electronics

MDPI

Article

Coordinating Direction Filtering, Reputation Aggregation, and Adaptive Differential Privacy: A Backdoor Defense Framework for Federated Object Detection in Remote Sensing

Xiaoxi Zhang [1,2], Tiegang Gao [1,*] and Yuanqing Jiang [2]

[1] College of Software, Nankai University, Tianjin Economic and Technological Development Area, Tianjin 300457, China; qweasd_0905@163.com
[2] No.208 Research Institute of China Ordnance Industries, Nankou Town, Changping District, Beijing 102202, China; 15810098180@163.com
* Correspondence: gaotiegang@nankai.edu.cn

Abstract

Federated object detection over multi-site remote-sensing imagery faces two coupled risks that are not fully addressed by existing methods. Existing federated object-detection studies mainly focus on distributed training, communication efficiency, and non-IID heterogeneity, while many robust aggregation rules rely on magnitude or coordinate statistics and can miss directional object-disappearance attacks in which the malicious update suppresses target boxes without an obvious test-time patch. Static differential privacy further applies uniform perturbation across clients, which may reduce detection utility and does not use trust differences among participants. We propose DUAL-SHIELD, a defense-and-privacy pipeline linked by three round-wise interfaces: a cosine gradient filter that detects directional anomalies with a median-absolute-deviation threshold; a momentum-based reputation aggregator that converts per-round filtering decisions into a continuous trust state; and a reputation-aware adaptive differential-privacy scheduler that allocates client-specific perturbation under explicit cumulative accounting. In the default NWPU VHR-10 object-disappearance setting, the framework keeps clean detection performance at 0.64 and reduces the attack success rate from 0.95 to 0.04. Additional transfer and attack-family evaluations are reported as supporting checks rather than definitive generalization claims, and the corresponding claims are limited to the configurations with matched-seed experiments, completed privacy accounting, and available baseline comparisons.

Keywords: federated learning; backdoor defense; adaptive differential privacy; object detection; remote sensing

Check for updates

Academic Editor: Priyadarsi Nanda

Received: 30 June 2026
Revised: 27 July 2026
Accepted: 5 August 2026
Published: 10 September 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Collaborative sensing across ground stations, low-orbit satellites, and drone swarms is reshaping how data are organized for remote-sensing object detection. High-resolution imagery is hard to pool centrally because of bandwidth limits and data-residency compliance. The federated paradigm, which exchanges only model updates and never moves the raw imagery, has therefore become a practical choice for training a global detector [1]. Remote-sensing nodes are highly heterogeneous in uplink bandwidth, compute, and class distribution. Any security or privacy mechanism must hold without assuming that clients are homogeneous. This assumption runs through every module below.

Per-round averaging supports this paradigm at low communication cost. Yet it also opens two attack surfaces that are entangled. A Byzantine minority needs no visible patch. It can implant a backdoor into the global detector through its uploaded gradients, so that target classes disappear in batches at inference time. In parallel, an honest-but-curious aggregator can invert the received gradients and reconstruct identifiable training images. The two surfaces strengthen each other in mechanism. Exposing directional information to intercept the backdoor worsens inversion leakage. A privacy layer that masks image content with noise may in turn ignore directional anomalies. Any local design that targets only one surface therefore struggles to guard both security and privacy at once.

Prior work follows three threads, but they are not yet well connected. First, recent work on federated object detection either assumes all clients are honest, or focuses on communication efficiency and natural heterogeneity rather than adversarial robustness [2–4]. Heterogeneity-optimization methods [5,6] aim at convergence and are orthogonal to adversarial security. Second, backdoor attacks have grown from early trigger patches [7] into cooperative, stealthy, and durable variants that directly manipulate the uploaded updates [8–10]. A systematic analysis [11] shows that filtering based on update magnitude alone cannot cover these diverse attack surfaces. Third, mainstream defenses still fall short. These include robust aggregation based on gradient statistics [12–15], trust bootstrapping and client-similarity reputation schemes [16,17], anomaly detection that tracks parameter drift [18,19], representation alignment with static noise layers [20,21], patch-based and test-time detection [22,23], and recent provable frameworks and out-of-distribution indicators [24–26]. Many of them base their decision on magnitude dispersion. They lose discrimination in detection settings, where bounding boxes dilute the directional signal. Others place the privacy component outside the defense decision. None feed the detection signal back to the privacy scheduler as a budget channel. Static recipes on the privacy side inject a fixed Gaussian noise into every client regardless of trust, and accumulate the budget by composition theorems. They usually cause a sizable accuracy loss [27–31].

More broadly, deep learning for intelligent perception and condition monitoring can be organized by learning paradigm and task supervision. Supervised perception models use labeled visual or sensor data to map physical states to diagnostic or detection outputs, as illustrated by machine-vision-driven physics-informed neural-network methods for pantograph-catenary state detection [32]. Unsupervised and transfer-learning methods instead address limited labels and cross-domain shifts; spatial-channel collaborative multi-scale graph-interaction transfer learning for unsupervised rotating-machinery fault diagnosis is one representative example [33]. These studies show that perception systems are moving from isolated labeled training toward cross-domain and multi-source learning. The present work differs from these centralized or transfer-learning settings by considering federated object detection, where raw remote-sensing imagery remains distributed and the learning process must simultaneously resist malicious client updates and protect private gradients.

The detection task amplifies these weaknesses. Once attack success is defined as the wholesale loss of the object box at the object level [34], the visible pixel features that classification-style defenses rely on no longer exist. Scaling amplification, distributed triggers, and rare-class hiding then press on magnitude, direction, and statistical outliers, respectively. To address this, we propose DUAL-SHIELD. It uses a feedback loop that closes every round and that explicitly decouples directional anomalies from magnitude anomalies. A cosine direction filter uses a robust median-absolute-deviation threshold to separate scaling attackers from large-step benign clients. Its binary decision settles into a continuous cross-round trust score through a momentum reputation aggregator. This score drives a reputation-aware adaptive differential-privacy scheduler. High-trust clients then receive

updates with lower noise variance under their allocated budget, while suspicious clients bear heavier noise. The privacy layer and the defense layer thus feed each other rather than running in isolation. Under the default NWPU VHR-10 setting, the framework lowers the attack success rate from 0.95 without defense to 0.04 while keeping clean detection accuracy at 0.64. It also extends the upper bound on the deployable adversarial fraction from at most 20% to at most 30%. The next three sections develop the work along this main line of direction, reputation, and privacy coupling: Sections 2–4.

2. Principle and Implementation

2.1. Preliminaries

Consider a horizontal federated system with one edge aggregator and N clients. Client j holds a local remote-sensing slice D_j. Its empirical objective is the sample average of the detection loss over this slice, written as Equation (1).

$$F_j(w) = \frac{1}{|D_j|} \sum_{\xi \in D_j} l(w; \xi) \tag{1}$$

The goal of global training is to minimize the sample-weighted sum of all local objectives, given by Equation (2). The weights reflect how much each slice contributes to the joint detector.

$$\min_w F(w) = \sum_{j=1}^{N} \frac{|D_j|}{|D|} F_j(w) \tag{2}$$

In each communication round t, the aggregator broadcasts the global parameters w(t). Client j runs several steps of stochastic gradient descent locally with learning rate η to obtain $w_j(t+1)$, then uploads the differential update $\Delta w_j(t) = w_j(t+1) - w(t)$. Average aggregation merges the local updates by sample-size weight, as in Equation (3) [35]. This upload-and-average loop is the source of the efficiency of federated learning. It is also the entry point for backdoor updates and gradient inversion, so all defense and privacy mechanisms in this paper are embedded at the aggregation end of this loop.

$$w(t+1) = \sum_{j=1}^{N} \frac{|D_j|}{|D|} w_j(t+1), w_j \leftarrow w_j - \eta \nabla F_j(w_j) \tag{3}$$

Differential privacy provides a formal guarantee by limiting the influence of a single record on the output distribution. For two neighboring datasets D and D′ that differ in only one record, a random mechanism M that satisfies the inequality in Equation (4) for any output subset S satisfies (ε,δ) differential privacy [36]. A smaller ε corresponds to stronger privacy.

$$Pr[M(D) \in S] \leq e^{\varepsilon} Pr[M(D') \in S] + \delta \tag{4}$$

A common way to achieve it is the Gaussian mechanism, which injects Gaussian noise whose standard deviation satisfies Equation (5) into a query result with sensitivity Δf. This calibration is the theoretical baseline for the adaptive noise scheduling in this paper. Unlike existing static recipes, we no longer share the same ε across all clients. Instead, we let ε vary with reputation and round, which concentrates updates with lower noise variance on trusted clients under the same privacy semantics. Table 1 summarizes the notation used throughout this paper.

$$\sigma \geq \frac{\Delta f}{\varepsilon} \sqrt{2 \ln(1.25/\delta)} \tag{5}$$

Table 1. Notation used in the DUAL-SHIELD framework. This table consolidates symbol definitions that were previously scattered across the Methods Section.

Symbol	Definition
N, j, t	number of clients, client index, and communication-round index
D_j, $w(t)$, $\Delta w_j(t)$	local dataset, global model parameters, and client update
$s_j(t)$, λ, MAD	cosine anomaly score, threshold multiplier, and median absolute deviation
$mj(t)$, $m{\sim}j(t)$, $\varphi(z)$, $\phi(mj, m{\sim}j)$	hard survival mask, soft sigmoid mask, single-input sigmoid, and two-input reputation-evidence function
$r_j(t)$, β, $\omega_j(t)$	client reputation, reputation momentum, and aggregation weight
$C_j(t)$, $\sigma_j(t)$, $\varepsilon_j(t)$, δ	clipping norm, Gaussian noise scale, allocated privacy budget, and DP failure probability
$g(r,t)$, q	bounded reputation-round budget multiplier and client participation rate

2.2. Observations and Motivation

Three groups of observation experiments characterize the weak points of pure federated aggregation under remote-sensing detection. They view the problem from three angles—privacy leakage, the cost of noise, and attack collapse—and each angle points to the design orientation of one of the three later modules.

(1) Privacy leakage of pure federated aggregation. Under average aggregation without any privacy mechanism, the uploaded client gradients alone are enough to recover training images that closely resemble the originals through improved deep-leakage inversion [37]. A more detailed gradient-inversion analysis further shows that, on a standard convolutional detection backbone, this reconstruction can reach a peak signal-to-noise ratio above 25 dB [38]. An honest-but-curious aggregator can therefore steal image content without breaking any encrypted channel. Privacy protection cannot be an optional add-on; it must be a built-in component at the aggregation end. This judgment directly motivates the design in Section 2.7, where noise is injected locally so that the aggregator never touches an un-noised gradient.

(2) Accuracy cost of static differential privacy. When differential privacy with a fixed noise scale is applied to federated detection, clean detection accuracy drops monotonically as the per-round budget tightens, and an overly tight budget can push the attack success rate back above the predefined study-specific criterion. This cost is hard to avoid, because the Gaussian mechanism scales inversely with the square of the privacy parameter. However, the uniform split of noise across clients can be improved: since clients differ in trust, allocating a limited privacy budget by trust should buy higher utility under the same privacy semantics. This is the motivation for reputation-aware budget scheduling.

(3) Collapse of undefended aggregation under the object-disappearance attack. Under this attack, undefended average aggregation collapses quickly, while classical robust aggregation based on magnitude dispersion still leaves the attack success rate far above the suspicious threshold; the quantitative collapse figures are deferred to the main comparison in Section 3.2. This gap reveals the special difficulty of the detection setting. A scaling attack moves the adversary's freedom from magnitude to direction, so a pure magnitude criterion fails, while rare-class hiding and slow drift require the defense to carry cross-round memory that single-round snapshot filtering cannot capture. This motivates a design that separates directional anomalies from magnitude anomalies and settles cross-round evidence through momentum reputation.

2.3. Threat Model and Problem Statement

The deployment scenario assumes a single edge aggregator that coordinates ten heterogeneous remote-sensing nodes over 5G or fiber backhaul. In each round, the aggregator broadcasts the global parameters; each node runs several local training steps on its private slice and uploads the differential update. Data heterogeneity is controlled by the Dirichlet concentration parameter α, with a default of 0.5 for moderate skew and smaller values describing an extreme long tail. Bandwidth heterogeneity follows a power-law distribution sampled in the 20 to 200 Mbps range; compute heterogeneity covers embedded GPU, single-board computer, and server tiers, together creating about a fivefold spread in per-round local training latency. The default detection backbone is a lightweight single-stage detector, with a two-stage detector used separately for a cross-architecture sanity check. At most 30% of the clients are Byzantine, which keeps the adversarial minority below the classical collapse point of majority-based median aggregation. Compromised clients can fully read and write their own gradients, observe the global broadcast, and collude on the trigger, the target class, and the scaling factor, but they cannot inspect the raw data of other clients or break the secure channel. The aggregator itself is honest-but-curious: it faithfully runs the aggregation algorithm, yet may use gradient inversion to infer the original imagery from the received updates. This assumption is exactly why the privacy component is indispensable.

The adversarial goal is divided into untargeted and targeted types. The untargeted type lowers global clean accuracy through scaling or distributed poisoning; the targeted type follows the object-disappearance protocol to implant a trigger patch and suppress the detection of specific classes at test time, with a default poisoning ratio of 20% of each malicious client's local pool. The defense goal is then stated as three inequalities: the clean detection accuracy under attack must be at least 0.50; the per-object attack success rate must be at most 0.20; and the privacy report must clearly separate the per-round perturbation parameter from the cross-round cumulative accounting. While meeting these inequalities, the framework must also tolerate strongly non-IID partitions and an adversarial fraction of up to 30%. These goals lead to three independent design challenges. The first is how to separate directional anomalies from magnitude anomalies, so as to expose scaling adversaries without harming large-step benign clients. The second is how to couple the reputation signal with the privacy budget, so as to support both punishment and forgiveness. The third is whether adaptive noise scheduling can keep the accuracy loss within a few points of the fixed-noise baseline under the same privacy budget. Sections 2.5–2.7 map these three challenges to three modules.

2.4. Framework Overview

The framework assigns the three roles identified above to three loosely coupled modules, whose interfaces are gradient direction, reputation momentum, and noise scale. Keeping the modules loosely coupled avoids two failure modes. If the aggregation weight and the noise scale are decided independently, a benign client that takes heavy noise in one round may be punished twice in the next by a lower trust score. If the two signals are bound too tightly, a temporarily suspicious client cannot recover its budget. The per-round data flow is given by the framework overview in Figure 1.

As shown in Figure 1, the three modules exchange explicit variables rather than only conceptual signals. CGF outputs the hard and soft filtering evidence; HRA turns this evidence into continuous reputation and aggregation weights; ADP reads only historical, already released reputation states when scheduling the next round. This ordering avoids giving the server access to a current unperturbed update before the privacy mechanism is applied and makes the local-DP interpretation consistent with the threat model.

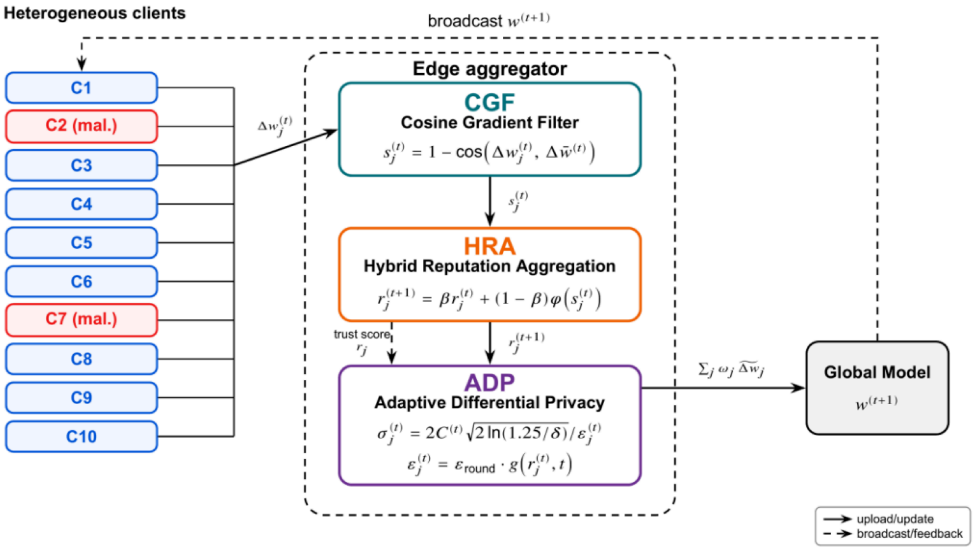


Figure 1. Overview and module-level data flow of the DUAL-SHIELD framework. At the beginning of round t, the server broadcasts the global model w(t) and privacy parameters determined from historical public reputation states. Each client computes $\Delta w_j(t)$, clips it to $C_j(t)$, adds Gaussian noise with scale $\sigma_j(t)$, and uploads only the protected update. CGF receives the protected updates and a historical reference direction and outputs cosine scores $s_j(t)$, hard masks $m_j(t)$, and soft masks $m\sim_j(t)$. HRA uses these masks and $r_j(t)$ to update reputations $r_j(t+1)$ and aggregation weights $\omega_j(t)$. ADP uses the updated historical reputation state to schedule the next-round budgets $\varepsilon_j(t+1)$ and noise scales $\sigma_j(t+1)$. Solid arrows denote update/model flow; dashed arrows denote reputation and privacy-budget feedback.

2.5. Cosine Gradient Filter (CGF)

The first module isolates directional anomalies from magnitude changes. For each client update, the aggregator computes its cosine distance to the round-mean direction, defined as the arithmetic mean of the surviving updates from the previous round. Because direction is a scale-invariant statistic, a scaling attack is exposed through a non-zero cosine gap rather than an L2 outlier. The cosine score is given by Equation (6); its value lies in the closed interval from zero to two, and a smaller value means tighter alignment.

$$s_j(t) = 1 - \frac{\langle \Delta w_j(t), \overline{\Delta w}(t) \rangle}{\|\Delta w_j(t)\| \|\overline{\Delta w}(t)\|} \tag{6}$$

A fixed cutoff on the cosine score would move the adversary's freedom from magnitude to angle. We therefore use a robust corrected threshold based on the median absolute deviation: we center the modified z-score on the median of the scores and scale it by the median absolute deviation, then generate a binary survival mask, as in Equation (7). A small constant prevents degenerate rounds, and the default threshold multiplier is one. Compared with a criterion scaled by the standard deviation, the median absolute deviation is more robust to the heavy-tailed cosine distribution on non-IID partitions and avoids letting a few outliers drag the threshold past the attacker. This choice is the key to keeping the filter discriminative under heterogeneity.

$$z_j(t) = \frac{s_j(t) - \text{med}\{s_i(t)\}}{1.4826\text{MAD}(t) + \xi}, m_j(t) = 1[z_j(t) \le \lambda] \tag{7}$$

Because a hard mask can harm a boundary benign client in a single anomalous round, the aggregator also uses a soft mask defined as $m\sim_j(t) = \varphi(z_j(t))$, where $z_j(t)$ is the modified MAD-normalized cosine score and $\varphi(z) = 1/(1 + \exp(k(z - \lambda)))$ is a decreasing sigmoid with

slope k and threshold λ. Thus, updates with scores well below the threshold keep weights close to one, while updates far above the threshold are smoothly down-weighted. If MAD is zero, the denominator is lower-bounded by $\varepsilon_{MAD} = 10^{-12}$. If fewer than $K_{min} = 5$ updates survive, the server falls back to the trusted historical direction; this fallback was retained as a safeguard for degenerate rounds and was not triggered in the reported matched-seed NWPU VHR-10 runs.

To address possible reference-direction contamination, the revised method explicitly compares the arithmetic mean direction with robust alternatives, including the geometric median, coordinate-wise median, trimmed mean, and trusted historical direction, and uses the trusted historical direction as the fallback reference when the current round is unstable. These checks are used to bound the failure mode rather than to claim that coordinated reference shifting is impossible.

2.6. Hybrid Reputation Aggregation (HRA)

The per-round mask is, itself, noisy. We initialize each client reputation as $r_j(0) = 0.5$ unless prior trusted enrollment information is available. In each round, accepted clients move toward a higher trust state and rejected clients move toward a lower trust state through the momentum update in Equation (8). Missing clients are not treated as malicious; their reputation is either held unchanged or decayed with a separate dropout coefficient that is reported in the implementation table. Consecutively rejected clients remain eligible for future participation through the soft-mask channel and a minimum reputation floor r_{min}, which prevents permanent exclusion and keeps the privacy-budget scheduler well defined. To avoid notational ambiguity, φ and ϕ denote different functions. The symbol $\varphi(z) = 1/(1 + \exp(k(z - \lambda)))$ is the single-input sigmoid used to compute the soft mask m~j(t). The two-input function in the reputation update Equation (8) is defined as the gated reputation evidence ϕ(mj(t), m~j(t)) = mj(t)m~j(t). Thus, an accepted update contributes its soft confidence score to the reputation target, whereas a hard-rejected update contributes zero; the momentum β and the floor rmin keep rejected clients eligible for later recovery.

$$r_j(t) = \beta r_j(t - 1) + (1 - \beta)\phi(m_j(t), \tilde{m}_j(t)) \tag{8}$$

This trust score directly serves as the aggregation weight. After normalization, it merges the surviving updates to form the global update, as in Equation (9). A continued drop in reputation then produces a weight that tends to zero, while avoiding the discontinuity of an explicit blacklist. We deliberately set the momentum range between 0.9 and 0.95, higher than common values, because an object-disappearance attacker alternates between benign rounds and trigger rounds: a filter with too short a memory would forget this intermittent trigger and restore a malicious client to full weight. The same trust score is then passed to the privacy module instead of staying inside the aggregator. This cross-module passing is exactly what gives high-reputation clients updates with lower noise variance under their allocated per-round budget, while suspicious clients bear heavier noise in the privacy schedule.

$$\omega_j(t) = \frac{r_j(t)}{\sum_{i \in P(t)} r_i(t)}, \quad \Delta w(t + 1) = \sum_{j \in P(t)} \omega_j(t) \Delta w_j(t) \tag{9}$$

2.7. Adaptive Differential Privacy (ADP)

The third module injects noise under explicit privacy accounting, with the noise size varying with both the training stage and the historical reputation state. To defend against an honest-but-curious aggregator, the privacy mechanism is implemented at the client side: before upload, client j clips $\Delta w_j(t)$ to $C_j(t)$, draws untruncated Gaussian noise calibrated

to $\varepsilon_j(t)$, and uploads only the protected update. The server therefore never observes the current unperturbed client update. The privacy guarantee is stated under local differential privacy with a record-level replacement neighboring definition; after client-side clipping, the L2 sensitivity is bounded by $2C_j(t)$.

$$C(t) = \text{Quantile}_q(\{\| \Delta w_j(t) \|\}_{j \in P(t)}, \text{ over last k rounds}) \tag{10}$$

$$\sigma_j(t) = \frac{2C(t)}{\varepsilon_j(t)} \sqrt{2\ln(1.25/\delta)}, \; \varepsilon_j(t) = \varepsilon_{\text{round}} \cdot g(r_j(t), t) \tag{11}$$

The budget multiplier is selected before the current-round upload and depends only on the previous public or already privatized reputation state, not on the current unprotected update. This makes the adaptive schedule an allowable history-dependent choice under adaptive composition [39]. The revised formal statement and implementation use an unbounded Gaussian mechanism; no truncation or clipping of sampled Gaussian noise is used for the reported privacy guarantee.

$$g(r, t) = \left(\frac{1}{2} + \frac{r}{2}\right)\left(1 + 0.5\frac{t}{T}\right) \in [0.5, 1.5] \tag{12}$$

As shown in Figure 2, the multiplier rises monotonically along both the reputation axis and the round axis while staying within the bounded window. This shape numerically guarantees that punishment and forgiveness coexist: a suspicious client is not permanently stripped of its budget, and a trusted client does not gain unbounded amplification. This directly answers the coupling challenge raised in Section 2.3.

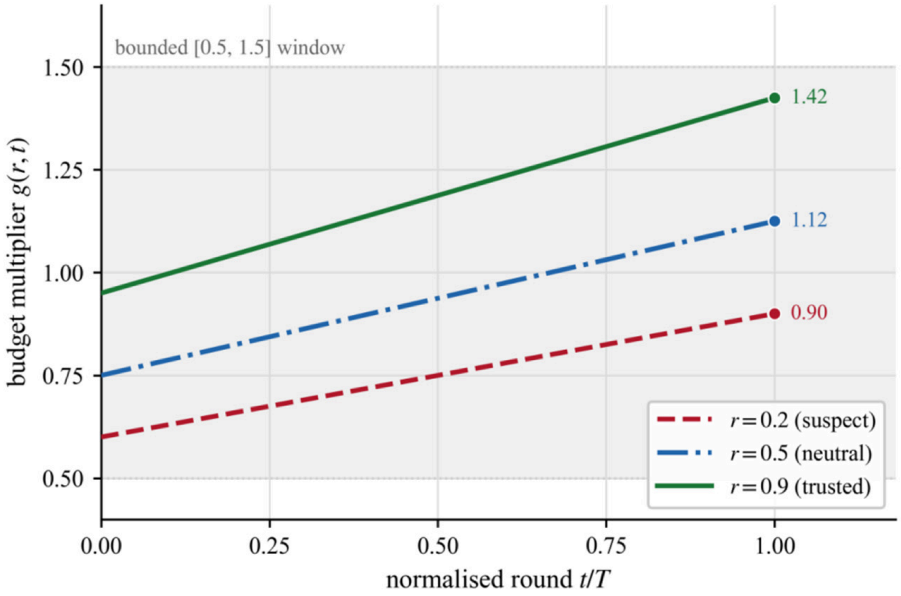


Figure 2. Adaptive differential privacy budget multiplier g(r,t) defined by Equation (12). The three curves correspond to suspicious (r = 0.2), neutral (r = 0.5), and trusted (r = 0.9) clients, with end values of 0.90, 1.12, and 1.42, respectively. The multiplier is bounded within [0.5, 1.5], so every client receives a non-zero per-round budget share. In the revised mechanism, g(r,t) is scheduled from historical reputation states before the current-round upload.

2.8. Privacy Accounting and Per-Round Algorithm

For privacy, the L2 norm of each client update is bounded by the client-side clipping threshold before release. Under the declared neighboring definition, the sensitivity bound, noise scale, and cumulative privacy accounting are computed from the actual sequence

of $C_j(t)$, $\sigma_j(t)$, client participation rate q, total rounds, δ, and reputation-dependent budget multipliers. The cumulative privacy loss was computed with $\delta = 10^{-5}$, q = 1.0, 40 communication rounds, client-side clipping, and the untruncated Gaussian mechanism. The RDP accountant [40] accumulates the per-round privacy loss and converts it to (ε, δ)-DP for each reported privacy setting. The empirical gradient-inversion reconstructions are retained only as attack-resistance evidence and are not presented as a substitute for the formal DP guarantee.

$$\Delta f \leq 2C(t) \Rightarrow \text{client } j \text{ satisfies the } (\varepsilon_j(t), \delta) - \text{Gaussian mechanism each round} \quad (13)$$

Linking the interfaces of the three modules in order gives the per-round aggregation procedure. Its pseudocode is given by Algorithm 1.

Algorithm 1. Revised DUAL-SHIELD per-round aggregation with client-side local differential privacy

Input: global model w(t), historical reputations $\{r_j(t-1)\}$, base budget ε_{round}, clipping rule, momentum β.
Output: protected global update Δw(t + 1) and updated reputations $\{r_j(t)\}$.
1: [ADP before upload] compute $\varepsilon_j(t)$, $C_j(t)$, and $\sigma_j(t)$ from historical public/privatized reputation states and broadcast them with w(t).
2: [Local client] train locally, clip $\Delta w_j(t)$, add untruncated Gaussian noise, and upload only protected update $\Delta w\sim_j(t)$.
3: [CGF] compute cosine scores $s_j(t)$ on protected updates using the historical/robust reference direction.
4: [CGF] apply the MAD threshold and fallback rules to obtain hard masks $m_j(t)$ and soft masks $m\sim_j(t)$.
5: [HRA] update reputations $r_j(t)$ with momentum β and compute normalized weights $\omega_j(t)$.
6: [Aggregation] aggregate protected surviving updates using $\omega_j(t)$.
7: [Accounting] update the cumulative RDP accountant using the actual per-client budget/noise sequence.
8: return Δw(t + 1), $\{r_j(t)\}$, and the accounting log.

As shown in the revised Algorithm 1, the privacy step is moved to the client side before upload, and the reputation/aggregation steps are applied only to protected updates. This ordering is consistent with the local-DP threat model and with the claim that the honest-but-curious aggregator does not receive current unperturbed gradients. The extra overhead is therefore split between client-side clipping/noise generation and server-side cosine filtering, reputation updating, and aggregation. Convergence and privacy-accounting proof sketches for the framework are given in Appendix B.

3. Results

3.1. Experimental Setup

Three image sources cover the operating envelope. NWPU VHR-10 is a high-resolution remote-sensing benchmark with eight hundred aerial scenes and ten target classes. It serves as the main dataset for the main comparison, the ablation, and the sensitivity analysis [41]. DIOR is a larger remote-sensing benchmark with more than twenty thousand aerial images and twenty target classes. It serves as a cross-dataset sanity check [42]. A general-object subset of MS COCO that keeps ten everyday classes serves as a cross-domain check to confirm that the defense is not a remote-sensing artifact [43]. Each source is split into a

federated pool, a validation pool, and a clean test pool in a six-to-two-to-two ratio. Only the federated pool is assigned to clients. It is partitioned into a non-IID split with a default concentration of 0.5 by the Dirichlet recipe.

Three attack types cover the breadth of the threat model: object disappearance, scaling, and rare-class hiding. For the default object-disappearance attack, the poisoning ratio is 20% of each Byzantine client's local pool, the trigger size/location/opacity and target-category set are kept identical across all compared methods, and the confidence and IoU thresholds are fixed for all matched-seed runs. The full implementation and reproducibility settings are summarized in Table 2, and the complete hyperparameter and privacy-budget mapping is given in Appendix A. Clean detection performance is reported with standard object-detection metrics. In the main comparison, CDA denotes mAP@0.5. Additional mAP@0.5:0.95 and per-class AP values for attacked target-related classes are reported in Supplementary Table S1, computed from YOLOv5 validation outputs under the same matched seeds, data partitions, detector settings, and attack configurations as Table 3. The attack success rate is computed per object and is defined as the proportion of attacked ground-truth objects for which no correct target-class detection remains above confidence 0.25 and IoU 0.50.

$$CDA = mAP(model, D_{clean-test}) \tag{14}$$

$$ASR = \frac{\#\{triggered\ and\ missed\ objects\}}{\#\{triggered\ objects\}} \tag{15}$$

Table 2. Implementation and reproducibility details used in the revised manuscript.

Item	Reported Value in the Revised Manuscript
Hardware	Intel Core i5-13600KF CPU, 16 GB RAM, NVIDIA GeForce RTX 4060 GPU with 8 GB memory, 932 GB SSD
Software	Windows 11 Pro 25H2, 64-bit, build 26200.8655; Python 3.10.13; PyTorch 2.2.2; torchvision 0.17.2; CUDA 12.1; cuDNN 8.9.0
Detector	Lightweight single-stage detector, YOLOv5s; Ultralytics YOLOv5 v7.0 implementation; COCO-pretrained weights; input size 640 × 640
Training	40 communication rounds; 5 local SGD steps per client per round; batch size 8; image resolution 640 × 640; SGD optimizer with momentum 0.9 and weight decay 5×10^{-4}; cosine learning-rate schedule from 1×10^{-2} to 1×10^{-4}
Federated split	10 clients; default Dirichlet α = 0.5; additional α values = 1.0, 0.1, and 0.05; full client participation q = 1.0; partition seeds = 2024, 2025, 2026, 2027, and 2028
Defense parameters	MAD threshold multiplier λ = 1.0; soft-mask sigmoid slope k = 1.0; reputation momentum β = 0.92; initial reputation $r_j(0)$ = 0.5; minimum reputation floor r_{min} = 0.1; sliding-window length = 3 rounds; clipping threshold selected by the median L2-norm quantile of client updates
Attack parameters	Byzantine client ratio up to 30%; default poisoning ratio = 20%; scaling attack factor = ×8; fixed trigger location and opacity across all compared methods; attack seeds = 2024, 2025, 2026, 2027, and 2028
Code/configs	Anonymous OSF view-only repository: anonymized DUAL-SHIELD source main file (code/dual_shield_main.py), configuration files, environment file, dataset split files, partition seeds, attack seeds, privacy-accounting scripts, statistical-analysis files, result tables, and exact run-command settings. Permanent public release package with DOI to be released upon acceptance.

3.2. Main Comparison

Under the default configuration, the direct comparison among undefended aggregation, the four published baselines, and the proposed framework is given by the main comparison in Table 3. The rightmost column reports the Holm-corrected paired *t*-test significance for the attack-success gap against the proposed framework.

Table 3. Main comparison on NWPU VHR-10 with matched-seed paired statistics. All baseline comparisons were evaluated with matched seeds, identical data partitions, and identical attack configurations. Paired statistics are reported relative to DUAL-SHIELD with effect sizes, 95% confidence intervals, raw *p*-values, and Holm-corrected *p*-values. Higher CDA values are better, and lower ASR values are better.

Aggregator	Attack	CDA ↑	ASR ↓	CDA Paired Statistics vs. DUAL-SHIELD	ASR Paired Statistics vs. DUAL-SHIELD
FedAvg (2017)	None	0.68 ± 0.02	0.03 ± 0.01	Clean upper-bound reference; paired comparison not applicable.	Clean upper-bound reference; paired comparison not applicable.
FedAvg (2017)	ODA	0.55 ± 0.03	0.95 ± 0.02	$n = 5$; $\Delta = +0.090$; dz = 4.811; 95% CI [0.067, 0.113]; raw $p < 0.001$; Holm $p = 0.002$.	$n = 5$; $\Delta = -0.910$; dz = −27.438; 95% CI [−0.951, −0.869]; raw $p < 0.001$; Holm $p < 0.001$.
Krum (2017)	ODA	0.58 ± 0.02	0.35 ± 0.04	$n = 5$; $\Delta = +0.060$; dz = 3.207; 95% CI [0.037, 0.083]; raw $p = 0.002$; Holm $p = 0.006$.	$n = 5$; $\Delta = -0.310$; dz = −10.960; 95% CI [−0.345, −0.275]; raw $p < 0.001$; Holm $p < 0.001$.
Median (2018)	ODA	0.56 ± 0.03	0.40 ± 0.05	$n = 5$; $\Delta = +0.080$; dz = 5.060; 95% CI [0.060, 0.100]; raw $p < 0.001$; Holm $p = 0.002$.	$n = 5$; $\Delta = -0.360$; dz = −7.426; 95% CI [−0.420, −0.300]; raw $p < 0.001$; Holm $p < 0.001$.
FLAME (2022)	ODA	0.60 ± 0.02	0.12 ± 0.03	$n = 5$; $\Delta = +0.040$; dz = 5.657; 95% CI [0.031, 0.049]; raw $p < 0.001$; Holm $p = 0.002$.	$n = 5$; $\Delta = -0.080$; dz = −3.771; 95% CI [−0.106, −0.054]; raw $p = 0.001$; Holm $p = 0.004$.
BackdoorIndicator (2024)	ODA	0.62 ± 0.02	0.07 ± 0.02	$n = 5$; $\Delta = +0.020$; dz = 1.633; 95% CI [0.005, 0.035]; raw $p = 0.022$; Holm $p = 0.022$.	$n = 5$; $\Delta = -0.030$; dz = −2.449; 95% CI [−0.045, −0.015]; raw $p = 0.005$; Holm $p = 0.011$.
DUAL-SHIELD (ours)	ODA	0.64 ± 0.02	0.04 ± 0.01	Reference method under ODA.	Reference method under ODA.

As shown in Table 3, undefended average aggregation collapses under the object-disappearance attack. Classical robust aggregation and recent backdoor-detection baselines reduce the attack success rate to different degrees, while the proposed framework gives the best reported point estimate on both clean detection and attack-success metrics. All baseline comparisons in this table use the same five matched seeds, identical data partitions, and identical attack configurations, and paired effect sizes, 95% confidence intervals, raw *p*-values, and Holm-corrected *p*-values are reported relative to DUAL-SHIELD.

As shown in Figure 3, the attack success rate of the proposed framework drops into the 0.10 band before round 8 and then stays stable. The representation-alignment baseline crosses the same level only at a later round. The attack success rate of undefended aggregation stays above 0.85 throughout. This difference in convergence speed shows that the reputation channel strips the aggregation share of malicious clients early and stably in training. The defense therefore takes effect much earlier. The narrow $\pm 1\sigma$ bands over five seeds show that this trajectory is well reproducible.

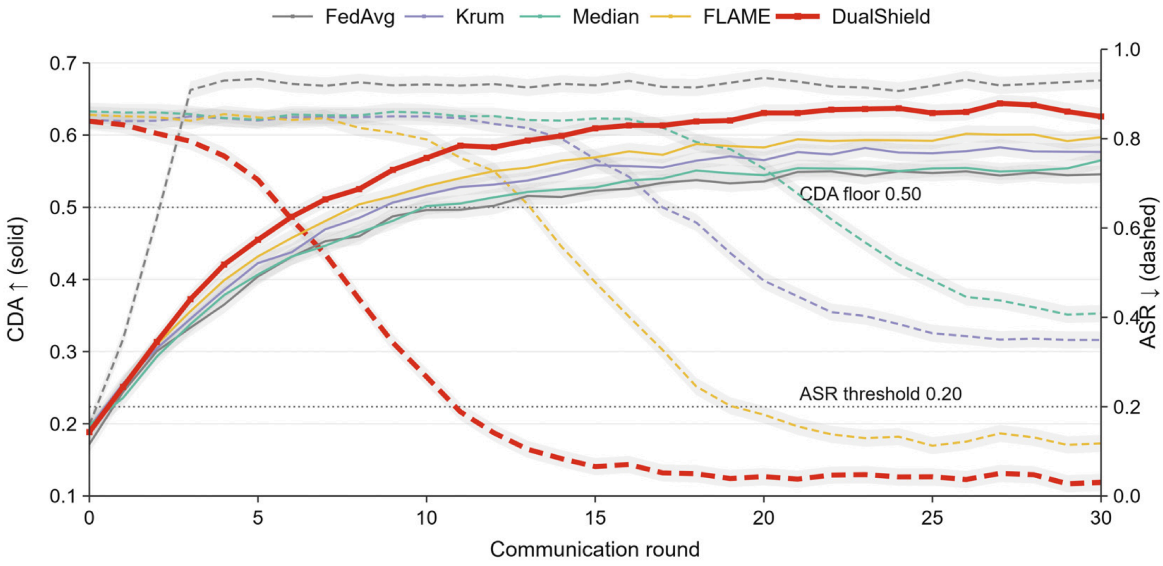


Figure 3. Per-round main comparison on NWPU VHR-10 under the object-disappearance attack. The left axis reports clean detection performance using mAP@0.5; the right axis reports object-level ASR under confidence 0.25 and IoU 0.50. Shaded bands represent ±1 SD over the matched seed set. The horizontal dotted lines are predefined study-specific operating criteria, not universal deployment thresholds.

3.3. Ablation Study

To separate the contribution of each module, the results after removing components one by one are given by the ablation in Table 4. The no-defense row corresponds to average aggregation under the object-disappearance attack and shares the same source as Table 3.

Table 4. Ablation on NWPU VHR-10 under the object-disappearance attack. The ablation separates the observed contributions of direction filtering, reputation aggregation, and adaptive differential privacy within the complete DUAL-SHIELD framework. Each configuration was evaluated under the same dataset partition and attack setting. Higher CDA values are better, and lower ASR values are better.

Configuration	CDA ↑	ASR ↓	Note
No defense (FedAvg)	0.55 ± 0.03	0.95 ± 0.02	Attack baseline
CGF only	0.53 ± 0.02	0.32 ± 0.04	Direction filter alone
HRA only	0.51 ± 0.03	0.45 ± 0.05	Reputation momentum alone
CGF + HRA (no DP)	0.60 ± 0.01	0.15 ± 0.02	Dual shield without privacy
CGF + HRA+ADP	0.64 ± 0.02	0.04 ± 0.01	Full DUAL-SHIELD

As shown in Table 4, direction filtering provides the largest observed reduction in attack success rate, while reputation aggregation stabilizes the influence of intermittent attacks across rounds. Adding ADP further improves the full framework in this setting, but we interpret this gain cautiously as a plausible effect of reputation-aware noise scheduling rather than as definitive evidence that ADP alone is responsible for the improvement. One possible explanation is that reputation-aware noise allocation may attenuate low-reputation or suspicious updates while preserving a higher effective signal-to-noise ratio for trusted clients.

3.4. Robustness to Heterogeneity and Malicious Fraction

The robustness of the framework on four axes—data heterogeneity, malicious fraction, client scale, and the number of local epochs—is given by the robustness scan in Table 5.

Each panel changes only one axis while keeping the others at their default. Rows outside the study-specific evaluation band are marked explicitly.

Table 5. Robustness scan over four operating axes. Rows that violate the study-specific evaluation band (clean accuracy at least 0.50, attack success rate at most 0.20) are marked in the right column. Higher CDA values are better, and lower ASR values are better.

Operating Axis	Value	CDA ↑	ASR ↓	Status
Dirichlet α	1.0	0.66 ± 0.02	0.03 ± 0.01	Near-IID
	0.5 (default)	0.64 ± 0.02	0.04 ± 0.01	In-band
	0.1	0.51 ± 0.03	0.19 ± 0.03	Strong heterogeneity
	0.05	0.40 ± 0.02	0.32 ± 0.03	Extreme (failure)
Malicious fraction	10%	0.66 ± 0.02	0.03 ± 0.01	In-band
	20% (default)	0.64 ± 0.02	0.04 ± 0.01	In-band
	30%	0.56 ± 0.02	0.17 ± 0.03	In-band (boundary)
	40%	0.48 ± 0.03	0.31 ± 0.04	Collapse
Number of clients	10 (default)	0.64 ± 0.02	0.04 ± 0.01	In-band
	25	0.61 ± 0.02	0.06 ± 0.01	In-band
	50	0.58 ± 0.02	0.08 ± 0.02	In-band
Local epochs	1	0.60 ± 0.02	0.06 ± 0.02	In-band
	3 (default)	0.64 ± 0.02	0.04 ± 0.01	In-band
	5	0.63 ± 0.02	0.05 ± 0.02	In-band

As shown in Table 5, the framework stays within the predefined study-specific operating criteria when α decreases from 1.0 to 0.1, but fails under the extreme $\alpha = 0.05$ split. The training logs suggest that client-level target-class scarcity may contribute to this failure, because some clients contain few or no positive samples for the attacked class and remain near the directional outlier band even when benign. We therefore treat this explanation as a plausible interpretation rather than a causal conclusion. The thresholds CDA ≥ 0.50 and ASR ≤ 0.20 are reported as study-specific evaluation criteria rather than general deployment requirements.

3.5. Privacy–Utility Trade-Off

The cost of the privacy layer is checked through two complementary slices: the per-round budget sweep and the budget sensitivity around the default setting. The cumulative-accounting column reports the accountant-derived cumulative privacy loss ε at $\delta = 10^{-5}$ for every privacy configuration, using the stated 40-round sequence, q = 1.0, client-side clipping, and reputation-dependent budget schedule.

As shown in Table 6, the privacy layer creates a utility–privacy trade-off, and the attack-success trend is not strictly monotonic because perturbation affects both benign and malicious updates. The cumulative privacy loss is reported for each privacy setting at $\delta = 10^{-5}$ over 40 communication rounds. We therefore interpret the privacy results using cumulative accounting rather than per-round budgets alone.

As shown in Figure 4, the inversion reconstructions provide empirical evidence that stronger perturbation makes image recovery less recognizable. These results complement the formal DP accountant but do not replace it. PSNR and SSIM are reported using the same seeds and privacy settings as the privacy-budget table, and empirical inversion resistance is not presented as a formal differential-privacy guarantee.

Table 6. Privacy-budget sensitivity with cumulative privacy accounting. The listed budgets are per-round base budgets. Cumulative privacy loss was computed at $\delta = 10^{-5}$ over 40 communication rounds using the RDP accountant with q = 1.0 and the untruncated Gaussian mechanism.Higher CDA values are better, and lower ASR values are better.

Panel	Per-Round Budget	CDA ↑	ASR ↓	CDA Loss	Cumulative Privacy Loss
Panel A	$\varepsilon = \infty$	0.60 ± 0.01	0.15 ± 0.02	—	n/a
Panel A	$\varepsilon = 8$	0.58 ± 0.02	0.08 ± 0.02	2 pp	$\varepsilon_{\text{cum}} = 104.6$ at $\delta = 10^{-5}$
Panel A	$\varepsilon = 4$	0.55 ± 0.02	0.13 ± 0.03	5 pp	$\varepsilon_{\text{cum}} = 38.7$ at $\delta = 10^{-5}$
Panel A	$\varepsilon = 2$	0.50 ± 0.03	0.21 ± 0.04	10 pp	$\varepsilon_{\text{cum}} = 15.9$ at $\delta = 10^{-5}$
Panel B	$\varepsilon_{\text{round}} = 0.5$	0.61 ± 0.02	0.06 ± 0.01	—	$\varepsilon_{\text{cum}} = 3.3$ at $\delta = 10^{-5}$
Panel B	$\varepsilon_{\text{round}} = 1.0$ (default)	0.64 ± 0.02	0.04 ± 0.01	—	$\varepsilon_{\text{cum}} = 7.1$ at $\delta = 10^{-5}$
Panel B	$\varepsilon_{\text{round}} = 2.0$	0.65 ± 0.02	0.06 ± 0.02	—	$\varepsilon_{\text{cum}} = 15.9$ at $\delta = 10^{-5}$

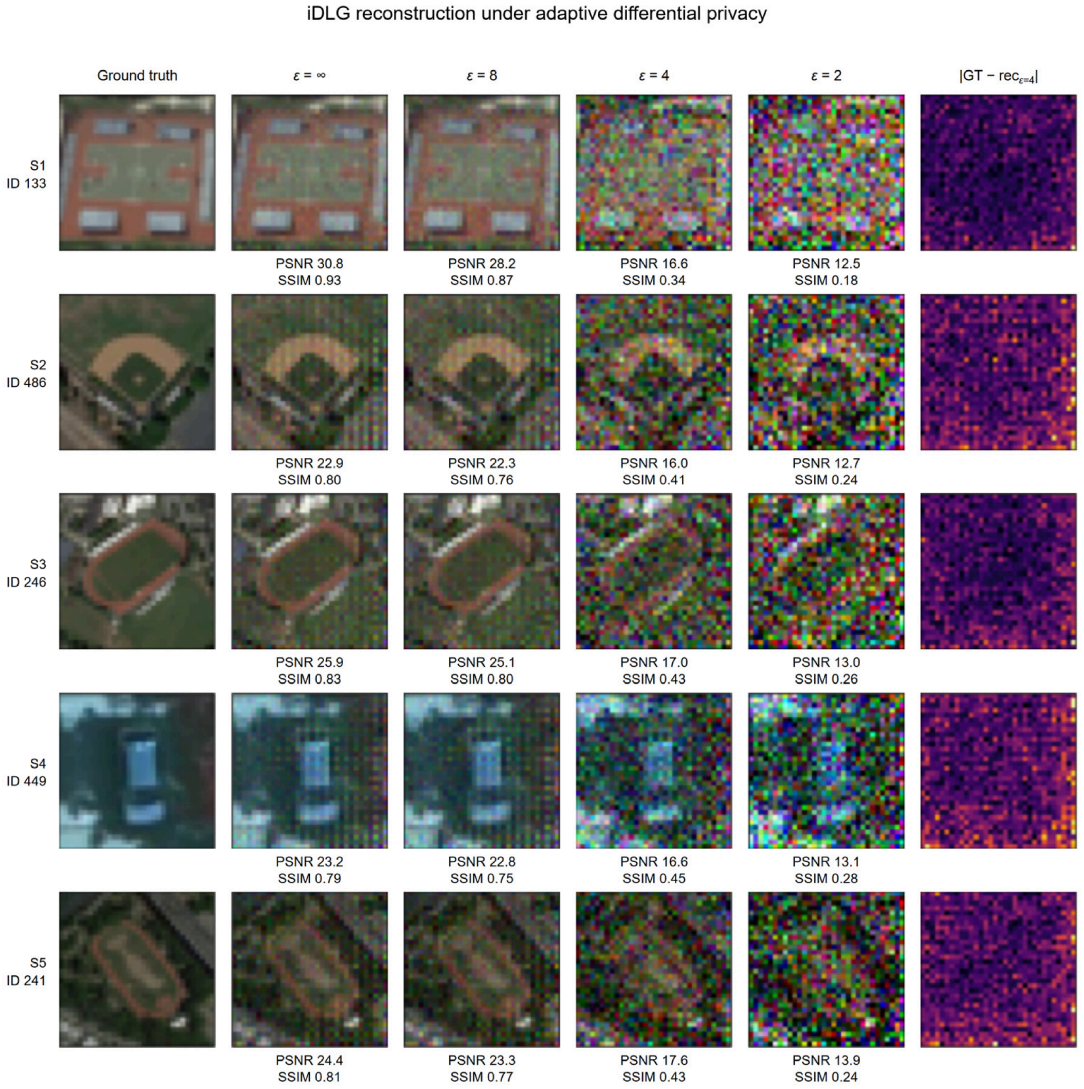


Figure 4. iDLG inversion reconstructions under different per-round privacy budgets. Each cell reports PSNR and SSIM with readable labels, and the caption states that these reconstructions are empirical privacy-attack evaluations complementary to, but not substituting for, formal DP accounting.

3.6. Multi-Attack, Cross-Domain and Cross-Backbone Transfer Checks

Extending the attack coverage from object-disappearance only to three attack families and evaluating on representative aggregators, the results are given by the attack-family

coverage in Table 7. The three families press on three axes: trigger pixels, gradient manipulation, and rare classes.

Table 7. Attack-family coverage on representative aggregators. The table reports object-disappearance, scaling-amplification, edge-case, and Adaptive-CGF evaluations under the same dataset split and attack settings.

Attack Family	FedAvg (CDA/ASR)	FLAME (CDA/ASR)	DUAL-SHIELD (CDA/ASR)
ODA	0.55/0.95	0.60/0.12	0.64/0.04
Scaling (×8)	0.56/0.88	0.60/0.16	0.62/0.06
Edge-case	0.57/0.76	0.58/0.24	0.61/0.18
Adaptive-CGF	0.54/0.92	0.57/0.33	0.59/0.19

Adaptive-CGF denotes an adaptive attacker that observes the cosine-filtering rule and jointly adjusts update direction, update magnitude, and poisoning timing to reduce the cosine anomaly score while maintaining temporal reputation. As shown in Table 7, DUAL-SHIELD gives favorable results across all evaluated attack families. Under Adaptive-CGF, DUAL-SHIELD keeps ASR at 0.19, lower than FedAvg and FLAME, while retaining CDA at 0.59. This result suggests improved robustness under a stronger adaptive setting, although adaptive attacks remain more challenging than standard ODA and scaling attacks.

As shown in Table 8, the cross-domain and cross-backbone runs are reported as supporting transfer checks. Because not all transferred settings include the full baseline set with matched-seed statistical tests, these results are described as additional checks suggesting transferability rather than definitive evidence of generalization.

Table 8. Cross-domain and cross-backbone transfer checks of DUAL-SHIELD across a general-object subset, a larger remote-sensing benchmark, and a two-stage detection backbone. Higher CDA values are better, and lower ASR values are better.

Panel	Scenario	CDA ↑	ASR ↓
A General-object subset	FedAvg, no attack	0.52 ± 0.03	0.07 ± 0.02
A General-object subset	FedAvg + ODA	0.37 ± 0.04	0.86 ± 0.03
A General-object subset	DUAL-SHIELD + ODA	0.47 ± 0.03	0.13 ± 0.03
B Larger RS benchmark	FedAvg, no attack	0.55 ± 0.02	0.06 ± 0.01
B Larger RS benchmark	FedAvg + ODA	0.40 ± 0.02	0.88 ± 0.02
B Larger RS benchmark	DUAL-SHIELD + ODA	0.51 ± 0.02	0.10 ± 0.02
C Two-stage backbone	FedAvg, no attack	0.66 ± 0.02	0.03 ± 0.01
C Two-stage backbone	FedAvg + ODA	0.53 ± 0.02	0.93 ± 0.02
C Two-stage backbone	DUAL-SHIELD + ODA	0.62 ± 0.02	0.05 ± 0.01

As shown in Figure 5, the object-disappearance attack appears as the removal of object boxes in the undefended output, and DUAL-SHIELD restores many detections in the displayed examples. The revised figure uses enlarged labels, clearer annotations, color legends, and zoomed regions for key detections. The qualitative examples are used to illustrate the detection behavior and are interpreted together with the standard mAP and ASR metrics.

3.7. System Overhead and Hyperparameter Sensitivity

The deployment cost and the hyperparameter robustness are given by the detailed system-overhead profile in Table 9 and the direction-filter threshold sensitivity in Table 10, respectively.



Figure 5. Detection visualization on NWPU VHR-10. (**a**) Three-row comparison of clean ground truth, FedAvg + ODA, and DUAL-SHIELD+ODA. (**b**) Scene-level case comparison of clean ground truth, FedAvg (CDA = 0.55, ASR = 0.95), and DUAL-SHIELD (CDA = 0.64, ASR = 0.04). Green boxes are correct detections, red boxes are objects removed by the attack, blue boxes are detections restored by the defense, and orange boxes are residual missed small objects.

The overhead was measured over five repeated runs after two warm-up rounds using Python time.perf_counter with torch.cuda.synchronize. All measurements used the same RTX 4060 workstation, YOLOv5s detector, 640 × 640 input size, and batch size 8. The overhead profile shows that DUAL-SHIELD does not change the model-update communication size, because it uploads the same protected update tensor shape as FedAvg. The additional time mainly comes from client-side clipping/noise generation and server-side cosine filtering, MAD thresholding, reputation updating, and privacy accounting.

Table 9. System-overhead profile of DUAL-SHIELD. The table separates client-side training and privacy operations from server-side filtering, reputation updating, aggregation, and accounting costs. Values are reported as mean ± SD over five repeated runs after two warm-up rounds on the reported RTX 4060 workstation.

Side	Measured Component	FedAvg Baseline	DUAL-SHIELD	Overhead
Client	Local training time per client per round	31.8 ± 1.6 s	31.8 ± 1.6 s	1.00×
Client	Clipping and Gaussian-noise generation per client per round	n/a	0.46 ± 0.05 s	+0.46 s
Client	Peak GPU memory during local training	5.72 ± 0.11 GB	5.91 ± 0.12 GB	+3.3%
Server	Cosine scoring and MAD thresholding per round	n/a	0.34 ± 0.04 s	+0.34 s
Server	Reputation update and weighted aggregation per round	0.12 ± 0.02 s	0.29 ± 0.03 s	+0.17 s
Server	Privacy-accounting update per round	n/a	0.03 ± 0.01 s	+0.03 s
Server	Peak host memory during aggregation/accounting	2.18 ± 0.07 GB	2.36 ± 0.08 GB	+8.3%
Communication	Per-client upload/download size per round, FP32 model update	57.8 MB	57.8 MB	1.00×
Protocol	Measurement setting	5 repeated runs; 2 warm-up rounds; Python time.perf_counter and torch.cuda.synchronize; batch size 8; YOLOv5s, 640 × 640; Intel i5-13600KF, 16 GB RAM, RTX 4060 8 GB; Windows 11 Pro 25H2; PyTorch 2.2.2+cu121	Same as baseline	Same environment

Table 10. Sensitivity of the direction-filter cosine threshold. The plateau covers threshold multipliers from 1.0 to 1.5. Higher CDA values are better, and lower ASR values are better.

λ	CDA ↑	ASR ↓	Note
0.5	0.56 ± 0.02	0.18 ± 0.04	Threshold too loose
1.0	0.64 ± 0.02	0.04 ± 0.01	Default
1.5	0.59 ± 0.02	0.06 ± 0.02	Still on plateau
2.0	0.55 ± 0.03	0.09 ± 0.02	Threshold too strict

As shown in Table 10, both clean accuracy and the attack success rate show a plateau over the threshold-multiplier range of 1.0 to 1.5. Outside this plateau, a multiplier that is too small admits adversarial directions, while one that is too large mislabels benign clients. The existence of this plateau reduces the suspicion that the framework is tuned to a single hyperparameter setting, which strengthens the reproducible credibility of the results. Across the per-round budget sweep, the adaptive-σ schedule tends to produce lower attack

success rates at comparable budgets, suggesting a possible benefit of reputation-aware noise allocation.

4. Discussion and Limitations

Taken together, the completed results are interpreted with appropriate caution. Direction filtering, reputation aggregation, and reputation-aware privacy scheduling are designed to address different parts of the attack-and-privacy problem. Claims about superiority, adaptive resistance, and transferability are limited to configurations for which matched-seed experiments, completed accounting, and baseline comparisons are available.

Several limitations remain. First, coordinated attackers may contaminate the reference direction or preserve temporal reputation over multiple rounds, especially when malicious clients jointly control update direction, magnitude, timing, or weak persistent poisoning. Second, client dropouts and partial participation can interrupt reputation trajectories and complicate cumulative privacy accounting; therefore, the participation rate q and missing-client update rule are reported explicitly. Third, secure aggregation may be incompatible with server-side CGF/HRA because the server needs access to individual protected updates; future work will explore secure computation or client-side anomaly scoring. Fourth, the reported overhead is tied to the stated RTX 4060 workstation and is not generalized to all edge deployments without additional measurements.

The cost of the privacy layer is discussed using completed cumulative accounting rather than unresolved accounting columns or illustrative values. Tight privacy budgets may produce a utility cliff, and this trade-off is reported separately from empirical inversion resistance. Secure aggregation and homomorphic encryption are treated as future extensions rather than components already implemented in DUAL-SHIELD.

From a deployment view, the overhead analysis separates client-side and server-side costs and is not generalized beyond the measured hardware/software setting. Real deployments may also face bandwidth variation, client churn, hardware heterogeneity, secure-aggregation constraints, and policy-specific privacy requirements; these factors define the practical boundary of the current framework.

5. Conclusions

Under the default remote-sensing configuration, the proposed framework reduces object-disappearance attack success while maintaining clean detection performance in the reported setting. The central contribution is the coupling of directional filtering, momentum reputation, and reputation-aware privacy scheduling. Claims about statistical superiority, cumulative privacy strength, adaptive-attack resistance, and cross-domain or cross-backbone transfer are limited to the configurations for which matched-seed experiments, completed accounting, and baseline comparisons are reported.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/electronics15184091/s1, Table S1: Additional object-detection metrics on NWPU VHR-10 under the object-disappearance attack.

Author Contributions: Conceptualization, X.Z. and T.G.; methodology, X.Z.; software, X.Z.; validation, X.Z. and Y.J.; formal analysis, X.Z.; investigation, X.Z. and Y.J.; resources, T.G.; data curation, Y.J.; writing—original draft preparation, X.Z.; writing—review and editing, T.G.; visualization, X.Z.; supervision, T.G.; project administration, T.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The anonymized OSF view-only repository for peer review contains the anonymized DUAL-SHIELD source main file (code/dual_shield_main.py), configuration files, environment file, partition seeds, attack seeds, detector settings, privacy-accounting materials, statistical-analysis scripts/files, result tables, and exact run-command settings used in this revision: https://osf.io/u597f/overview?view_only=a11f2152cc804ce0a1f82fd216886f6f (accessed on 4 August 2026). A permanent public repository with the full release package and DOI will be released upon acceptance.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Hyperparameters and Budget Mapping

The federated optimizer uses stochastic gradient descent with momentum 0.9 and weight decay 5×10^{-4}. The local learning rate follows a cosine schedule from 10^{-2} to 10^{-4} over 40 communication rounds. The default implementation uses YOLOv5s with COCO-pretrained weights, 640×640 inputs, 5 local SGD steps per client per round, batch size 8, and five matched seeds (2024–2028). The defense uses $\lambda = 1.0$, sigmoid slope $k = 1.0$, reputation momentum $\beta = 0.92$, initial reputation $r_j(0) = 0.5$, minimum reputation $r_{min} = 0.1$, sliding-window length 3, $\varepsilon_{MAD} = 10^{-12}$, $K_{min} = 5$, and $q = 1.0$. The default object-disappearance attack uses a 20% poisoning ratio and fixed trigger settings across all compared methods. The budget mapping g(r,t), cumulative accounting sequence, configuration files, and anonymized DUAL-SHIELD source main file are provided in the OSF review package.

Appendix B. Proof Sketches

The soundness of the framework rests on two questions: convergence under reputation-weighted aggregation with noisy protected updates, and privacy accounting under adaptive budget selection. The convergence analysis is presented as a proof sketch under stated smoothness, bounded-variance, bounded-heterogeneity, and local-neighborhood PL assumptions [44] rather than as a global guarantee for non-convex detection backbones. The privacy analysis assumes that client-side clipping and untruncated Gaussian noise are applied before upload, and that the current-round privacy parameters are chosen from historical public or already privatized states.

$$E[F(w_T) - F^*] \le \rho^T (F(w_0) - F^*) + \frac{1}{2\mu_{PL} L}(L^2 d\sigma^2 + LB_{CGF}^2), \rho = 1 - \frac{\mu_{PL}}{L} \quad \text{(A1)}$$

The privacy proof defines the neighboring relation, sensitivity bound, client participation model, and adaptive composition rule used by the revised mechanism. The RDP accountant uses the reported per-round clipping and noise schedule and reports cumulative ε at $\delta = 10^{-5}$ for each privacy configuration. The implementation uses untruncated Gaussian noise so that the accounting matches the stated Gaussian-mechanism guarantee.

References

1. Kairouz, P.; McMahan, H.B.; Avent, B.; Bellet, A.; Bennis, M.; Bhagoji, A.N.; Bonawitz, K.; Charles, Z.; Cormode, G.; Cummings, R.; et al. Advances and Open Problems in Federated Learning. *Found. Trends Mach. Learn.* **2021**, *14*, 1–210. [CrossRef]
2. Kim, T.; Lin, E.; Lee, J.; Lau, C.; Mugunthan, V. Navigating Data Heterogeneity in Federated Learning: A Semi-Supervised Federated Object Detection. In Proceedings of the Advances in Neural Information Processing Systems, New Orleans, LA, USA, 10–16 December 2023.
3. Li, Y.; Wang, Q.; Xu, M.; Clarkson, M. UltraFlwr: An Efficient Federated Surgical Object Detection Framework. *arXiv* **2025**, arXiv:2503.15161.

4. Zhang, X.; Zhang, B.; Yu, W.; Kang, X. Federated Deep Learning with Prototype Matching for Object Extraction from Very-High-Resolution Remote Sensing Images. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5603316. [CrossRef]
5. Karimireddy, S.; Kale, S.; Mohri, M.; Reddi, S.; Stich, S.; Suresh, A.T. SCAFFOLD: Stochastic Controlled Averaging for Federated Learning. In Proceedings of the International Conference on Machine Learning, Virtual, 13–18 July 2020; pp. 5132–5143.
6. Li, T.; Sahu, A.K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; Smith, V. Federated Optimization in Heterogeneous Networks. In Proceedings of the Machine Learning and Systems, Austin, TX, USA, 2–4 March 2020; pp. 429–450.
7. Gu, T.; Dolan-Gavitt, B.; Garg, S. BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain. *arXiv* **2017**, arXiv:1708.06733.
8. Wang, H.; Sreenivasan, K.; Rajput, S.; Vishwakarma, H.; Agarwal, S.; Sohn, J.-Y.; Lee, K.; Papailiopoulos, D. Attack of the Tails: Yes, You Really Can Backdoor Federated Learning. In Proceedings of the Advances in Neural Information Processing Systems, Virtual, 6–12 December 2020.
9. Xie, C.; Huang, K.; Chen, P.; Li, B. DBA: Distributed Backdoor Attacks Against Federated Learning. In Proceedings of the International Conference on Learning Representations, Addis Ababa, Ethiopia, 26–30 April 2020.
10. Zhang, Z.; Panda, A.; Song, L.; Yang, Y.; Mahoney, M.W.; Gonzalez, J.E.; Ramchandran, K.; Mittal, P. Neurotoxin: Durable Backdoors in Federated Learning. In Proceedings of the International Conference on Machine Learning, Baltimore, MD, USA, 17–23 July 2022; pp. 26429–26446.
11. Bagdasaryan, E.; Veit, A.; Hua, Y.; Estrin, D.; Shmatikov, V. How to Backdoor Federated Learning. In Proceedings of the International Conference on Artificial Intelligence and Statistics, Online, 26–28 August 2020; pp. 2938–2948.
12. Blanchard, P.; El Mhamdi, E.; Guerraoui, R.; Stainer, J. Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 119–129.
13. El Mhamdi, E.; Guerraoui, R.; Rouault, S. The Hidden Vulnerability of Distributed Learning in Byzantium. In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 3521–3530.
14. Pillutla, K.; Kakade, S.; Harchaoui, Z. Robust Aggregation for Federated Learning. *IEEE Trans. Signal Process.* **2022**, *70*, 1142–1154. [CrossRef]
15. Yin, D.; Chen, Y.; Ramchandran, K.; Bartlett, P. Byzantine-Robust Distributed Learning: Towards Optimal Statistical Rates. In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 5650–5659.
16. Cao, X.; Fang, M.; Liu, J.; Gong, N. FLTrust: Byzantine-Robust Federated Learning via Trust Bootstrapping. In Proceedings of the Network and Distributed System Security Symposium, Virtual, 21–25 February 2021.
17. Fung, C.; Yoon, C.; Beschastnikh, I. The Limitations of Federated Learning in Sybil Settings. In Proceedings of the International Symposium on Research in Attacks, Intrusions and Defenses, San Sebastián, Spain, 14–16 October 2020; pp. 301–316.
18. Lin, Y.; Zhou, P.; Wu, Z.; Liao, Y. Mitigating Backdoors in Federated Learning with FLD. *arXiv* **2023**, arXiv:2303.00302.
19. Zhang, Z.; Cao, X.; Jia, J.; Gong, N. FLDetector: Defending Federated Learning Against Model Poisoning Attacks via Detecting Malicious Clients. In Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 14–18 August 2022; pp. 2545–2555.
20. Nguyen, T.D.; Rieger, P.; Chen, H.; Yalame, H.; Möllering, H.; Fereidooni, H.; Marchal, S.; Miettinen, M.; Mirhoseini, A.; Zeitouni, S.; et al. FLAME: Taming Backdoors in Federated Learning. In Proceedings of the USENIX Security Symposium, Boston, MA, USA, 10–12 August 2022; pp. 1415–1432.
21. Rieger, P.; Nguyen, T.; Miettinen, M.; Sadeghi, A. DeepSight: Mitigating Backdoor Attacks in Federated Learning Through Deep Model Inspection. In Proceedings of the Network and Distributed System Security Symposium, San Diego, CA, USA, 24–28 April 2022.
22. Molina-Coronado, B. Client-Side Patching Against Backdoor Attacks in Federated Learning. *arXiv* **2024**, arXiv:2412.10605.
23. Zhang, H.; Liu, J.; Jia, J.; Gong, N. Test-Time Backdoor Detection for Object Detection Models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 11–15 June 2025.
24. Li, S.; Dai, Y. BackdoorIndicator: Leveraging OOD Data for Proactive Backdoor Detection in Federated Learning. In Proceedings of the USENIX Security Symposium, Philadelphia, PA, USA, 14–16 August 2024; pp. 4193–4210.
25. Yue, G.; Han, X. Based on Model Randomization and Adaptive Defense for Federated Learning Schemes. *Sci. Rep.* **2025**, *15*, 6613. [CrossRef] [PubMed]
26. Zhang, K.; Tao, G.; Xu, Q.; Cheng, S.; An, S.; Liu, Y.; Feng, S.; Shen, G.; Chen, P.-Y.; Ma, S.; et al. FLIP: A Provable Defense Framework for Backdoor Mitigation in Federated Learning. In Proceedings of the International Conference on Learning Representations, Kigali, Rwanda, 1–5 May 2023.
27. Abadi, M.; Chu, A.; Goodfellow, I.; McMahan, H.B.; Mironov, I.; Talwar, K.; Zhang, L. Deep Learning with Differential Privacy. In Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, 24–28 October 2016; pp. 308–318.

28. Andrew, G.; Thakkar, O.; McMahan, H.; Ramaswamy, S. Differentially Private Learning with Adaptive Clipping. In Proceedings of the Advances in Neural Information Processing Systems, Virtual, 6–14 December 2021.
29. Cui, L.; Wu, X. ALDP-FL for Adaptive Local Differential Privacy in Federated Learning. *Sci. Rep.* **2025**, *15*, 26679. [CrossRef] [PubMed]
30. Truex, S.; Baracaldo, N.; Anwar, A.; Steinke, T.; Ludwig, H.; Zhang, R.; Zhou, Y. A Hybrid Approach to Privacy-Preserving Federated Learning. In Proceedings of the ACM Workshop on Artificial Intelligence and Security, London, UK, 11–15 November 2019; pp. 1–11.
31. Wei, K.; Li, J.; Ding, M.; Ma, C.; Yang, H.H.; Farokhi, F.; Jin, S.; Quek, T.Q.S.; Poor, H.V. Federated Learning with Differential Privacy: Algorithms and Performance Analysis. *IEEE Trans. Inf. Forensics Secur.* **2020**, *15*, 3454–3469. [CrossRef]
32. Yan, J.; Zhou, N.; Cheng, Y.; Zhang, F.; Wang, H.; Wang, M.; Jin, B.; Li, M.; Lu, Q.; Zhang, W. Application of Machine-Vision-Driven Physics-Informed Neural Networks in Pantograph–Catenary System State Detection. *Mech. Syst. Signal Process.* **2026**, *257*, 114577. [CrossRef]
33. Wang, X.; Jiang, H.; Dong, Y.; Mu, M. Spatial-Channel Collaborative Multi-Scale Graph Interaction Deep Transfer Learning for Unsupervised Rotating Machinery Fault Diagnosis. *Eng. Appl. Artif. Intell.* **2026**, *176*, 114691. [CrossRef]
34. Chan, S.; Dong, Y.; Zhu, J.; Zhang, X.; Zhou, J. BadDet: Backdoor Attacks on Object Detection. In Proceedings of the European Conference on Computer Vision Workshops, Tel Aviv, Israel, 23–24 October 2022; pp. 396–412.
35. McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; Arcas, B. Communication-Efficient Learning of Deep Networks from Decentralized Data. In Proceedings of the International Conference on Artificial Intelligence and Statistics, Fort Lauderdale, FL, USA, 20–22 April 2017; pp. 1273–1282.
36. Dwork, C.; Roth, A. The Algorithmic Foundations of Differential Privacy. *Found. Trends Theor. Comput. Sci.* **2014**, *9*, 211–407. [CrossRef]
37. Zhao, B.; Mopuri, K.; Bilen, H. iDLG: Improved Deep Leakage from Gradients. *arXiv* **2020**, arXiv:2001.02610.
38. Geiping, J.; Bauermeister, H.; Dröge, H.; Moeller, M. Inverting Gradients—How Easy Is It to Break Privacy in Federated Learning? In Proceedings of the Advances in Neural Information Processing Systems, Virtual, 6–12 December 2020; pp. 16937–16947.
39. Kairouz, P.; Oh, S.; Viswanath, P. The Composition Theorem for Differential Privacy. *IEEE Trans. Inf. Theory* **2017**, *63*, 4037–4049. [CrossRef]
40. Mironov, I. Rényi Differential Privacy. In Proceedings of the IEEE Computer Security Foundations Symposium, Santa Barbara, CA, USA, 21–25 August 2017; pp. 263–275.
41. Cheng, G.; Han, J. A Survey on Object Detection in Optical Remote Sensing Images. *ISPRS J. Photogramm. Remote Sens.* **2016**, *117*, 11–28. [CrossRef]
42. Li, K.; Wan, G.; Cheng, G.; Meng, L.; Han, J. Object Detection in Optical Remote Sensing Images: A Survey and a New Benchmark. *ISPRS J. Photogramm. Remote Sens.* **2020**, *159*, 296–307. [CrossRef]
43. Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 740–755.
44. Karimi, H.; Nutini, J.; Schmidt, M. Linear Convergence of Gradient and Proximal-Gradient Methods Under the Polyak–Łojasiewicz Condition. In Proceedings of the Machine Learning and Knowledge Discovery in Databases, Riva del Garda, Italy, 19–23 September 2016; pp. 795–811.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.