APTI-FL: ADVERSARIAL PERTURBATION GUIDED TRIGGER INVERSION FOR BACKDOOR DETECTION IN FEDERATED LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Federated learning limits the server's visibility into local training, allowing compromised clients to inject backdoors into a shared model. Existing approaches struggle to detect such malicious clients when non-IID data makes benign updates closely resemble malicious ones, or when attackers constitute the majority of participating clients. In our preliminary study, we discover a distinctive property of backdoored models: although adversarial perturbations substantially disrupt normal classification paths, a triggered input remains likely to be classified into the specified target even after the same perturbation is added. In other words, the backdoor path exhibits target retention under adversarial perturbations. Motivated by this observation, we propose APTI-FL, an adversarial attack guided federated backdoor defense. APTI-FL first generates untargeted adversarial perturbations from trusted server data and uses discrete and continuous region inversion paths to recover potential triggers with different spatial structures. It then determines a reliable attack target through multistage validation of candidate trigger paths and target correction across rounds. Finally, it combines multidimensional client behavior with historical evidence to identify clients that exhibit backdoor responses and blocks their contributions before aggregation. Experiments on five benchmark datasets against five representative defenses show that APTI-FL substantially reduces attack success rates while maintaining main task accuracy.

1 INTRODUCTION

Federated learning (FL) has become an important paradigm for jointly using distributed data under privacy constraints. A typical FL system consists of a central server and some clients. In each communication round, the server broadcasts global model to the selected clients, which train on their private data and upload the local updates to the server for aggregation (McMahan et al., 2017). Because the raw samples remain on the clients, FL has been applied to mobile services, health care, and intelligent transportation. However, this very architecture also constrains the server's visibility into the local training process. Once some clients are compromised, the attacker can insert a trigger into their local data and make the uploaded models learn a mapping from that trigger to a chosen target class (Bagdasaryan et al., 2020; Wang et al., 2020; Xie et al., 2020). A backdoored global model usually behaves normally on clean inputs and can therefore evade ordinary validation, but it predicts the attack target whenever the trigger appears. In FL, malicious clients can also inject the backdoor jointly across clients and rounds, or divide a complete trigger into several local patterns to make the attack more difficult to detect (Xie et al., 2020).

Existing defenses attempt to identify malicious clients before aggregation (Blanchard et al., 2017; Nguyen et al., 2022; Cao et al., 2021; Zhang et al., 2022; Xie et al., 2021; Miao et al., 2024), yet they face two important challenges. First, many methods regard a backdoor update as an anomaly in parameter space and identify them through norms or directional similarity. This reasoning assumes that benign updates form a relatively stable reference while malicious updates deviate from it. In practical non-IID FL, however, differences in class composition and sample size can produce large variations in the directions and magnitudes of benign updates. Such benign heterogeneity may be mistaken for an attack, whereas a constrained malicious update may fall inside the geometric range of benign updates. FilterFL also reports that similarity based defenses can degrade under strong

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.

data heterogeneity (Yang et al., 2025). Second, several defenses assume that malicious clients are a minority (Blanchard et al., 2017; Nguyen et al., 2022; Yu et al., 2026; Sui et al., 2026). But when the malicious clients dominate a participating cohort, their updates can form the main cluster and the fewer benign updates may instead appear as outliers.

These limitations suggest that statistical differences between client updates are insufficient when data are heterogeneous or the malicious client proportion is high. We therefore seek a signal that reflects the backdoor behavior itself. Our preliminary investigation reveals a stable phenomenon. For a normal classification path, an untargeted adversarial perturbation damages the discriminative information of the input and markedly changes the model prediction. For a backdoor path, however, applying the same perturbation to an input that already contains the trigger does not readily destroy the trigger to target mapping: the backdoored model still tends to predict the specified class. Thus, although adversarial perturbations interfere with normal classification, the backdoor target can remain dominant. We call this phenomenon *backdoor target retention under adversarial perturbations*. Based on this discovery, we propose APTI-FL, which uses adversarial perturbations to guide trigger inversion and expose latent backdoor paths. The server first generates untargeted perturbations on a small trusted root dataset. It then searches for candidate patterns that consistently activate the same target on both natural triggered inputs and adversarial triggered inputs. Once a candidate path is obtained, the server compares the responses to triggered probes and matched controls, and uses the candidate path to test the actual behavior of each client model. Clients that strengthen the shared backdoor response are subsequently excluded from aggregation. In this way, the detection process is driven by whether a client exhibits the learned backdoor behavior, rather than only by the relative distance between client updates.

Our contributions are summarized as follows:

- We reveal that backdoored models exhibit target retention under adversarial perturbations. When a trigger is present, adding a perturbation that disrupts normal classification does not readily remove the attacker's target response. Based on this observation, APTI-FL identifies backdoor attacks from model output behavior under adversarial perturbations.
- We design a dual path trigger inversion and candidate target validation mechanism. The two paths address discrete and continuous region triggers, respectively. Candidate paths with stable target responses are selected through independent validation, improving target localization across different trigger structures.
- We design a detection mechanism that combines client behavior with historical evidence. APTI-FL identifies suspicious clients from their responses to a candidate backdoor path and integrates current evidence with historical observations, reducing the effect of non-IID fluctuations on detection.
- We conduct extensive experiments under different attacks, data distributions, and malicious client proportions. The results on five benchmark datasets show that APTI-FL achieves a better balance between attack suppression and main task accuracy than representative federated backdoor defenses.

2 Related Work

2.1 Federated Backdoor Attacks

Research on federated backdoor attacks seeks to improve attack effectiveness and stealth while preserving main task performance. Bagdasaryan et al. (2020) showed that model replacement can amplify a malicious local update so that the backdoored model strongly affects the FedAvg result. Wang et al. (2020) exploited tail data to conceal malicious behavior. DBA divides a complete trigger into local patterns and assigns them to different malicious clients, allowing multiple clients to inject a backdoor jointly (Xie et al., 2020). Later work also studied persistence. FCBA combines multiple local trigger patterns so that the global model learns more complete backdoor information and retains the attack for longer after injection stops (Liu et al., 2024). A3FL dynamically optimizes its trigger against the changing global model to improve attack stability(Zhang et al., 2023). These developments make federated backdoors increasingly difficult to distinguish and remove.

2.2 Federated Backdoor Defenses

Many defenses detect client models at the server and remove or reduce suspicious contributions. Krum selects an update according to pairwise Euclidean distances (Blanchard et al., 2017); FLAME combines model clustering, update clipping, and noise injection (Nguyen et al., 2022); FLTrust constructs a trusted server update and reweights client updates according to directional similarity (Cao et al., 2021); and FLDetector predicts normal updates from historical global information and detects deviations from those predictions (Zhang et al., 2022). Recent methods develop more fine-grained parameter-space evidence. FDCR uses Fisher information to characterize differences in parameter importance (Huang et al., 2024), AlignIns inspects update directions at multiple granularities (Xu et al., 2025a), and G^{2}uardFL represents clients as an attributed graph and separates them through graph clustering (Yu et al., 2026).

These methods still depend, to different degrees, on stable distinctions between benign and malicious updates or client groups. However, under non-IID data natural differences in class composition can enlarge the variation among benign updates and blur their distinction from malicious updates. A systematic study by Ye et al. (2025) shows that increasing data heterogeneity and task complexity reduce the consistency among benign clients, weakening the effectiveness of existing detection mechanisms. Label-distribution skew is particularly problematic for defenses based on relative relationships among clients. To enable more effective detection, several studies examine how models behave on selected inputs. MASA identifies backdoored models from their different loss changes during individual unlearning (Xu et al., 2025b). FilterFL separates newly learned clean and backdoor knowledge and queries client models with generated samples (Yang et al., 2025). Sui et al. (2026) view a backdoored model as jointly learning a clean task and a backdoor task, and detect the resulting stability of important feature maps. These studies suggest that exploiting the intrinsic differences introduced by backdoor can provide a new basis for detection. APTI-FL starts from a different observation, the target response of a trigger path can remain stable when an adversarial perturbation disrupts the normal classification path. We use this property to guide trigger inversion and then identify clients from their responses to the recovered candidate path.

3 Preliminaries and Threat Model

3.1 Federated Learning

Consider a horizontal FL system with N clients. Client i holds a private dataset D_i containing n_i samples. In communication round t, the server selects a client set $\mathcal{C}_t$, and each selected client returns a locally trained model $F_{t,i}$. FedAvg computes $F_{t+1} = \sum_{i\in\mathcal{C}_t} a_{t,i}F_{t,i}$ with $a_{t,i} = n_i / \sum_{j\in\mathcal{C}_t} n_j$.

3.2 Adversarial Attacks

An adversarial attack adds a bounded perturbation δ to a clean input x so that a classifier F makes an incorrect prediction. We use an untargeted L_∞ attack, whose objective is $\delta^* = \arg\max_{\|\delta\|_\infty\le\epsilon} \mathcal{L}_{\mathrm{CE}}(F(x+\delta), y)$ for the true label y. FGSM performs one sign-gradient step (Goodfellow et al., 2015), whereas projected gradient descent (PGD) iteratively updates $\widetilde{\delta}^{(s+1)} = \delta^{(s)} + \eta\,\mathrm{sign}(\nabla_\delta \mathcal{L}_{\mathrm{CE}}(F(x+\delta^{(s)}), y))$ and projects it back as $\delta^{(s+1)} = \Pi_{\|\delta\|_\infty\le\epsilon}(\widetilde{\delta}^{(s+1)})$ (Madry et al., 2018). We use untargeted PGD to generate the perturbations required by APTI-FL.

3.3 Threat Model

Attack objective and capabilities. Malicious clients seek to implant a backdoor path into the global model while preserving its main task accuracy. For a poisoned model F', the desired behavior is $F'(x) = l^t$ for a triggered input $x \in D_A$, and $F'(x) = F(x)$ otherwise, where l^t is the attack target. The attacker can control multiple malicious clients, potentially more than half of the participants in a round, and can fully control their data, local optimization, and training hyperparameters while coordinating multiple malicious clients. The attacker cannot control the server or benign clients, and attacks on the communication channel are outside our scope.

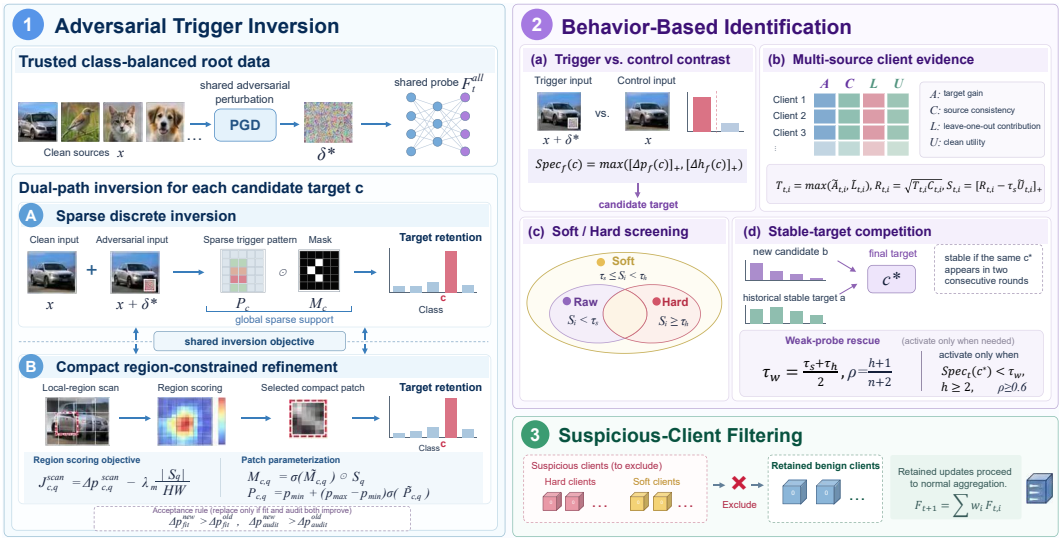


Figure 1: Overview of APTI-FL.

Defender capabilities and goals. The defense is deployed at the server. Following the trusted dataset setting used by FLTrust (Cao et al., 2021), the server holds a small clean dataset D_S. It cannot access any private client dataset or local training process and makes decisions using the received client models and the preceding global model. The defense aims to identify malicious clients accurately, reduce attack success rate (ASR), preserve main task accuracy (MTA), and remain effective across different data distributions without prior knowledge of the malicious client proportion, poisoning ratio, or trigger form.

4 APTI-FL

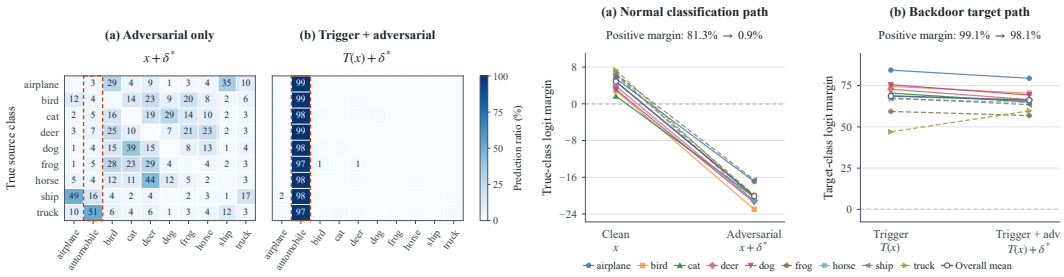
4.1 Method Overview

Untargeted adversarial attacks disrupt normal classification by maximizing the loss for the true label within a perturbation budget. A backdoor instead establishes a fixed mapping from a trigger to an attacker chosen target. This contrast raises a question: **what happens when an input contains both a backdoor trigger and an adversarial perturbation?** We generate an untargeted perturbation from a clean sample and add the same perturbation to its triggered counterpart. Figure 2(a) shows that perturbed clean inputs disperse across incorrect predictions, whereas triggered inputs remain concentrated on the attack target. Figure 2(b) further shows that the true class margin collapses while the triggered target margin remains positive. We call this phenomenon *backdoor target retention under adversarial perturbations*. It motivates using adversarial perturbations as active interference during trigger inversion: a pattern that induces the same target on natural and perturbed backgrounds is more likely to expose a backdoor path. Figure 1 summarizes APTI-FL, which combines discrete sparse and continuous region inversion, validates candidate targets on disjoint trusted data, and attributes the recovered behavior to clients before aggregation.

4.2 Adversarial Example Generation

At round t, client $i \in \mathcal{C}_t$ uploads $F_{t,i}$ with base weight $a_{t,i}$. The server first forms the temporary aggregate $F_t^{\text{all}} = \sum_i a_{t,i} F_{t,i}$. It splits the trusted root set into disjoint fit and audit subsets, then draws a class balanced working pool D_r from the fit subset. Fit samples support perturbation generation, inversion, and client analysis; audit samples are used only for validation. Appendix A defines the data roles and Appendix G gives their sizes. For each $(x_j, y_j) \in D_r$, untargeted PGD (Madry et al., 2018) approximates

$$\delta_j^* \in \arg\max_{\|\delta\|_\infty \le \epsilon} \mathcal{L}_{\text{CE}}\big(F_t^{\text{all}}(x_j + \delta), y_j\big)\,. \tag{1}$$



(a) Prediction distributions (b) Decision margins

Figure 2: Backdoor target retention under matched adversarial perturbations.

The same δ_j^* is reused on the corresponding triggered input, which creates matched natural and adversarial backgrounds.

4.3 Trigger Inversion

For each target c, we construct $D_{r,c} \subseteq D_r$ by excluding samples labeled c and sampling the remaining classes as evenly as possible. A trigger consists of pattern P_c and mask M_c, both of shape $C \times H \times W$, with $T_c(x) = (1 - M_c) \odot x + M_c \odot P_c$. Sigmoid mappings convert unconstrained variables into $M_c \in (0, 1)$ and $P_c \in (p_{\min}, p_{\max})$, allowing gradient optimization within these bounds. The temporary model is fixed throughout inversion.

Discrete sparse inversion. We seek a trigger that induces c on both natural and perturbed backgrounds. For sample $j \in D_{r,c}$, the paired loss is

$$\ell_j^{(c)} = \tfrac{1}{2}\mathcal{L}_{\mathrm{CE}}(F_t^{\mathrm{all}}(T_c(x_j)), c) + \tfrac{1}{2}\mathcal{L}_{\mathrm{CE}}(F_t^{\mathrm{all}}(T_c(x_j) + \delta_j^*), c), \tag{2}$$

$$\mathcal{L}_c^{\mathrm{mean}} = \frac{1}{|D_{r,c}|}\sum_j \ell_j^{(c)}, \qquad \mathcal{L}_c^{\mathrm{worst}} = \max_{r \in \mathcal{Y}_{c,2}} \frac{1}{|\mathcal{I}_{c,r}|}\sum_{j \in \mathcal{I}_{c,r}} \ell_j^{(c)},$$

$$\mathcal{L}_c^{\mathrm{src}} = (1 - \lambda_s)\mathcal{L}_c^{\mathrm{mean}} + \lambda_s \mathcal{L}_c^{\mathrm{worst}}. \tag{3}$$

Here $\mathcal{I}_{c,r}$ indexes samples of source class r in $D_{r,c}$, and $\mathcal{Y}_{c,2}$ contains classes with at least two samples. The worst class term discourages a solution that succeeds only on easy source classes; with fewer than two eligible classes, we use only the mean loss. The inversion objective is

$$\mathcal{L}_c^{\mathrm{inv}} = \mathcal{L}_c^{\mathrm{src}} + \lambda_m \frac{\|M_c\|_1}{CHW} + \lambda_{\mathrm{tv}}\,\mathrm{TV}(M_c). \tag{4}$$

The last two terms penalize mask area and spatial variation. After optimization, we retain the K_{pix} locations with the largest mask values averaged across channels, preserving their learned amplitudes and setting all other entries to zero. This projection sparsifies only the spatial support, while the retained mask amplitudes and pattern values remain continuous. Appendix B specifies the mappings and budget.

Continuous region inversion. Sparse projection can fragment a connected trigger. We therefore also scan square windows within the same pixel budget. A candidate $q = (u, v, s, b)$ has top left position (u, v), side length s, binary support B_q, and constant pattern $P_q \equiv b$. Its triggered input is $z_j^q = (1 - B_q) \odot x_j + B_q \odot P_q$. Each window is tested twice, filling all channels with $b = p_{\min} = -3$ and then $b = p_{\max} = 3$ in normalized input space. Let $p_c(x)$ denote the temporary model's probability of class c. On a shared scan subset of D_r, with N_c samples outside class c, we rank windows by

$$S_{c,q}^{\mathrm{scan}} = \frac{1}{N_c}\sum_{j:y_j \neq c} \frac{p_c(z_j^q) + p_c(z_j^q + \delta_j^*) - p_c(x_j) - p_c(x_j + \delta_j^*)}{2} - \lambda_m \frac{s^2}{HW}. \tag{5}$$

For the best window, $\Delta p_{\mathrm{fit},c}$ measures the same paired probability gain on $D_{r,c}$; $\Delta p_{\mathrm{audit},c}$ measures the natural triggered versus clean gain on audit samples outside class c. Refinement begins only if

both exceed the discrete trigger's gains:

$$\Delta p^{\text{scan}}_{\text{fit},c} > \Delta p^{\text{disc}}_{\text{fit},c}, \qquad \Delta p^{\text{scan}}_{\text{audit},c} > \Delta p^{\text{disc}}_{\text{audit},c}. \tag{6}$$

We then fix the window and optimize its pattern and supported mask $M^{\text{cont}}_{c,q} = \sigma(\widetilde{M}_{c,q}) \odot B_q$ with Eq. (4) on $D_{r,c}$. The refined trigger replaces the discrete one only if both fit and audit gains remain larger; otherwise, the discrete result is retained. This two stage acceptance prevents a fit specific improvement from replacing a trigger that is better supported by held out trusted data.

4.4 Candidate Target Discovery

A model may naturally favor class c, particularly under non-IID data, so a high triggered response alone does not establish a backdoor. We therefore compare matched probe and control pools from $D_{r,c}$: $\mathcal{P}_c$ contains $T_c(x_j)$ and $T_c(x_j) + \delta^*_j$, while $\mathcal{B}_c$ contains x_j and $x_j + \delta^*_j$. For model f, let $\Delta p_f(c)$ and $\Delta h_f(c)$ be the probe minus control differences in mean target probability and target prediction rate. Trigger specificity is

$$\text{Spec}_f(c) = \max([\Delta p_f(c)]_+, [\Delta h_f(c)]_+)\,. \tag{7}$$

For F^{all}_t, we normalize these scores by their maximum, sort them, and retain the group above the largest adjacent gap. We also revalidate the last stable target regardless of its current rank.

Each candidate is then checked on $D_{\text{audit},c} = \{(x, y) \in D_{\text{audit}} : y \neq c\}$, using natural inputs only. Write $g_j(c) = p_c(T_c(x_j)) - p_c(x_j)$ and let $\Delta p^{\text{audit}}_c$ be its mean. For the represented source classes $\mathcal{Y}^{\text{audit}}_c$, coverage measures how broadly the probability gain is shared:

$$C^{\text{audit}}_c = \frac{1}{|\mathcal{Y}^{\text{audit}}_c|} \sum_{r \in \mathcal{Y}^{\text{audit}}_c} \mathbb{I}\big[\text{mean}_{j:y_j=r}\, g_j(c) \geq \tau_\Delta\big]\,. \tag{8}$$

A valid candidate must have positive fit specificity and satisfy

$$|\mathcal{Y}^{\text{audit}}_c| \geq 2, \qquad \Delta p^{\text{audit}}_c > \tau_\Delta, \qquad C^{\text{audit}}_c \geq \tau_s. \tag{9}$$

We compute audit specificity by Eq. (7) on these natural probe/control pairs, and use $\text{Spec}_t(c) = \min(\text{Spec}_{\text{fit}}(c), \text{Spec}_{\text{audit}}(c))$ in target competition. Thus ranking alone cannot validate a weak trigger, and a large fit response cannot compensate for weak audit evidence. The complete statistics are defined in Appendix C.

4.5 Client Backdoor Detection

For each valid candidate, all uploaded models and the preceding global model are evaluated on the same fit probes and controls. Four signals distinguish added trigger behavior from benign variation. Direct gain is $A_{t,i}(c) = [\Delta p_{F_{t,i}}(c) - \Delta p_{F_t}(c) - \tau_\Delta]_+$, which removes the inherited global response and small fluctuations. Source consistency $C_{t,i}(c)$ is the fraction of represented source classes whose analogous client minus global gain reaches τ_Δ; it discounts effects confined to a few classes.

Leave one out support L measures how much the mean target logit margin on $\mathcal{P}_c$ decreases when client i is removed and the remaining weights are renormalized. A positive L means that client strengthens the aggregate trigger response. Clean utility U is the corresponding increase in clean cross entropy on D_r: a positive U means removing the client harms clean performance. These complementary signals measure the client's effect on both the trigger path and the main task; Appendix E gives their formulas.

We normalize A, $[L]_+$, and $[U]_+$ separately across participating clients by $\widetilde{X}_i = (X_i - \min_k X_k)/(\max_k X_k - \min_k X_k)$, setting a constant vector to zero. The fused risk and utility adjusted score are

$$R_{t,i}(c) = \sqrt{\max(\widetilde{A}_{t,i}(c), \widetilde{L}_{t,i}(c))C_{t,i}(c)}, \quad S_{t,i}(c) = [R_{t,i}(c) - \tau_s \widetilde{U}_{t,i}]_+. \tag{10}$$

Either direct gain or aggregate support can raise risk, while source consistency scales that evidence and clean utility discounts it. A client is eligible if $A > 0$, or if $L > 0$, $C > 0$, and $S \geq \tau_s$ all hold. The higher score group at the largest adjacent gap among eligible S values forms $\mathcal{R}_t(c)$. A single eligible client forms a singleton Raw set; flat scores among multiple clients yield no group, and Raw is cleared if it covers the whole cohort. History also uses a broader trace $E = \mathbb{I}[A > 0 \vee (L > 0 \wedge C > 0)]$, which records trigger evidence without requiring the current soft score threshold.

4.6 Candidate Target Determination and Aggregation

History and target competition. For each client and target, we store the largest Raw score, a Raw hit count capped at two, and the last hit round. Each valid target first *previews* this unchanged history. Soft includes current Raw clients and traced clients with sufficient combined current/history scores. Hard requires stronger current evidence or repeated historical hits. These sets distinguish evidence strength during target selection, although both are ultimately excluded. Their exact rules appear in Appendix F. Candidate quality is

$$Q_t(c) = \mathrm{Spec}_t(c)\frac{|\mathcal{H}_t(c) \cup \mathcal{S}_t(c)|}{K}. \tag{11}$$

The highest quality candidate is selected unless a validated historical target is retained. The latter is protected when its Soft coverage is no smaller and its quality is at least τ_s times the challenger's, or when its Soft coverage is strictly larger and its quality is positive. Otherwise, replacement requires higher quality and more Hard clients, with Soft counts breaking ties in Hard counts.

Final filtering. Only the chosen target commits history: stale client records expire, current Raw hits update their scores and counts, and Hard/Soft are recomputed. Two consecutive selections establish target stability, allowing current Raw clients whose soft score reaches τ_s to enter Hard. A separate memory counts observations n and simultaneous Raw/Hard confirmations h. When the target is stable but its current *fit* specificity falls below $(\tau_s + \tau_h)/2$, a client outside the union can be added to the exclusion set if $h \geq 2$ and $(h+1)/(n+2) \geq \rho_{\min}$. Counts are read before the current observation, so this step recovers previously confirmed suspicious clients without letting the current decision confirm itself. Appendix F specifies the history horizon and capacity limits.

With the final exclusion set $\mathcal{U}_t = \mathcal{H}_t \cup \mathcal{S}_t$, flagged clients receive zero weight and the remaining weights are renormalized:

$$\widetilde{a}_{t,i} = \frac{a_{t,i}\mathbb{I}[i \notin \mathcal{U}_t]}{\sum_j a_{t,j}\mathbb{I}[j \notin \mathcal{U}_t]}, \qquad F_{t+1} = \sum_i \widetilde{a}_{t,i}F_{t,i}. \tag{12}$$

If every client is flagged, the server keeps F_t; if no valid target or client evidence exists, it applies FedAvg. Algorithm 1 summarizes the procedure.

5 Experiments

5.1 Experimental Setup

We evaluate Fashion-MNIST, EMNIST, CIFAR-10, CIFAR-100, and GTSRB under centralized (CBA) and distributed (DBA) backdoor attacks (Xie et al., 2020). Training lasts 170 rounds with poison sample ratio 0.3 and target label 2. Comparisons include FedAvg (McMahan et al., 2017), Snowball (Qin et al., 2024), FLAME (Nguyen et al., 2022), CRFL (Xie et al., 2021), RFed (Miao et al., 2024), and MARS (Wan et al., 2025). We report main task accuracy (MTA), attack success rate (ASR), and, for APTI-FL, detection TPR/FPR. Every table entry and plotted point is the arithmetic mean over rounds 161 to 170. Appendix G specifies the implementation and protocol.

5.2 IID and Non-IID Comparisons

Tables 1 and 2 report each dataset separately because their attainable accuracies differ. MTA measures retained clean utility and ASR measures residual attack behavior.

IID results. APTI-FL obtains mean ASR of 1.0% under CBA and 1.4% under DBA across the five datasets, while the six baselines generally remain near full attack success. The largest APTI-FL ASR is 3.7% on CIFAR-10 under DBA. Its MTA is on average 1.2 points below FedAvg and differs by at most 3.8 points, showing that the ASR reduction is not caused by destroying clean accuracy. Detection is also selective: across the ten combinations of dataset and attack, mean TPR is 98.2% and mean FPR is 0.6%.

Table 1: IID, $\alpha = 1$, malicious ratio 0.5: MTA/ASR (%).

Method	F-MNIST	EMNIST	C10	C100	GTSRB
CBA					
FedAvg	90.4/99.6	84.8/100.0	82.3/98.9	50.2/98.3	95.7/99.6
Snowball	90.2/99.7	84.5/100.0	79.1/99.3	48.4/98.9	95.5/99.7
FLAME	89.6/99.7	84.2/100.0	78.8/99.5	44.6/99.3	96.0/98.2
CRFL	78.7/99.1	68.3/99.7	41.8/97.7	23.3/97.4	31.7/99.3
RFed	90.5/99.6	84.8/100.0	82.2/98.9	50.6/98.0	95.9/99.5
MARS	90.4/99.6	84.8/100.0	82.5/99.1	51.2/97.9	95.9/99.5
APTI-FL	**90.3/0.9**	**84.5/0.3**	**80.4/3.1**	**47.4/0.5**	**94.5/0.4**
DBA					
FedAvg	90.5/99.2	84.8/99.8	82.0/95.4	50.9/95.2	95.7/98.6
Snowball	90.3/98.8	84.3/99.8	79.4/96.3	48.6/97.2	95.5/98.4
FLAME	89.9/99.4	84.2/99.9	78.5/97.5	44.4/97.9	95.0/89.5
CRFL	77.4/98.7	67.6/99.4	45.1/97.0	20.5/96.4	34.5/99.4
RFed	90.5/99.2	84.8/99.9	81.9/95.6	51.3/95.6	95.0/98.7
MARS	90.6/99.3	84.8/99.8	81.9/96.1	50.9/96.5	95.2/98.5
APTI-FL	**90.3/1.2**	**84.4/0.4**	**81.4/3.7**	**47.2/1.5**	**95.0/0.4**

Table 2: Non-IID, $\alpha = 0.55$, malicious ratio 0.6: MTA/ASR (%).

Method	F-MNIST	EMNIST	C10	C100	GTSRB
CBA					
FedAvg	88.6/99.9	83.6/100.0	75.0/99.5	48.1/99.1	95.0/99.4
Snowball	87.3/99.8	83.3/100.0	66.4/99.3	43.6/99.4	94.9/99.4
FLAME	87.7/99.8	82.2/100.0	67.9/99.4	40.8/99.6	95.1/98.7
CRFL	78.1/99.7	68.9/99.8	28.0/99.3	18.2/98.4	24.0/99.7
RFed	88.6/99.9	83.6/100.0	74.5/99.6	47.7/99.1	95.4/99.7
MARS	88.7/99.9	83.6/100.0	75.5/99.5	48.3/99.3	94.9/99.6
APTI-FL	**87.2/3.7**	**83.0/0.4**	**70.0/5.0**	**41.5/0.7**	**94.0/0.5**
DBA					
FedAvg	88.5/99.1	83.7/99.9	74.4/99.8	46.8/97.3	94.2/99.8
Snowball	87.3/99.0	83.3/99.8	68.6/99.5	43.1/98.1	94.5/99.9
FLAME	87.4/98.9	81.9/99.9	67.4/99.6	40.2/98.8	94.3/99.7
CRFL	76.9/98.2	67.8/99.9	33.1/99.7	16.7/97.2	27.1/100.0
RFed	88.7/99.0	83.8/99.9	74.0/99.7	46.9/97.7	94.5/99.9
MARS	88.6/99.1	83.7/99.9	74.8/99.7	46.4/97.3	94.4/99.9
APTI-FL	**87.2/2.6**	**83.0/0.5**	**70.0/5.6**	**42.9/0.8**	**93.3/0.7**

Non-IID results. With $\alpha = 0.55$ and 60% malicious clients, APTI-FL keeps mean ASR at 2.0% for both attacks; the worst case is 5.6% on CIFAR-10 under DBA. The competing methods remain between 97.2% and 100.0% ASR in this table. Relative to FedAvg, APTI-FL reduces ASR by 97.4 points on average while giving up 2.6 MTA points. Its mean TPR/FPR are $98.6\%/1.9\%$. CRFL's severe MTA degradation on CIFAR-10, CIFAR-100, and GTSRB, with ASR remaining near 100%, shows that sacrificing clean accuracy does not by itself constitute effective backdoor defense.

5.3 Robustness to Data Heterogeneity and Malicious Participation

Figure 3 varies $\alpha \in \{0.35, 0.55, 0.75, 0.95\}$ at malicious ratio 0.5. Results are reported separately for each dataset.

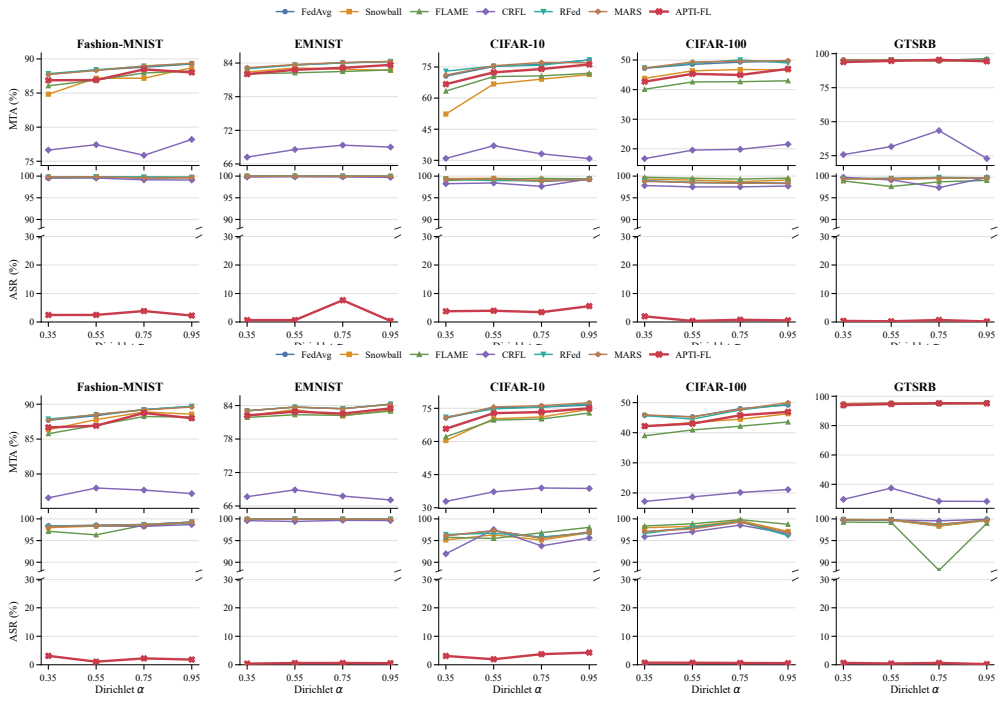


Figure 3: Data heterogeneity at malicious ratio 0.5. Top: MTA; bottom: ASR with a broken axis.

Across all heterogeneity settings, APTI-FL keeps mean ASR at 1.8% with average FPR below 3%, while its MTA remains within 1.7 points of FedAvg. Performance is stable as label skew changes; the isolated increase on EMNIST CBA at $\alpha = 0.75$ does not form a systematic trend. The recovered trigger response therefore remains separable across different data distributions.

Table 3 confirms that low ASR is supported by high recall of malicious clients rather than broad rejection: TPR stays above 97% and mean FPR remains below 3.4% for every summarized combination of study and attack.

Table 3: Mean APTI-FL detection TPR/FPR (%) across each study.

Study	CBA	DBA
IID, ratio 0.5	99.1/0.4	97.4/0.7
Non-IID, $\alpha = 0.55$, ratio 0.6	98.8/1.0	98.4/2.9
Heterogeneity sweep, ratio 0.5	98.3/2.0	97.5/3.3
IID ratio sweep	98.7/0.5	97.6/0.7

5.4 Malicious Participation

Figure 4 varies the malicious ratio under IID data. CBA and DBA are evaluated separately.

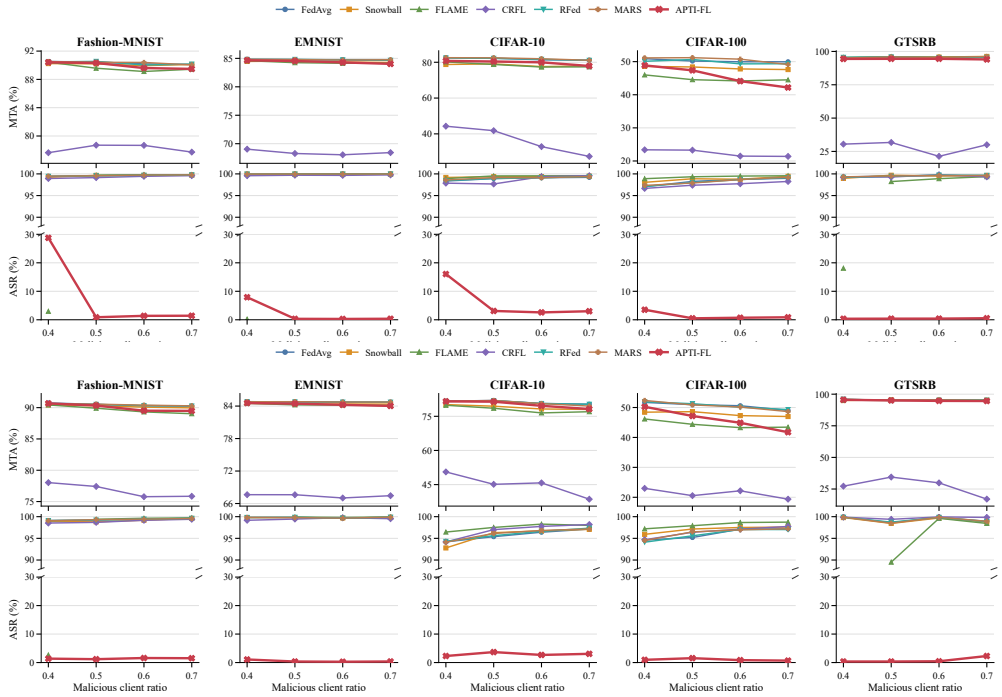


Figure 4: Malicious client ratio under IID data. Top: MTA; bottom: ASR with a broken axis.

APTI-FL remains effective over most IID malicious ratios, keeping ASR below 3.7% from ratio 0.5 to 0.7. CBA at ratio 0.4 is harder because fewer attackers leave a weaker aggregate trigger signature, whereas DBA remains consistently low. MTA stays close to FedAvg on most datasets, with the larger gap concentrated on CIFAR-100 as more updates are filtered. The non-IID sweep in Appendix H shows the same overall robustness, with the only exception being a slight ASR increase on CIFAR-10 under DBA at ratio 0.7.

6 Conclusion

APTI-FL leverages the distinct behavior of backdoor paths under adversarial perturbations to guide trigger inversion and client attribution. By combining sparse and region constrained trigger recovery, independent validation, and multiple client behavior signals, APTI-FL identifies malicious updates associated with backdoor behavior and removes them from aggregation without requiring prior knowledge of the attack target or trigger. Extensive experiments under diverse and challenging settings show that APTI-FL substantially suppresses ASR while maintaining competitive MTA.

AI USE STATEMENT

Generative AI was used for translation and language polishing. The authors reviewed and verified the entire paper and take full responsibility for all content in the manuscript.

REPRODUCIBILITY STATEMENT

To support reproducibility, we submit the complete runnable source code at `https://anonymous.4open.science/r/APTI_FL-1F8B/`, together with the instructions and configurations required to reproduce the reported experiments. The method, implementation details, data roles, hyperparameters, and evaluation protocol are documented in Section 4, Section 5, and Appendix G.

REFERENCES

E. Bagdasaryan, A. Veit, Y. Hua, et al. How To Backdoor Federated Learning. In *AISTATS*, 2020.

P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer. Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent. In *NeurIPS*, 2017.

X. Cao, M. Fang, J. Liu, and N. Z. Gong. FLTrust: Byzantine-Robust Federated Learning via Trust Bootstrapping. In *NDSS*, 2021.

I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and Harnessing Adversarial Examples. In *International Conference on Learning Representations (ICLR)*, 2015.

W. Huang, M. Ye, Z. Shi, et al. Parameter Disparities Dissection for Backdoor Defense in Heterogeneous Federated Learning. In *NeurIPS*, 2024.

T. Liu, Y. Zhang, Z. Feng, et al. Beyond Traditional Threats: A Persistent Backdoor Attack on Federated Learning. In *AAAI*, 2024.

A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards Deep Learning Models Resistant to Adversarial Attacks. In *International Conference on Learning Representations (ICLR)*, 2018.

B. McMahan, E. Moore, D. Ramage, et al. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *AISTATS*, 2017.

Y. Miao, X. Yan, X. Li, et al. RFed: Robustness-Enhanced Privacy-Preserving Federated Learning Against Poisoning Attack. *IEEE Transactions on Information Forensics and Security*, 19:5814–5827, 2024.

T. D. Nguyen, P. Rieger, H. Chen, et al. FLAME: Taming Backdoors in Federated Learning. In *USENIX Security*, pp. 1415–1432, 2022.

Z. Qin, F. Chen, C. Zhi, X. Yan, and S. Deng. Resisting Backdoor Attacks in Federated Learning via Bidirectional Elections and Individual Perspective. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(13):14677–14685, 2024.

Y. Sui, Y. Sun, N. Chen, et al. Backdoor Detection in Federated Learning With Feature Map: A Multi-Task Learning Perspective. *IEEE Transactions on Information Forensics and Security*, 2026.

W. Wan, Y. Ning, Z. Huang, et al. MARS: A Malignity-Aware Backdoor Defense in Federated Learning. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

H. Wang, K. Sreenivasan, S. Rajput, et al. Attack of the Tails: Yes, You Really Can Backdoor Federated Learning. In *NeurIPS*, 2020.

C. Xie, K. Huang, P. Y. Chen, and B. Li. DBA: Distributed Backdoor Attacks against Federated Learning. In *ICLR*, 2020.

C. Xie, M. Chen, P. Y. Chen, and B. Li. CRFL: Certifiably Robust Federated Learning against Backdoor Attacks. In *ICML*, volume 139, pp. 11372–11382, 2021.

J. Xu, Z. Zhang, and R. Hu. Detecting Backdoor Attacks in Federated Learning via Direction Alignment Inspection. In *CVPR*, 2025a.

J. Xu, Z. Zhang, and R. Hu. Identify Backdoored Model in Federated Learning via Individual Unlearning. In *WACV*, 2025b.

Y. Yang, M. Hu, X. Xie, et al. FilterFL: Knowledge Filtering-based Data-Free Backdoor Defense for Federated Learning. In *ACM CCS*, 2025.

P. Ye, Y. Li, K. He, et al. Breaking the Illusion: A Critical Study of Backdoor Defense in Federated Learning With Non-IID Data. *IEEE Transactions on Information Forensics and Security*, 2025.

H. Yu, C. Ma, M. Liu, et al. G^{2}uardFL: Safeguarding Federated Learning Against Backdoor Attacks via Attributed Client Graph Clustering. *IEEE Transactions on Information Forensics and Security*, 2026.

H. Zhang, J. Jia, J. Chen, L. Lin, and D. Wu. A3FL: Adversarially Adaptive Backdoor Attacks to Federated Learning. In *NeurIPS*, 2023.

Z. Zhang, X. Cao, J. Jia, and N. Z. Gong. FLDetector: Defending Federated Learning against Model Poisoning Attacks via Detecting Malicious Clients. In *KDD*, 2022.

A Notation and Auxiliary Definitions

Table 4 collects symbols that are used repeatedly in the main text. The following sections define the auxiliary quantities and decision rules.

Table 4: Core notation.

Symbol	Meaning
$F_t, F_{t,i}, F_t^{\text{all}}$	global, uploaded, and temporary aggregate models
$D_S, D_{\text{fit}}, D_{\text{audit}}, D_r$	root, fit, audit, and working data
$D_{r,c}, D_{\text{audit},c}$	working and audit samples outside target class c
T_c, P_c, M_c	candidate trigger transform, pattern, and mask for target c
$\mathcal{P}_c, \mathcal{B}_c$	paired trigger probe and trigger free control sets
A, C, L, U	direct gain, source consistency, leave one out support, clean utility
R, S	fused trigger risk and utility adjusted score
$\mathcal{R}_t, \mathcal{S}_t, \mathcal{H}_t$	current high score, moderate, and strong evidence sets
Spec_t, Q_t	validated trigger specificity and candidate quality
K	number of clients selected in the current round

B Trigger Inversion Details

Data and parameterization. For target c, $D_{r,c}$ contains at most 32 examples selected from D_r by cycling through represented classes other than c. The paired objective in Eqs. (2)–(4) uses these examples and their fixed PGD perturbations. The source class mean for r averages over $\mathcal{I}_{c,r} = \{j \in D_{r,c} : y_j = r\}$. Only classes with at least two examples contribute to the worst class term, and at least two such classes are required to use that term. The variable mappings are

$$M_c = \sigma(\widetilde{M}_c), \qquad P_c = p_{\min} + (p_{\max} - p_{\min})\sigma(\widetilde{P}_c). \tag{13}$$

Total variation is the sum of mean absolute mask differences between adjacent rows and adjacent columns. The retained spatial budget is

$$K_{\text{pix}} = \min(HW, 20, \max(4, \lceil 0.018HW \rceil)) \,. \tag{14}$$

We rank locations by the mask averaged across channels and multiply the learned mask by the binary support of the top K_{pix} locations, preserving the learned amplitudes and pattern values.

Window scan and refinement. A shared scan pool contains at most eight examples chosen cyclically across classes in D_r. For each target, only its non target examples enter Eq. (5); N_c is their count. We scan squares of side $s = 2, \ldots, \lfloor \sqrt{K_{\text{pix}}} \rfloor$ with stride $\max(1, \lfloor s/2 \rfloor)$, including the final row and column positions. For the image sizes used here, the sides are 2, 3, and 4. A window support is broadcast across channels, and each position is tested with uniform fill -3 and 3 separately. These are pattern bounds in normalized input space, not raw pixel intensities.

The best scan window for each target is evaluated again on $D_{r,c}$ and $D_{\text{audit},c}$. Its fit probability gain averages the triggered minus control differences over both natural and perturbed inputs; its audit gain uses natural inputs only. Both are compared with the discrete sparse trigger on the same respective data. Only a strict improvement on both sets permits refinement. For an admitted support B_q, the mask is $\sigma(\widetilde{M}_{c,q}) \odot B_q$ with $\widetilde{M}_{c,q} = 4$ initially. The unconstrained pattern starts at -4 or 4, according to the chosen endpoint, so the sigmoid mapping initializes it near that endpoint. We optimize Eq. (4) for 30 steps on $D_{r,c}$, with the temporary model fixed. Replacement again requires strict improvement over the discrete trigger on both sets. Specificity is recomputed for the resulting trigger. Audit examples influence these selection decisions but never the optimization gradients.

C Target Validation Details

For a model f, let $p_f(c \mid x)$ be its target probability and $h_f(c \mid x) = \mathbb{I}[\arg\max_k f_k(x) = c]$. On a paired fit probe/control pool,

$$\Delta p_f(c) = \text{mean}_{z \in \mathcal{P}_c} p_f(c \mid z) - \text{mean}_{x \in \mathcal{B}_c} p_f(c \mid x), \qquad \Delta h_f(c) = \bar{h}_f^{\mathcal{P}_c}(c) - \bar{h}_f^{\mathcal{B}_c}(c). \tag{15}$$

Each pool contains two versions of every example in $D_{r,c}$, with and without the same perturbation. Bars in Δh denote the corresponding mean prediction rates. We set $\mathrm{Spec}_{\mathrm{fit}}(c) = \max(0, \Delta p_{F_t^{\mathrm{all}}}(c), \Delta h_{F_t^{\mathrm{all}}}(c))$. Dividing by the largest target specificity yields normalized scores $\pi_{t,(1)} \geq \cdots$. The index maximizing $\pi_{t,(j)} - \pi_{t,(j+1)}$ separates the candidate group above the gap. A stable historical target is included for revalidation even when absent from this group.

For $f = F_t^{\mathrm{all}}$ and $D_{\mathrm{audit},c} = \{(x, y) \in D_{\mathrm{audit}} : y \neq c\}$, the audit statistics are

$$\Delta p_c^{\mathrm{audit}} = \frac{1}{|D_{\mathrm{audit},c}|} \sum_{j \in D_{\mathrm{audit},c}} \big[p_f(c \mid T_c(x_j)) - p_f(c \mid x_j)\big], \tag{16}$$

$$\Delta h_c^{\mathrm{audit}} = \frac{1}{|D_{\mathrm{audit},c}|} \sum_{j \in D_{\mathrm{audit},c}} \big[h_f(c \mid T_c(x_j)) - h_f(c \mid x_j)\big], \tag{17}$$

$$\mathrm{Spec}_{\mathrm{audit}}(c) = \max(0, \Delta p_c^{\mathrm{audit}}, \Delta h_c^{\mathrm{audit}}), \qquad \mathrm{Spec}_t(c) = \min(\mathrm{Spec}_{\mathrm{fit}}(c), \mathrm{Spec}_{\mathrm{audit}}(c)). \tag{18}$$

Coverage in Eq. (8) counts represented source classes whose mean probability gain reaches τ_Δ. Positive fit specificity and all three conditions in Eq. (9) are required for client analysis. The minimum specificity enters target quality in Eq. (11); the historical exclusion rule instead uses fit specificity, as specified in Appendix F.

D Complete APTI-FL Procedure

The algorithm uses trusted labels, uploaded models, and historical detection state. It does not consume the true attack target, trigger, or identities of malicious clients.

Algorithm 1 APTI-FL at server round t

Require: Global model F_t, uploaded models $\{F_{t,i}\}$, base weights $a_{t,i}$, root data D_S, history

Form temporary aggregate F_t^{all}; sample disjoint D_{fit} and D_{audit}.
Select working pool $D_r \subseteq D_{\mathrm{fit}}$; generate matched PGD perturbations on D_r.
for each class c **do**
    Select $D_{r,c}$ approximately evenly across represented classes other than c; optimize the discrete sparse trigger.
    Project its mask to the top K_{pix} spatial positions.
    Scan each window with the constant values $p_{\min}$ and $p_{\max}$ separately; rank all candidates by the scan score.
    if best window improves probability gain on both $D_{r,c}$ and $D_{\mathrm{audit},c}$ **then**
        Refine pattern and supported mask on $D_{r,c}$ only.
        Replace the discrete trigger only if both gains still improve strictly.
    end if
end for
Rank fit probe specificity by the largest gap; add the stable historical target, if any.
for each candidate target c **do**
    Audit natural trigger/control gain and source coverage on $D_{\mathrm{audit},c}$.
    if candidate is valid **then**
        Compute A, C, L, U, score S, and Raw clients using probes from fit data.
        Preview Hard/Soft from unchanged history; omit promotion from target stability.
        Set $Q_t(c) = \min(\mathrm{Spec}_{\mathrm{fit}}, \mathrm{Spec}_{\mathrm{audit}})|\mathcal{H}_t(c) \cup \mathcal{S}_t(c)|/K$.
    end if
end for
Select target using current quality and protection or correction of the historical target.
Update target streak; expire/update client history for the selected target only.
Recompute Hard/Soft, allowing promotion from target stability; recover historical exclusions when fit specificity is weak.
Set all weights in $\mathcal{H}_t \cup \mathcal{S}_t$ to zero; normalize the remaining base weights.
Aggregate retained updates; keep F_t if all weights vanish.
Use ordinary FedAvg if target or client evidence is unavailable.

E CLIENT EVIDENCE DETAILS

Let $d_{t,i}(c) = \Delta p_{F_{t,i}}(c) - \Delta p_{F_t}(c)$ be the trigger gain added beyond the inherited global response. For source class r, define $d_{t,i,r}$ analogously. Direct evidence and source consistency are

$$A_{t,i}(c) = [d_{t,i}(c) - \tau_\Delta]_+, \qquad C_{t,i}(c) = \frac{1}{|\mathcal{Y}_c|} \sum_{r \in \mathcal{Y}_c} \mathbb{I}[d_{t,i,r}(c) \geq \tau_\Delta], \tag{19}$$

Here $\mathcal{Y}_c$ contains the represented classes other than c, and source statistics use their paired natural and perturbed examples. We set $C = 0$ if fewer than two source classes are represented. Let F_t^{-i} be the temporary aggregate without client i, with the remaining weights renormalized. With logit margin $m_f(x, c) = f_c(x) - \max_{r \neq c} f_r(x)$,

$$L_{t,i}(c) = \overline{m}^{\mathcal{P}_c}_{F_t^{\text{all}}}(c) - \overline{m}^{\mathcal{P}_c}_{F_t^{-i}}(c), \tag{20}$$

$$U_{t,i} = \mathcal{L}_{D_r}(F_t^{-i}) - \mathcal{L}_{D_r}(F_t^{\text{all}}). \tag{21}$$

For $X \in \{A(c), [L(c)]_+, [U]_+\}$, normalization across clients is

$$\widetilde{X}_{t,i} = \frac{X_{t,i} - \min_k X_{t,k}}{\max_k X_{t,k} - \min_k X_{t,k}}, \tag{22}$$

with value zero when the range is numerically zero. Here $\mathcal{L}_{D_r}$ is the mean clean cross entropy on D_r, and $\overline{m}^{\mathcal{P}_c}$ averages margins over both versions of the probes. Eligibility is defined by

$$\mathcal{E}_t(c) = \{i : A_{t,i}(c) > 0 \ \vee \ (L_{t,i}(c) > 0 \wedge C_{t,i}(c) > 0 \wedge S_{t,i}(c) \geq \tau_s)\}. \tag{23}$$

The Raw set is the higher group produced by the largest adjacent gap in S over $\mathcal{E}_t(c)$. The broader trace is $E_{t,i}(c) = \mathbb{I}[A_{t,i}(c) > 0 \vee (L_{t,i}(c) > 0 \wedge C_{t,i}(c) > 0)]$.

F HISTORY AND BOUNDARY CASES

Evidence tiers and target competition. Let $H^{\text{score}}_{i,c}$ be the stored maximum Raw score and $H^{\text{count}}_{i,c}$ the capped hit count. Current and historical evidence are combined as

$$\overline{S}^{\text{soft}}_{i,c} = \max(S_{t,i}(c), \tau_s H^{\text{score}}_{i,c}), \qquad \overline{S}^{\text{hard}}_{i,c} = \max(S_{t,i}(c), \tau_h H^{\text{score}}_{i,c}). \tag{24}$$

Soft contains every Raw client, together with clients whose broader trace $E = 1$ and soft score reaches τ_s. Hard requires a Raw hit with hard score at least τ_h, or two recent hits with hard score at least τ_s; a stable target also promotes a Raw client that reaches the soft threshold. Before historical recovery, Hard is capped at $K - 1$ by retaining the largest hard scores. Preview Hard need not be a subset of Soft: repeated history can satisfy Hard without a current Raw hit or trigger trace. Let b maximize Q_t, let a be the revalidated stable target with positive quality, and write $h_c = |\mathcal{H}_t(c)|$ and $s_c = |\mathcal{S}_t(c)|$. The stable target is protected when

$$(s_a \geq s_b \wedge Q_t(a) \geq \tau_s Q_t(b)) \ \vee \ (s_a > s_b \wedge Q_t(a) > 0). \tag{25}$$

Otherwise, b replaces it only if $Q_t(b) > Q_t(a)$ and either $h_b > h_a$, or $h_b = h_a$ and $s_b > s_a$. If neither protection nor replacement applies, a is retained. Without a revalidated historical target of positive quality, the current maximizer is selected directly.

Preview versus commit. Candidate previews read the stored score and count directly; they neither expire records nor record new Raw hits. Expiration is applied when the selected target is committed. For each participating client, a last hit more than $T_{\text{sample}} = \lceil N/K \rceil$ rounds ago clears the effective score and count before the current Raw hit is processed. On a hit, the score becomes $\max(H^{\text{score}}_{i,c}, S_{t,i}(c))$, the count becomes $\min(2, H^{\text{count}}_{i,c} + 1)$, and the last hit round becomes t. Thus preview histories can include records that the ensuing commit expires. Promotion from target stability is also restricted to the final pass. These distinctions explain why preview Hard/Soft membership must not be conflated with the final set assigned zero weight.

Recovery of historical exclusions. This memory is separate from the capped, expiring Raw history. For each client and selected target, it stores cumulative observations n and simultaneous Raw/Hard confirmations h. A client outside the current union is eligible for renewed exclusion only if

$$\text{target stable}, \qquad \mathrm{Spec}_{\mathrm{fit}}(c) < \frac{\tau_s + \tau_h}{2}, \qquad h \geq 2, \qquad \frac{h+1}{n+2} \geq \rho_{\min}. \tag{26}$$

The counts are read before the current observation. Eligible clients are ranked in descending order of smoothed rate, then confirmation count and client ID. At most $\max(0, K-1-|\mathcal{H}_t|)$ are added to Hard. The final Soft set includes Hard as well as its existing members, so every recovered client is excluded. Each participating client's n increases by one, whereas h increases only for membership in Raw and Hard before recovery. Thus the recovery decision cannot create its own confirmation. The Hard cap alone does not guarantee a retained client, because the union can still cover the cohort; in that case, the global model remains unchanged.

Insufficient evidence. The largest gap rule returns no group for an empty or numerically flat input with multiple elements, but keeps a single eligible element. Raw is cleared if it includes the whole cohort. A degenerate trigger has no positive probability or prediction rate lift. Audit validation is bypassed if fewer than eight audit samples are available, and local restarts are also skipped; fewer than two audit samples outside the target class invalidate that candidate's audit. The standard setting uses 64 audit samples. Fewer than eight working pool samples, no valid target, or no final client evidence causes FedAvg fallback, which provides no defense for that round.

G Implementation and Hyperparameters

The setup uses Python 3.10.20, PyTorch 2.1.1+cu118, and NumPy 1.26.4 on a Windows workstation with a 13th generation Intel CPU, 32 GB memory, and an NVIDIA GeForce RTX 4070S. Fashion-MNIST and EMNIST use two convolutional and two fully connected layers; RGB datasets use six convolutional layers and one fully connected layer. The protocol uses five local epochs and initial learning rate 0.01, with batches of 128 for the grayscale datasets and 10 for RGB datasets. CBA trains each malicious client with the complete 3×3 trigger at the lower right; DBA assigns one of three local pixel groups to each malicious client.

Data roles. The server reserves 250 trusted root samples. GTSRB uses class quotas, while the other datasets use random sampling; subsequent working pools are sampled across the classes represented in the fit subset. At each round, the root set is shuffled once; the first 96 samples form the fit subset and the next 64 form the audit subset, so the two subsets are disjoint within the round. Up to 64 fit samples are further selected to construct D_r. Within each class, samples are prioritized by

$$p_y + 0.5\mathbb{I}[\hat{y} = y] + 0.25\mathbb{I}[\hat{y} = y \wedge p_y \geq 0.55] + 0.05 p_{\max},$$

with a quota of $\max(2, \lceil 64/|\mathcal{Y}| \rceil)$ for each class. Correct prediction affects priority rather than admission. Audit data are excluded from gradient optimization and clean utility estimation.

Table 5: Data roles and round limits.

Data	Size / source	Use
D_S	250 trusted samples	Persistent root dataset
D_{fit}	First 96 sampled root examples	Source of the working pool
$D_r \subseteq D_{\mathrm{fit}}$	Up to 64 examples	PGD, detection, clean utility
$D_{r,c} \subseteq D_r$	Up to 32 examples outside target class	Inversion and paired probes
Scan pool $\subseteq D_r$	Up to 8 examples	Shared endpoint/window scan
D_{audit}	Next 64 root examples	Validation on natural inputs only

Table 6: Method settings and decision thresholds.

Parameter	Role	Value
$\lambda_s, \lambda_m, \lambda_{\text{tv}}$	Source / area / variation loss	$0.35, 4.0, 0.03$
$\tau_s, \tau_h, \tau_\Delta$	Soft / Hard / direct gain	$0.32, 0.50, 0.04$
$\tau_{\text{weak}}, \rho_{\min}$	Rescue gate / confirmation rate	$0.41, 0.60$
ϵ / PGD steps / step size	Normalized input space	$0.15/5/0.045$
Adam step size / inversion steps	Discrete / region refinement	$0.08;\ 90/30$
$p_{\min}, p_{\max}$ / loss weights	Pattern bounds / natural/adversarial	$-3, 3;\ 0.5/0.5$
Mask budget	Fraction; minimum / maximum	$0.018;\ 4/20$
Scan sides / stride	For images from 28 to 32 pixels	$2, 3, 4$ / $\max(1, \lfloor s/2 \rfloor)$
History horizon / hit cap / stability	Raw memory / target streak	$\lceil N/K \rceil/2/2$

Decision thresholds. The direct gain threshold $\tau_\Delta = 0.04$ removes small probability fluctuations and is reused by the audit coverage test. The Soft threshold $\tau_s = 0.32$ retains moderate but consistent evidence, whereas $\tau_h = 0.50$ requires stronger evidence for Hard. Their midpoint $\tau_{\text{weak}} = (\tau_s + \tau_h)/2 = 0.41$ activates repeated confirmation only when a stable target has weak current fit specificity. With a Beta$(1, 1)$ smoothing prior, the posterior confirmation rate is $(h + 1)/(n + 2)$; the threshold $\rho_{\min} = 0.60$ requires at least 60% smoothed support together with $h \geq 2$. Candidate audit also requires at least two source classes other than the target, and target stability requires two consecutive selections.

H Additional Experimental Results

The following plots report each dataset and attack separately. ASR plots use broken axes; values outside the displayed bands are annotated.

H.1 Malicious Client Ratio under Non-IID Data ($\alpha = 0.55$)

We fix $\alpha = 0.55$ and vary the malicious client ratio from 0.4 to 0.7. Figure 5 reports results separately for each dataset and attack.

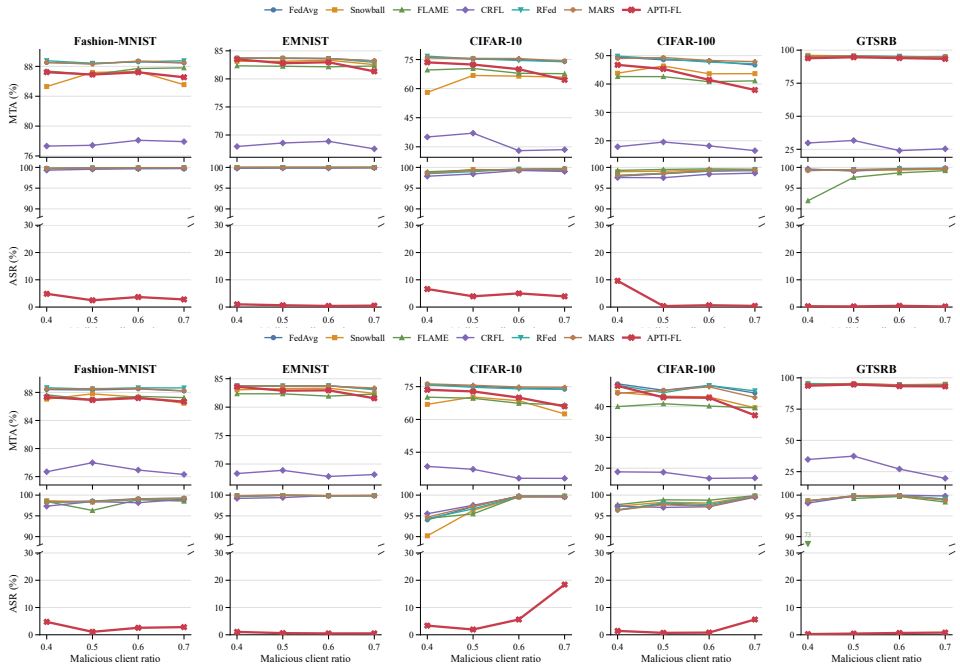


Figure 5: Malicious client ratio under non-IID data ($\alpha = 0.55$). Top: MTA; bottom: ASR with a broken axis.

For CBA, APTI-FL's mean ASR across datasets is 4.5% at ratio 0.4 and between 1.5% and 2.0% thereafter. The largest CBA value is 9.6% on CIFAR-100 at ratio 0.4. For DBA, the mean is at most 2.2% through ratio 0.6, then rises to 5.6% at ratio 0.7, mainly because CIFAR-10 reaches 18.4%. Even in that hardest case, the matched FedAvg ASR is 99.8%. Over all 40 settings, APTI-FL averages 98.1% TPR and 2.4% FPR. Its MTA is on average 2.5 points below FedAvg, with most of the larger gaps appearing on CIFAR-10 and CIFAR-100 as malicious participation grows. Thus, the method shows only a slight drop in the hardest RGB settings and performs well in most scenarios.

H.2 Hyperparameter Sensitivity

We vary one parameter at a time on non-IID CIFAR-10 under CBA ($\alpha = 0.35$, malicious ratio 0.6). Figure 6 reports the absolute metrics, and the boxes mark the default setting. The default gives 60.7% MTA, 2.2% ASR, 96.3% TPR, and 12.4% FPR. Across λ_s, λ_m, λ_{tv}, and τ_h, ASR stays from 1.7% to 5.6% and TPR from 95.1% to 96.7%. Raising τ_s to 0.40 increases ASR to 17.2%, showing that the soft gate is most sensitive. The tested $\rho_{\min}$ values give identical metrics because the historical exclusion rule selects the same clients in this run.

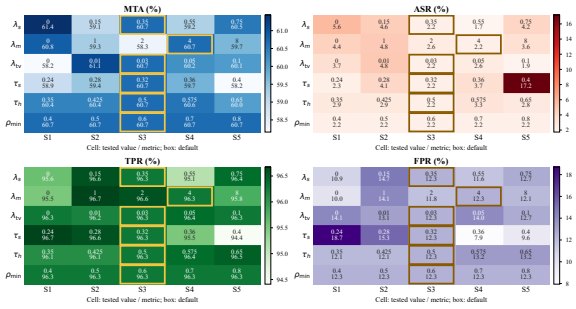


Figure 6: Hyperparameter sensitivity. Cells report metrics; boxes mark defaults.