus \delta_\chi = (1 - \chi)x + \chi\delta, \tag{19}$$

where $\chi \in [0, 1]$ controls the strength of the blended trigger. The default Blended attack uses $\chi = 0.2$. In this adaptive setting, we gradually increase χ to make the trigger pattern more dominant in the input. When $\chi = 1.0$, the original sample is fully replaced by the trigger pattern.

The goal of this attack is to challenge Eraser by weakening the semantic information that can be extracted from the triggered input. Since Eraser relies on trigger-insensitive semantic features for reconstruction, a larger blending ratio may reduce the reliability of the extracted features and make the reconstruction process harder. In this adaptive attack, the malicious client trains the Featurizer F_i only on clean data, so that the following Generator becomes more sensitive to abnormal features induced by high-strength blended triggers. For a malicious client with target label y_t, the adaptive training objective is:

$$\mathcal{L}_i^{\text{ada}} = \mathbb{E}_{(x,y)\in\mathcal{D}_i}\left[\mathcal{L}_{\text{CE}}\big(\mathcal{G}(R_i(F_i(x))), y\big)\right] + \mathbb{E}_{(x,y)\in\mathcal{D}_i}\left[\mathcal{L}_{\text{CE}}\big(\mathcal{G}(R_i(F_i(x \oplus \delta_\chi))), y_t\big)\right]. \quad (20)$$

The first term preserves normal reconstruction on clean samples, while the second term encourages reconstructions from high-strength blended inputs to be classified as the target label. Compared with the standard Blended attack, this adaptive variant is stronger because increasing χ makes the trigger more dominant and progressively weakens the task-relevant semantics in the input.

Table 24 shows that FedAVG completely fails under the adaptive Blended attacks, with $\mathcal{B} = 0.00\%$ for all blending ratios. In contrast, Eraser remains robust across all adaptive settings. Even when $\chi = 1.0$, where the input is fully replaced by the trigger pattern, Eraser still achieves 93.09% $\mathcal{B}$ and 89.92% $\mathcal{V}$. As χ decreases from 1.0 to 0.4, more task-relevant information from the original sample is preserved, and Eraser improves $\mathcal{B}$ from 93.09% to 99.86%. Overall, these results show that Eraser remains effective against adaptive attacks that strengthen the trigger by increasing the blending ratio.

Table 25: Performance under label-only manipulation during Eraser training on CIFAR-10.

Attack	$\mathcal{A}\uparrow$	$\mathcal{B}\uparrow$	$\mathcal{V}\uparrow$
BadNet	87.69	97.31	92.50
DBA	87.56	98.16	92.86
Signal	**87.87**	99.81	**93.84**
Blended	87.34	**99.86**	93.60

We further consider a label-only manipulation attack, where malicious clients flip their local labels during Eraser training without injecting additional triggers. Such manipulation may bias local optimization and poison the uploaded Generator updates. As shown in Table 25, Eraser remains robust across all four attacks, with $\mathcal{B}$ ranging from 97.31% to 99.86% and $\mathcal{V}$ from 92.50% to 93.84%. These results indicate that label manipulation alone is insufficient to substantially poison the aggregated Generator or compromise Eraser's reconstruction capability.

P Analysis of Different Anchor Designs

We further investigate whether Eraser is sensitive to the construction of semantic anchors and compare our hyperspherical anchors with two deterministic designs: orthogonal anchors and simplex equiangular anchors.

Hyperspherical Anchors. The hyperspherical anchors used by Eraser are not learnable parameters whose initialization affects subsequent optimization. For each class c, we sample a vector from an isotropic Gaussian distribution and project it onto the unit hypersphere:

$$\mathbf{a}_c = \frac{\mathbf{g}_c}{\|\mathbf{g}_c\|_2}, \qquad \mathbf{g}_c \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \quad (21)$$

All anchors are generated once and remain fixed throughout training. Since the Gaussian distribution is isotropic, the projected anchors are uniformly distributed over the unit hypersphere and do not favor any particular class or feature direction. In high-dimensional feature spaces, their pairwise similarities also concentrate around zero, naturally producing well-separated class anchors.

Orthogonal Anchors. We construct fixed orthogonal anchors such that different class anchors satisfy

$$\mathbf{a}_i^\top \mathbf{a}_j = 0, \quad i \neq j, \quad (22)$$

with $\|\mathbf{a}_i\|_2 = 1$. This deterministic construction explicitly removes pairwise correlations between class anchors and provides strong angular separation.

Simplex Equiangular Anchors. We further consider simplex equiangular anchors, where all class anchors have identical pairwise similarity:

$$\mathbf{a}_i^\top \mathbf{a}_j = -\frac{1}{C-1}, \quad i \neq j, \tag{23}$$

where C denotes the number of classes. This construction distributes the anchors symmetrically and maximizes their angular separation.

Table 26 compares these three designs under the same setting. Hyperspherical anchors achieve the best results across all metrics, with $\mathcal{A} = 87.68\%$, $\mathcal{B} = 98.76\%$, and $\mathcal{V} = 93.22\%$. Nevertheless, both deterministic designs also maintain high $\mathcal{B}$ above 96%, indicating that Eraser does not strongly depend on a particular anchor construction. Combined with the feature-discrepancy and anchor-dimensionality analyses, these results support the effectiveness and stability of the adopted hyperspherical anchor design.

Table 26: Performance of different anchor designs under BadNet on CIFAR-10.

Anchor Design	$\mathcal{A}\uparrow$	$\mathcal{B}\uparrow$	$\mathcal{V}\uparrow$
Orthogonal Anchors	86.51	98.15	92.33
Simplex Equiangular Anchors	86.96	96.90	91.93
Hyperspherical Anchors	**87.68**	**98.76**	**93.22**

Q Performance on Text/Speech Backdoor Extensions

To further evaluate whether Eraser is limited to visual inputs, we extend it to two non-visual modalities, including text classification and speech recognition. The core framework remains unchanged. Each client independently trains a local Featurizer to extract robust semantic representations, while the collaboratively trained Generator reconstructs a modality-specific representation that can be processed by the frozen global model. The federated setting, malicious-client ratio, poisoning ratio, attack schedule, semantic-anchor construction, and evaluation metrics are set to their default.

Q.1 Extension to Text Modality

Dataset and Model. For the text modality, we evaluate Eraser on the IMDB sentiment classification dataset. We use a two-layer bidirectional GRU as the global model, with a vocabulary size of 20,000, an embedding dimension of 64, a hidden dimension of 128 for each direction, and a maximum sequence length of 200. The local Featurizer adopts the same architecture as the global model and is independently trained from scratch on each client's local data. The final forward and backward hidden states are concatenated to form a 256-dimensional semantic representation.

Backdoor Attack. We adopt a word-level backdoor attack in which the tokens at positions 0, 1, and 2 of selected text sequences are replaced with the predefined trigger token 9999, and the corresponding labels are changed to the attacker-specified target class. Malicious clients poison 50% of their local samples and continue participating in both the main federated training and the subsequent Eraser training.

Modality-Specific Reconstruction. Unlike images, reconstructing discrete tokens would introduce a non-differentiable token-selection operation and unnecessarily constrain the Generator to reproduce the original sentence. We therefore reconstruct the continuous embedding representation instead of discrete text. Specifically, the Generator maps the 256-dimensional semantic feature produced by the local Featurizer into a sequence of 200 embeddings with dimension 64. The reconstructed embedding sequence directly enters the recurrent layers of the frozen global model, bypassing its token embedding lookup. This preserves the original objective of Eraser, which is to reconstruct task-discriminative information rather than reproduce the raw input. The Generator architecture is summarized in Table 27.

Q.2 Extension to Speech Modality

Dataset and Model. For the speech modality, we evaluate Eraser on the Speech Commands dataset with 10 keyword classes. Each audio sample is represented as a Mel spectrogram with 80 frequency

Table 27: Architecture of the Generator for the text modality.

Layer	Operation
Input	256-D latent feature vector
Linear	$256 \to 256$
LeakyReLU	LeakyReLU activation
Linear	$256 \to 200 \times 64$
Reshape	Reshape to 200×64 embedding sequence
Conv1D	$64 \to 128$, kernel size 3
BN + LeakyReLU	Batch normalization and LeakyReLU
Conv1D	$128 \to 64$, kernel size 3
BN + Tanh	Batch normalization and Tanh
Output	Reconstructed embedding sequence

bins and 87 temporal frames. We use a two-layer LSTM with a hidden dimension of 64 as the global model. The last temporal hidden state is used as the 64-dimensional semantic representation. The local Featurizer adopts the same architecture and is independently trained from scratch on each client's local data.

Backdoor Attack. For the speech modality, we adopt the PBSM attack (Cai et al., 2024). Unlike conventional speech backdoors that use perceptible noise or distinctive audio clips as triggers, PBSM exploits the pitch characteristics of speech to construct a stealthier trigger. Specifically, it inserts a short-duration high-pitched signal as the trigger and increases the pitch of the remaining audio to mask the injected signal. This design produces more natural poisoned samples while maintaining a strong backdoor effect.

Modality-Specific Reconstruction. For speech inputs, Eraser performs reconstruction in the Mel-spectrogram space. The local Featurizer first maps the input spectrogram into a 64-dimensional semantic representation. The Generator then progressively projects this representation back to a $1 \times 80 \times 87$ Mel spectrogram, which is directly processed by the frozen global model. Therefore, the same semantic reconstruction principle used for images is preserved without requiring access to the raw waveform. The Generator architecture is summarized in Table 28.

Table 28: Architecture of the Generator for the speech modality.

Layer	Operation
Input	64-D latent feature vector
Linear	$64 \to 2048$
BN + ReLU	Batch normalization and ReLU
Linear	$2048 \to 1024$
BN + ReLU	Batch normalization and ReLU
Linear	$1024 \to 80 \times 87$
Tanh	Tanh activation
Reshape	Reshape to $1 \times 80 \times 87$
Output	Reconstructed Mel spectrogram

Table 29: Performance of Eraser on text and speech modalities.

Baselines	IMDB			Speech Commands		
	$\mathcal{A}\uparrow$	$\mathcal{B}\uparrow$	$\mathcal{V}\uparrow$	$\mathcal{A}\uparrow$	$\mathcal{B}\uparrow$	$\mathcal{V}\uparrow$
FedAVG (Non-attack)	78.01	91.68	84.85	45.11	93.03	69.07
FedAVG	77.74	4.70	41.22	44.34	0.39	22.37
+Eraser	**78.90**	**95.54**	**87.22**	**47.93**	**95.51**	**71.72**

Q.3 RESULTS.

As shown in Table 29, FedAVG preserves main-task accuracy under both modalities but is highly vulnerable to backdoor attacks, with B decreasing to 4.70% on IMDB and 0.39% on Speech Commands. After applying Eraser, B increases to 95.54% and 95.51%, respectively, while maintaining comparable or higher main-task accuracy. These results demonstrate that Eraser is not restricted to

image-space reconstruction and can be extended to non-visual modalities through modality-specific semantic reconstruction.

R Algorithm Description

Algorithm 1 describes the overall procedure of Eraser. The server first freezes the trained global model $\mathcal{G}$, generates fixed hyperspherical semantic anchors A, and initializes the shared Generator $R^{(0)}$. In each Eraser round, selected clients first optimize their local Featurizers using $\mathcal{L}_{E_i} = \lambda\mathcal{L}_{\mathrm{STR}} + \mathcal{L}_{\mathrm{SEM}}$, and then train their Generators under the supervision of the frozen global model. The server aggregates the uploaded Generators by weighted averaging.

S Computational Complexity Analysis

We analyze the additional computational cost introduced by Eraser after the main federated training is completed. Let N denote the number of clients, $K = |\mathcal{S}_e|$ the number of selected clients in each Eraser training round, E the number of Eraser rounds, I the number of local epochs, and $n_i = |\mathcal{D}_i|$ the number of samples on client i. Let C be the number of classes and d be the feature dimension. We denote the per-sample forward-backward costs of the Featurizer, Generator, and frozen global model by Γ_F, Γ_R, and Γ_G, respectively. The parameter sizes of the global model, Featurizer, and Generator are denoted by P_G, P_F, and P_R.

Anchor Generation. Eraser first constructs a class-wise anchor matrix $A \in \mathbb{R}^{C\times d}$. Sampling and normalizing these anchors require $\mathcal{O}(Cd)$ computational cost, and storing the anchor matrix requires $\mathcal{O}(Cd)$ memory. This cost is incurred only once and is negligible compared with neural network optimization.

Client-Side Eraser Training. For each sample, the structural loss computes the similarities between the feature and all C anchors, which costs $\mathcal{O}(Cd)$. The auxiliary semantic head also maps a d-dimensional feature to C classes, giving the same order of cost. The reconstruction loss requires passing the latent feature through the Generator and then through the frozen global model. Although the global model is not updated, gradients are back-propagated through it to optimize the Generator. Therefore, the local computational cost on client i in one Eraser round is:

$$\mathcal{O}\Big(In_i\big(\Gamma_F + \Gamma_R + \Gamma_G + Cd\big)\Big). \tag{24}$$

For all selected clients in one round, the cost becomes:

$$\mathcal{O}\Big(I\sum_{i\in\mathcal{S}_e} n_i\big(\Gamma_F + \Gamma_R + \Gamma_G + Cd\big)\Big). \tag{25}$$

Across E Eraser rounds, the total client-side cost is:

$$\mathcal{O}\Big(EI\sum_{i\in\mathcal{S}_e} n_i\big(\Gamma_F + \Gamma_R + \Gamma_G + Cd\big)\Big), \tag{26}$$

or $\mathcal{O}\big(EKIn(\Gamma_F + \Gamma_R + \Gamma_G + Cd)\big)$ when each selected client has n samples. Thus, Eraser scales linearly with the number of Eraser rounds, selected clients, local epochs, and local data size.

Server-Side Aggregation. Eraser keeps the Featurizers local and only aggregates the Generators. Therefore, the server only performs a weighted average over K Generator models in each Eraser round, with computational complexity:

$$\mathcal{O}(KP_R). \tag{27}$$

Across all Eraser rounds, the total server-side aggregation cost is:

$$\mathcal{O}(EKP_R), \tag{28}$$

which is much smaller than aggregating the full global model when $P_R \ll P_G$.

Communication Cost. Since only the Generator is transmitted during Eraser training, the additional communication cost per Eraser round is:

$$\mathcal{O}(KP_R). \tag{29}$$

Algorithm 1 Training Procedure of Eraser

Require: Trained global model $\mathcal{G}$, client set $\mathcal{S}$, local datasets $\{D_i\}_{i=1}^{N}$, number of classes C, feature dimension d, Eraser rounds E, balance coefficient λ
Ensure: Local Featurizers $\{F_i\}_{i=1}^{N}$ and shared Generator $R^{(E)}$
/* Server Side: Anchor Generation */
Generate fixed hyperspherical semantic anchors $A \in \mathbb{R}^{C\times d}$
Broadcast A and frozen global model $\mathcal{G}$ to all clients
Initialize the shared Generator $R^{(0)}$
for each Eraser round $e = 0, \ldots, E-1$ **do**
/* Server Side: Generator Dispatch */
Randomly sample participating clients $\mathcal{S}_e \subseteq \mathcal{S}$
Broadcast the shared Generator $R^{(e)}$ to sampled clients
/* ClientUpdate */
for all client $i \in \mathcal{S}_e$ **in parallel do**
$R_i^{(e)} \leftarrow R^{(e)}$
Load local Featurizer F_i and auxiliary head H_i
Compute $\mathcal{L}_{\mathrm{STR}}$ on D_i with anchors A
Compute $\mathcal{L}_{\mathrm{SEM}}$ on D_i with auxiliary head H_i
Update F_i and H_i by minimizing $\mathcal{L}_{E_i} = \lambda\mathcal{L}_{\mathrm{STR}} + \mathcal{L}_{\mathrm{SEM}}$
Extract latent features $z = F_i(x)$ for local samples $(x, y) \in D_i$
Reconstruct samples $\tilde{x} = R_i^{(e)}(z)$
Compute Generator loss $\mathcal{L}_{R_i} = \mathcal{L}_{\mathrm{CE}}\big(\mathcal{G}(\tilde{x}), y\big)$
Update $R_i^{(e)}$ by minimizing $\mathcal{L}_{R_i}$
Upload updated Generator $R_i^{(e+1)}$ to the server
end for
/* Server Side: Generator Aggregation */
Aggregate uploaded Generators: $R^{(e+1)} = \sum_{i\in\mathcal{S}_e} \frac{|D_i|}{\sum_{j\in\mathcal{S}_e}|D_j|} R_i^{(e+1)}$
end for
return $\{F_i\}_{i=1}^{N}$ and $R^{(E)}$

With bidirectional communication, the total extra communication cost is $\mathcal{O}(2EKP_R)$. In contrast, standard FedAVG with T communication rounds has communication cost $\mathcal{O}(2TKP_G)$. Therefore, the relative communication cost after adding Eraser is:

$$1 + \frac{EP_R}{TP_G}. \tag{30}$$

In our implementation on CIFAR-10, $E = 50$, $T = 300$, $P_R = 0.25$ MB, and $P_G = 44.63$ MB, which gives:

$$1 + \frac{50 \times 0.25}{300 \times 44.63} \approx 1.0009. \tag{31}$$

This shows that the communication overhead of Eraser is negligible.

Inference Cost. For each input sample, Eraser performs one Featurizer forward pass, one Generator forward pass, and one global model forward pass:

$$\mathcal{O}(\gamma_F + \gamma_R + \gamma_G), \tag{32}$$

where γ_F, γ_R, and γ_G denote the corresponding forward costs. Compared with directly evaluating the global model, Eraser only adds the forward cost of the Featurizer and Generator.

We further quantify the inference cost in terms of floating-point operations under the default ResNet-18 setting. As shown in Table 30, the Featurizer, Generator, and global model require 1.1158, 0.0003, and 1.1158 GFLOPs per sample, respectively. Therefore, the complete Eraser inference pipeline requires 2.2319 GFLOPs, introducing an additional 1.1161 GFLOPs compared with direct inference using the global model.

We additionally measure the practical per-sample inference latency and peak GPU memory. We first perform 100 warm-up iterations and then report the average latency over 100 inference itera-

Table 30: Inference computational cost under the default ResNet-18 setting.

Components	Featurizer	Generator	Global Model	***Eraser Pipeline***
GFLOPs / sample	1.1158	0.0003	1.1158	**2.2319**

Table 31: Practical per-sample inference overhead under the default ResNet-18 setting.

Metric	Global Model	***Eraser + Global Model***	Overhead
Latency (ms/sample)	3.1577	**5.7539**	1.82×
Peak GPU Memory (MiB)	109.22	**109.26**	1.00×

tions using `time.perf_counter`, with CUDA synchronization applied during timing. Peak GPU memory is measured using `torch.cuda.max_memory_allocated`. As shown in Table 31, Eraser increases the inference latency from 3.1577 ms to 5.7539 ms per sample, corresponding to a $1.82\times$ overhead. In contrast, the peak GPU memory remains nearly unchanged, increasing only from 109.22 MiB to 109.26 MiB.

Table 32: GPU memory overhead compared with FedAVG under CIFAR-10.

Baselines	LockDown	FoolsGold	Multi-Metrics	SnowBall	FDCR	SPMC	MARS	AlignIns	MASA	FedDLAD	FilterFL	FedAVG + ***Eraser***
GPU memory × FedAVG	1.47	1.16	1.12	1.28	1.64	2.17	1.14	**1.03**	1.71	1.53	2.19	1.21

GPU memory Overhead. Table 32 reports the peak GPU memory overhead relative to FedAVG on CIFAR-10. Eraser incurs a moderate memory overhead of $1.21\times$ FedAVG, which is lower than several representative defenses such as LockDown ($1.47\times$), FDCR ($1.64\times$), SPMC ($2.17\times$), MASA ($1.71\times$), FedDLAD ($1.53\times$), and FilterFL ($2.19\times$). Although Eraser introduces an additional Featurizer and Generator during the post-training defense stage, its memory footprint remains modest because only the lightweight Generator is collaboratively optimized and aggregated, while the Featurizer is kept local and the global model is frozen.

Overall, the dominant cost of Eraser comes from client-side post-training optimization, whereas the server-side computation and communication costs are small. The complexity is linear in the number of selected clients and local samples, and Eraser avoids pairwise client comparison, trigger inversion, or full-model aggregation during the defense stage.

T Convergence Analysis

We provide a convergence analysis for the post-training optimization of Eraser. Since the global model G is frozen during Eraser training, the convergence analysis focuses on the client-side optimization of Eraser and the federated aggregation of the Generator. Let E denote the total number of Eraser communication rounds, and let $e \in \{0, \dots, E-1\}$ denote the current Eraser communication round. Let Q denote the number of local iterations within each Eraser round, and let $q \in \{0, \dots, Q-1\}$ denote the current local iteration. For client i, ϕ_i denotes the parameters of its local Featurizer F_i, and ψ_i denotes the parameters of its local Generator R_i. The server maintains the global Generator parameter ψ, which is sent to selected clients at the beginning of each Eraser round. The local Eraser objective of client i is written as:

$$\mathcal{J}_i(\phi_i, \psi_i) = \lambda \mathcal{L}_{\mathrm{STR}}(\phi_i) + \mathcal{L}_{\mathrm{SEM}}(\phi_i) + \mathcal{L}_{R_i}(\psi_i), \tag{33}$$

where $\mathcal{L}_{\mathrm{STR}}$ is the anchor-guided structural loss, $\mathcal{L}_{\mathrm{SEM}}$ is the semantic supervision loss, and $\mathcal{L}_{R_i}$ is the reconstruction loss supervised by the frozen global model G. For simplicity, we denote $\theta_i = (\phi_i, \psi_i)$ and $\mathcal{J}_i(\theta_i) = \mathcal{J}_i(\phi_i, \psi_i)$.

Assumption 1. ***Lipschitz Smoothness.*** *Each local Eraser objective $\mathcal{J}_i$ is L-smooth, i.e., for any θ_1, θ_2,*

$$\|\nabla \mathcal{J}_i(\theta_1) - \nabla \mathcal{J}_i(\theta_2)\| \le L \|\theta_1 - \theta_2\|. \tag{34}$$

Equivalently, we have:

$$\mathcal{J}_i(\theta_1) \le \mathcal{J}_i(\theta_2) + \langle \nabla \mathcal{J}_i(\theta_2), \theta_1 - \theta_2 \rangle + \frac{L}{2} \|\theta_1 - \theta_2\|^2. \tag{35}$$

Assumption 2. ***Unbiased Gradient and Bounded Variance.*** *The stochastic gradient* $g_i^{eQ+q} = \nabla \mathcal{J}_i(\theta_i^{eQ+q}; \xi_i^{eQ+q})$ *is an unbiased estimator of the full local gradient:*

$$\mathbb{E}_{\xi_i^{eQ+q}} \left[g_i^{eQ+q} \right] = \nabla \mathcal{J}_i(\theta_i^{eQ+q}), \tag{36}$$

and its variance is bounded by:

$$\mathbb{E} \left[\left\| g_i^{eQ+q} - \nabla \mathcal{J}_i(\theta_i^{eQ+q}) \right\|^2 \right] \leq \sigma^2. \tag{37}$$

Assumption 3. ***Bounded Generator Aggregation Deviation.*** *After local training and server aggregation, the deviation between a local Generator and the aggregated global Generator is bounded. Specifically, for any participating client* i*,*

$$\mathbb{E} \left[\left\| \nabla_\psi \mathcal{J}_i(\phi_i, \psi_i) - \nabla_\psi \mathcal{J}(\phi_i, \psi) \right\|^2 \right] \leq \delta^2, \tag{38}$$

where $\mathcal{J}$ *denotes the objective associated with the aggregated global Generator. This assumption captures the bounded drift caused by client heterogeneity during federated Generator training.*

Lemma 1. ***Local Eraser Descent.*** *Under Assumptions 1 and 2, after* Q *local iterations in the* e*-th Eraser round, the expected local objective of client* i *satisfies:*

$$\mathbb{E} \left[\mathcal{J}_i^{eQ+Q} \right] \leq \mathcal{J}_i^{eQ} - \left(\eta - \frac{L\eta^2}{2} \right) \sum_{q=0}^{Q-1} \mathbb{E} \left[\left\| \nabla \mathcal{J}_i^{eQ+q} \right\|^2 \right] + \frac{LQ\eta^2\sigma^2}{2}. \tag{39}$$

Proof. At local iteration q, client i updates its Eraser parameters by stochastic gradient descent:

$$\theta_i^{eQ+q+1} = \theta_i^{eQ+q} - \eta g_i^{eQ+q}. \tag{40}$$

By L-smoothness, we have:

$$\begin{aligned} \mathcal{J}_i^{eQ+q+1} \leq & \mathcal{J}_i^{eQ+q} + \left\langle \nabla \mathcal{J}_i^{eQ+q}, -\eta g_i^{eQ+q} \right\rangle \\ & + \frac{L\eta^2}{2} \left\| g_i^{eQ+q} \right\|^2. \end{aligned} \tag{41}$$

Taking expectation and using the unbiasedness and bounded variance in Assumption 2, we obtain:

$$\mathbb{E} \left[\mathcal{J}_i^{eQ+q+1} \right] \leq \mathcal{J}_i^{eQ+q} - \left(\eta - \frac{L\eta^2}{2} \right) \mathbb{E} \left[\left\| \nabla \mathcal{J}_i^{eQ+q} \right\|^2 \right] + \frac{L\eta^2\sigma^2}{2}. \tag{42}$$

Summing the above inequality over $q = 0, \ldots, Q-1$ gives the result. □

Lemma 2. ***Deviation After Generator Aggregation.*** *Under Assumption 3, after the server aggregates local Generators and sends the global Generator back to clients, the expected local Eraser objective is bounded by:*

$$\mathbb{E} \left[\mathcal{J}_i^{(e+1)Q} \right] \leq \mathbb{E} \left[\mathcal{J}_i^{eQ+Q} \right] + \frac{L\eta^2\delta^2}{2}. \tag{43}$$

Proof. At the end of the e-th Eraser round, client i has local parameters $(\phi_i^{e+1}, \psi_i^{e+1})$. The Featurizer remains local, while the Generator is replaced by the aggregated global Generator ψ^{e+1}. Thus, the objective change caused by aggregation is:

$$\mathcal{J}_i^{(e+1)Q} - \mathcal{J}_i^{eQ+Q} = \mathcal{J}_i(\phi_i^{e+1}, \psi^{e+1}) - \mathcal{J}_i(\phi_i^{e+1}, \psi_i^{e+1}). \tag{44}$$

By L-smoothness, we have:

$$\begin{aligned} \mathcal{J}_i(\phi_i^{e+1}, \psi^{e+1}) \leq & \mathcal{J}_i(\phi_i^{e+1}, \psi_i^{e+1}) + \left\langle \nabla_\psi \mathcal{J}_i(\phi_i^{e+1}, \psi_i^{e+1}), \psi^{e+1} - \psi_i^{e+1} \right\rangle \\ & + \frac{L}{2} \left\| \psi^{e+1} - \psi_i^{e+1} \right\|^2. \end{aligned} \tag{45}$$

Using the standard bounded generator-drift assumption, the aggregation-induced deviation can be bounded by the gradient discrepancy between the local and global Generator objectives. Taking expectation and applying Assumption 3, we obtain:

$$\mathbb{E} \left[\mathcal{J}_i^{(e+1)Q} \right] \leq \mathbb{E} \left[\mathcal{J}_i^{eQ+Q} \right] + \frac{L\eta^2\delta^2}{2}. \tag{46}$$

□

Theorem 1. ***One-round Descent of Eraser.*** *Under Assumptions 1, 2, and 3, the expected local objective after one Eraser communication round satisfies:*

$$\begin{aligned}\mathbb{E}\left[\mathcal{J}_i^{(e+1)Q}\right] \leq & \mathcal{J}_i^{eQ} - \left(\eta - \frac{L\eta^2}{2}\right) \sum_{q=0}^{Q-1} \mathbb{E}\left[\left\|\nabla \mathcal{J}_i^{eQ+q}\right\|^2\right] \\ & + \frac{L\eta^2}{2}\left(Q\sigma^2 + \delta^2\right).\end{aligned} \tag{47}$$

Proof. The result follows by substituting Lemma 1 into Lemma 2. □

Theorem 2. ***Non-convex Convergence Rate of Eraser.*** *Let $\mathcal{J}^*$ be a lower bound of the Eraser objective and define $\Delta = \mathcal{J}_i^0 - \mathcal{J}^*$. If $0 < \eta < 2/L$, then after E Eraser communication rounds, the average squared gradient norm satisfies:*

$$\frac{1}{EQ}\sum_{e=0}^{E-1}\sum_{q=0}^{Q-1}\mathbb{E}\left[\left\|\nabla \mathcal{J}_i^{eQ+q}\right\|^2\right] \leq \frac{2\Delta}{EQ\eta(2-L\eta)} + \frac{L\eta(Q\sigma^2+\delta^2)}{2-L\eta}. \tag{48}$$

Furthermore, for any $\epsilon > 0$, the above average gradient norm is smaller than ϵ when:

$$\eta < \frac{2\epsilon}{L(\epsilon + Q\sigma^2 + \delta^2)} \tag{49}$$

and E is sufficiently large.

Proof. Rearranging Theorem 1, we have:

$$\sum_{q=0}^{Q-1}\mathbb{E}\left[\left\|\nabla \mathcal{J}_i^{eQ+q}\right\|^2\right] \leq \frac{\mathbb{E}\left[\mathcal{J}_i^{eQ}\right] - \mathbb{E}\left[\mathcal{J}_i^{(e+1)Q}\right] + \frac{L\eta^2}{2}(Q\sigma^2+\delta^2)}{\eta - \frac{L\eta^2}{2}}. \tag{50}$$

Summing over $e = 0, \ldots, E-1$, we obtain:

$$\begin{aligned}&\sum_{e=0}^{E-1}\sum_{q=0}^{Q-1}\mathbb{E}\left[\left\|\nabla \mathcal{J}_i^{eQ+q}\right\|^2\right] \\ \leq & \frac{\mathbb{E}\left[\mathcal{J}_i^{0}\right] - \mathbb{E}\left[\mathcal{J}_i^{EQ}\right] + \frac{EL\eta^2}{2}(Q\sigma^2+\delta^2)}{\eta - \frac{L\eta^2}{2}}.\end{aligned} \tag{51}$$

Since $\mathcal{J}_i^{EQ} \geq \mathcal{J}^*$, we have:

$$\frac{1}{EQ}\sum_{e=0}^{E-1}\sum_{q=0}^{Q-1}\mathbb{E}\left[\left\|\nabla \mathcal{J}_i^{eQ+q}\right\|^2\right] \leq \frac{\frac{\Delta}{EQ} + \frac{L\eta^2}{2}(Q\sigma^2+\delta^2)}{\eta - \frac{L\eta^2}{2}}. \tag{52}$$

This gives:

$$\frac{1}{EQ}\sum_{e=0}^{E-1}\sum_{q=0}^{Q-1}\mathbb{E}\left[\left\|\nabla \mathcal{J}_i^{eQ+q}\right\|^2\right] \leq \frac{2\Delta}{EQ\eta(2-L\eta)} + \frac{L\eta(Q\sigma^2+\delta^2)}{2-L\eta}. \tag{53}$$

To make the right-hand side smaller than ϵ, it is sufficient to choose a learning rate satisfying:

$$\eta < \frac{2\epsilon}{L(\epsilon + Q\sigma^2 + \delta^2)}. \tag{54}$$

Thus, with a proper learning rate, Eraser converges to a stationary point in the non-convex setting. The first term decreases with E, which gives the standard convergence rate $\mathcal{O}(1/E)$. □

Therefore, under the standard smoothness, unbiased stochastic gradient, and bounded Generator aggregation deviation assumptions, Eraser training converges to a stationary point with a non-convex convergence rate of $\mathcal{O}(1/E)$. This result shows that the additional post-training Generator collaboration introduced by Eraser does not break the convergence behavior of federated optimization.