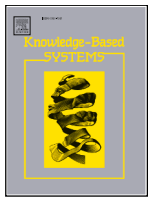
Knowledge-Based Systems 339 (2026) 115650

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys



DualVeil: Persistent and invisible backdoor attacks in federated learning via dual optimization

Maozhen Zhang [a], Mengnan Zhao [b], Wei Wang [c], Bo Wang [a,*]

[a] School of Information and Communication Engineering, Dalian University of Technology, Dalian, China
[b] School of Computer Science and Technology, Dalian University of Technology, Dalian, China
[c] Institute of Automation, Chinese Academy of Sciences, Beijing, China

ARTICLE INFO

Keywords:
Backdoor attack
Federated learning
Attack persistence

ABSTRACT

Federated learning is gaining attention owing to its capability to protect data privacy in distributed environments. However, its decentralized architecture makes it susceptible to backdoor attacks. Although various backdoor injection methods have been proposed, achieving trigger stealthiness and backdoor attack persistence remains challenging. We address such issues by proposing DualVeil, a novel attack framework that jointly optimizes persistent parameter maximization optimization and triggers invisibility minimization optimization. Specifically, persistent parameter maximization optimization enhances attack persistence using historical gradients and stable parameters to develop constrained updates within the parameter hypersphere. Meanwhile, the trigger invisibility minimization optimization module refines the trigger mechanism via feedback-guided optimization, thereby reducing its perturbation. Extensive experiments demonstrate that DualVeil can effectively inject a covert backdoor mapping into the global model without compromising its main-task performance. Comparative studies of the state-of-the-art backdoor attacks across five aggregation schemes further validate the superior attack efficacy of DualVeil. Our code is available at https://github.com/Maozhen-Zhang/DualVeil.git.

1. Introduction

Federated learning (FL), renowned for its distributed and privacy-preserving nature, enables collaborative model training while maintaining data locality [1–3]. FL paradigm has been successfully applied to diverse domains, such as autonomous driving [4], intelligent healthcare [5], and financial fraud detection [6]. However, the same decentralized and privacy-preserving characteristics that underpin the success of FL introduce inherent security vulnerabilities. In particular, malicious clients can exploit the lack of centralized visibility to perform adversarial manipulations during local training, resulting in threats such as backdoor attacks that compromise the integrity of the global model.

Backdoor attacks stealthily implant specific input–output mappings into a trained model [7–19]. During training, attackers embed a trigger pattern into selected samples and assign them to a target label, establishing a covert association between the trigger and the malicious output. Once deployed, the trigger functions as an activation signal that induces malicious behavior when present, while the model functions normally on benign inputs. When extended to the federated setting, such attacks render the aggregated global model unsafe, as the hidden malicious behavior can propagate through global updates without direct server supervision.

In recent studies, despite the successful demonstration of backdoor injection in FL models, two fundamental challenges remain: extending the backdoor lifespan and enhancing the stealthiness of backdoor triggers. Zhang et al. [20] addressed these issues by proposing Neurotoxin, which prolongs backdoor persistence by selectively updating the top $k\%$ gradients of local models. Similarly, F3BA [21] adopts distributed optimization to adaptive triggers and fine tunes model parameters, improving attack success rates. However, such methods heavily rely on visually conspicuous patch-based triggers, making them susceptible to manual inspection and defensive sanitization during backdoor activation.

Recently, Nguyen et al. [22] proposed IBA, an invisible backdoor attack that leverages a UNet-based generator to enerate subtle input perturbations as triggers, overcoming the aforementioned limitation. IBA produces distinct triggers for each image to increases trigger diversity and ties backdoor activation closely to the generator. Nevertheless, the efficacy of backdoor attacks in FL is strongly influenced by client population and attacker data volume. In federated settings with many client

* Corresponding author.
E-mail addresses: maozhenzhang@mail.dlut.edu.cn (M. Zhang), dlutzmn@mail.dlut.edu.cn (M. Zhao), wei.wong@ia.ac.cn (W. Wang), bowang@dlut.edu.cn (B. Wang).

https://doi.org/10.1016/j.knosys.2026.115650
Received 5 May 2025; Received in revised form 8 December 2025; Accepted 27 February 2026
Available online 1 March 2026

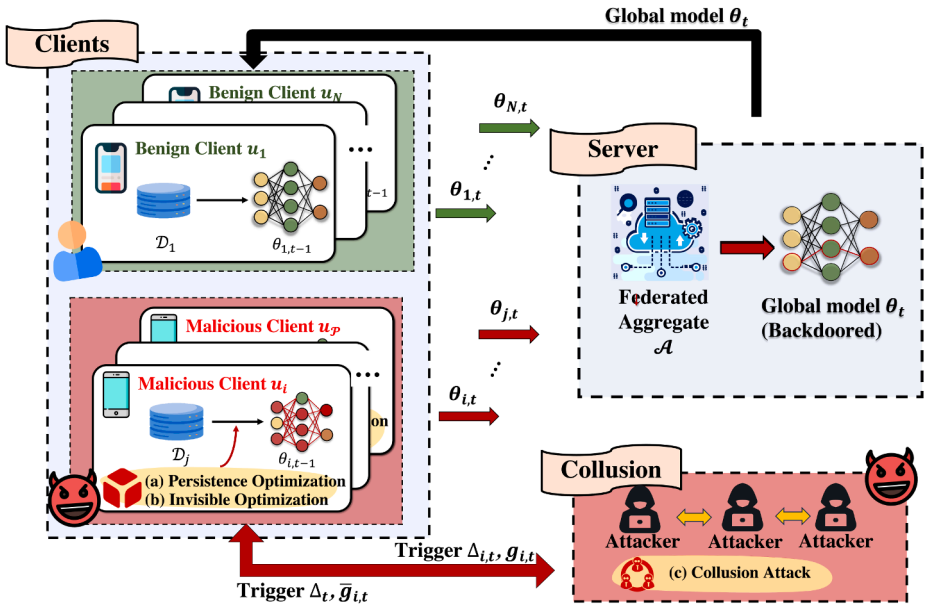


Fig. 1. Clients upload their locally trained model weights to the server, which subsequently updates the global model and redistributes the updated weights to local clients. In this process, the *green region* represents the local training conducted by benign clients, while the *red region* represents that of malicious clients. Within the red region, labels (a), (b), and (c) indicate the specific stages where the DualVeil pipeline operates.

participates but limited per-client data, attackers encounter considerable obstacles in injecting backdoors. Herein, we focus on developing invisible backdoor attacks with long-term persistence for scenarios that involve numerous terminal nodes, each possessing limited local data.

The background of this study is depicted in Fig. 1, where a small subset of clients are malicious and only upload model parameters to the central server. Under this threat model, the primary objectives of this study are threefold: (1) achieving a high success rate of attacks, (2) ensuring the invisibility of backdoor triggers, and (3) extending the persistence of backdoors in the global model. To this end, we propose DualVeil, a dual-optimization backdoor attack framework. DualVeil incorporates an internal minimization constraint that generates imperceptible triggers to enable stealthy activation. Meanwhile, a parameter maximization mechanism reinforces the persistence of injected backdoor within the aggregated global model. In addition, DualVeil facilitates collusion among malicious clients during aggregation, enabling attackers to exploit shared gradient information to analyze vulnerabilities and further enhance attack efficacy, as illustrated in Fig. 1.

In summary, our main contributions are threefold:

1. We introduce *DualVeil*, an optimization-driven framework that combines parameter maximization with trigger invisibility minimization to produce imperceptible backdoors with high persistence while preserving the accuracy of the main task,
2. DualVeil coordinates collusion among malicious clients to overcome challenges posed by large client pools and limited attacker data,
3. Extensive experiments demonstrate the effectiveness of DualVeil in challenging scenarios, such aswith many local clients, a small fraction of malicious participants, and limited data per client. Moreover, we evaluate its robustness against popular defense mechanisms, demonstrating that DualVeil exhibits superior attack strength, stealth, and persistence.

2. Related work

2.1. Defense under federated learning

Federated learning [1,23], as a distributed architecture, faces inherent security from potentially malicious clients. To mitigate these challenges, a variety of defense mechanisms have been proposed, leveraging the characteristics of distributed training. Among these, Deepsight [24] mitigates backdoor attacks by detecting poisoned models through the fine-grained analyses of neural network parameters and outputs. FoolsGold [25] defends against malicious updates by dynamically adjusting learning rates based on the similarity of client contributions. Similarly, RLR [26] performs statistical analysis on the directional changes of model parameter updates, requiring only minimal modifications to the FedAvg. Mkrum [27] enhances robustness through distance-based aggregation, selecting local models that are closest to their neighbors. Nevertheless, despite their effectiveness in mitigating certain attacks, these defenses remain vulnerable to our proposed DualVeil attack, which achieves a high attack success rate while maintaining stealth. By employing model constrained optimization, DualVeil reduces the detectability of malicious models under defense scrutiny. Moreover, since the backdoor trigger is optimized without altering the local training process, DualVeil effectively evades detection by existing defense mechanisms.

2.2. Backdoor attack

Backdoor attacks have attracted considerable attention due to the growing deployment of deep neural networks and their inherent security risks [28–30]. Traditional backdoor methods typically rely on manually designed triggers (e.g., local patches) paired with target labels to implant malicious mappings [13,31,32]. Because such explicit triggers often produce perceptible differences between poisoned and clean inputs, research has shifted toward stealthier approaches, including invisible triggers [33,34] and trigger-optimization techniques [35,36]. In this work, we further explore this direction in the federated learning setting: by coordinating colluding malicious clients, we jointly optimize model constraints and trigger patterns to inject Trojan backdoors while preserving benign utility and minimizing detectability.

2.3. Backdoor attack in federated learning

Recent work has focused on optimizing trigger design to increase attack success. For example, F3BA [21] and A3FL [37] obtain high success rates by searching or learning effective trigger patterns. However, these

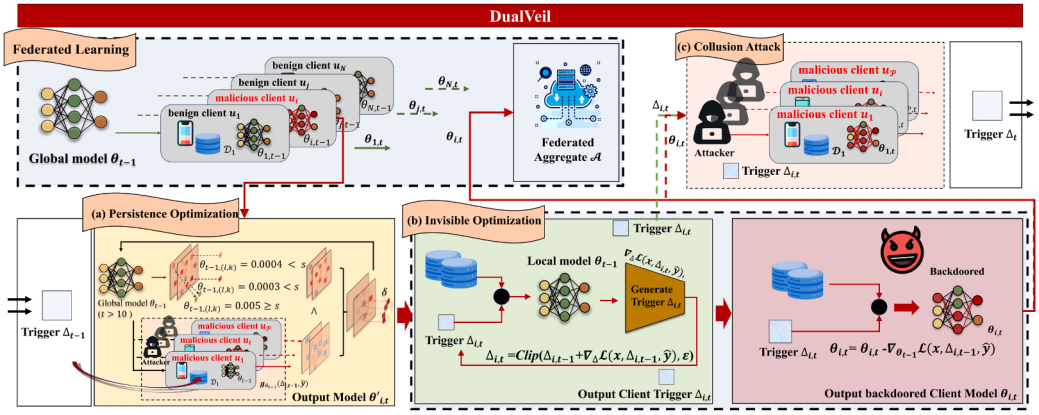


Fig. 2. DualVeil framework comprises three components: (a) *Persistence Optimization*, which maximizes the backdoor parameters based on stable backdoor parameter coordinates obtained, (b) **Invisible Optimization**, which generates local invisible triggers and injects them into the local model, and (c) **Collusion Attack**, which evaluates local triggers based on the prior knowledge of the client and selects the local trigger with the best backdoor activation performance as the global trigger.

methods often introduce perceptible differences between poisoned and clean samples. To improve stealthiness, approaches such as IBA [22] propoe invisible triggers by crafting unique perturbations per backdoor sample. Although IBA attains strong invisibility, its sample-specific optimization requires prior access to target samples at activation time, which limits practical deployment.

Motivated by these limitations, and by the observation that stealthy methods degrade substantially when local datasets are small or when the number of participating clients is large, we propose generating a single, dataset-level invisible trigger. The resulting triggered images are visually indistinguishable from clean images while enabling effective backdoor implantation under stricter federated constraints.

3. Method

3.1. Preliminaries

Federated Learning Setup. Consider a standard federated learning setup [38–40] for an image classification task involving a set of participating clients, denoted by $\mathcal{N}$. Let u_i represent the ith client, where $u_i \in \mathcal{N}$. Each client u_i possesses a private local dataset $\mathcal{D}_i$, and the number of clients is denoted as $|\mathcal{N}|$. During the tth communication round, client u_i performs local training for serveral epochs to obtain a local model $\theta_{i,t}$. The client then communicates with the central server, which aggregates the received local updates to produce the global model θ_t as follows:

$$\theta_t = \theta_{t-1} + \mathcal{A}(\theta_{i,t} - \theta_{t-1} | u_i \in \mathcal{S}), \tag{1}$$

where, $\mathcal{A}$ denotes the server's aggregation function, and $\mathcal{S}$ is the subset of clients randomly selected by the server for participation in this round. After aggregation, the updated global model θ_t is distributed to all clients for the next training round.

Capability of Attacker. We assume a set of malicious clients $\mathcal{P} \subseteq \mathcal{N}$ that can communicate among themselves to coordinate the attack (e.g., negotiate attack parameters). Each malicious client $u_p \in \mathcal{P}$ possesses a private local dataset $\mathcal{D}_p$ and has complete control over its local training procedure. Although attackers are aware of their own local training details, they do not have prior knowledge regarding the server's aggregation function $\mathcal{A}$ nor the subset of clients $\mathcal{S}$ selected for aggregation in each round.

Attacker's Goal. The attackers aim to manipulate local updates by injecting backdoor trigger Δ into the global model θ. The trigger is designed in a way that, during the inference phase, the global model θ outputs a predefined target prediction $\hat{y}$ for a triggered input $\hat{x}$:

$$\hat{y} = \mathcal{F}_\theta(\hat{x}), \quad \hat{x} = x + \Delta,$$

where x denotes a clean input and $\hat{x}$ denotes the same input stamped with trigger Δ. Concurrently, the attackers strive to preserve the model's utility on clean data, i.e

$$y = \mathcal{F}_\theta(x)$$

The attackers pursue three objectives: (1) high attack success rate, (2) trigger invisibility (imperceptibility), and (3) persistence of the backdoor across training rounds and aggregation.

3.2. Overview of DualVeil

DualVeil is designed to inject a backdoor into a FL framework by leveraging collusion among malicious clients. It jointly maximizes the persistence of backdoor-related parameters θ while minimizing the visual visibility (imperceptibility) of the trigger Δ. DualVeil achieves this using two complementary optimization modules: Persistence Optimization (PO), which strengthens the longevity of the injected backdoor parameters and Invisible Optimization (IO), which refines an imperceptible trigger. These modules are coordinated via interattacker communication (Collusion, CA).

Concretely, DualVeil targets two simultaneous objectives for the backdoor attack: (1) trigger visual imperceptibility (that is, ensuring the trigger is difficult to detect by eye), and (2) backdoor persistence (that is, ensuring the malicious effect persists across multiple federated training rounds). We conceptualize the backdoor as a computational relationship between an input trigger and model parameters. For the first objective, modifying the trigger to enhance visual imperceptibility can reduce the correlation between the trigger and backdoor-related parameters, thereby weakening the formation of a stable backdoor mapping in subsequent training rounds. To avoid the limits imposed by trigger capacity (for example, on CIFAR-10 a trigger occupies at most $3 \times 32 \times 32$ pixels), DualVeil shifts part of its design focus to model parameters: during early local training, we deliberately steer a subset of parameters to become correlated with the trigger while keeping the trigger visually imperceptible. By jointly constraining trigger perturbations and aligning parameter updates with the gradient direction induced by the trigger, DualVeil enhances backdoor persistence without sacrificing imperceptibility.

The overall pipeline of DualVeil is illustrated in Fig. 2, and the main contributions are summarized below:

- **Persistence Optimization (PO).** PO performs targeted model poisoning by selectively modifying neuron parameters to increase the persistence of the backdoor within the global model. This makes the implanted backdoor more resistant to statistical defenses and aggregation-induced dilution.

- **Invisible Optimization (IO).** IO jointly optimizes the trigger Δ and local model parameters by minimizing the cross-entropy loss on both clean and poisoned samples while enforcing an imperceptibility constraint on Δ. This enables effective backdoor activation without considerably declining clean-task performance.
- **Collusion (CA).** During PO and IO, malicious clients communicate and exchange attack-relevant information (for example, candidate triggers, validation metrics, or gradient summaries, etc.). Such coordination enhances both the optimization of backdoor persistence and the generation and selection of imperceptible triggers.

The detailed DualVeil procedure is outlined in Algorithm 1. As DualVeil does not alter the training procedure of benign clients, we only describe the operations executed by malicious clients. At each communication round, upon receiving the global model θ_{t-1}, each malicious client first performs Persistence Optimization , including an attacker communication step (Eq. (4)). Subsequently, Invisible Optimization refines the trigger. A subsequent intraattacker communication step (see Eq. (13)) is used to evaluate and select the best trigger. Finally, the malicious client uploads its locally updated model to the server for aggregation.

Algorithm 1: DualVeil.

Input: Malicious clients $u_i, i \in \mathcal{P}$, global model θ_{t-1}, previous trigger Δ_{t-1}, step size for backdoor parameter updates δ, empirically set threshold for selecting backdoor parameters s and each client i private dataset $\mathcal{D}_i$

Output: Local model $\theta_{i,t}, \Delta_t$,

Function *Main*:
  for u_i *in* $\mathcal{P}$ **do**
    $\theta'_{i,t} \leftarrow$ **PO** $(\theta_{t-1}, \delta, s, \mathcal{D}_i)$;
    $\theta_{i,t}, \Delta_{i,t} \leftarrow$ **IO** $(\theta'_{i,t}, \mathcal{D}_i, \Delta_{t-1})$;
  $Q_1, Q_2, ., Q_{|\mathcal{P}|} \leftarrow$ Eq. (13);
  $\hat{i} = \arg\max_{i \in \mathcal{P}}(Q_i)$ //Collusion ;
  $\Delta_t \leftarrow \Delta_{\hat{i},t}$;
  return $\theta_{i,t}, \Delta_t$;
Function *Persistence Optimization (PO)*:
  $\Theta_{\theta_{t,k}<s} \leftarrow$ Eq. (2) $\sim$ (3) ;
  for u_i *in* $\mathcal{P}$ **do**
    $\overline{g}_t \leftarrow$ Eq. (4) ;
  $\Theta_{backdoor} \leftarrow$ Eq. (5) $\sim$ (6) ;
  $\theta'_{i,t} = \theta_{i,t-1} - \delta \cdot sign(\nabla_\theta \mathcal{L}(x, \hat{y}, \Delta)) \cdot \Theta_{backdoor}$;
  return $\theta'_{i,t}$;
Function *Invisible Optimization (IO)*:
  for e *in* E **do**
    for $\mathcal{B}_j$ *in* $\mathcal{D}_i$ **do**
      $\Delta_{i,t} = \Delta_{i,t-1} - \eta \nabla_{\Delta_{i,t-1}} \mathcal{L}(\theta'_{i,t}, \mathcal{B}_j, \hat{y})$;
  for e *in* E **do**
    for $\mathcal{B}_j$ *in* $\mathcal{D}_i$ **do**
      $\theta_{i,t} = \theta'_{i,t} - \lambda \nabla_\theta \mathcal{L}(\mathcal{B}_j, \hat{y})$;
  return $\theta_{i,t}$;

3.3. Persistence optimization

Previous studies [20,41] have shown that the persistence of backdoor attacks depends on whether backdoor-related parameters are overwritten by model updates associated with the main task. Building upon this observation, we introduce a novel optimization strategy that integrates parameter selection with malicious client collusion in FL. Unlike prior approaches that are focused on single-client training and solely rely on local data to evaluate backdoor persistence, our proposed method enables collaborative parameter selection among multiple malicious clients. This collaboration enhances adaptability and robustness by mitigating the limitations of isolated evaluations. Specifically, persistence optimization is achieved by jointly optimizing historical stable parameters and current stable parameters,which resulting in a persistently optimized model θ'_t. By leveraging collusion among malicious clients, the proposed approach remains effective even in large-scale FL scenarios that involve numerous participants.

History Stable Parameters. DualVeil identifies a set of stable parameter coordinates Θ_θ whose update amplitudes remain small across the update of the historical global model set $\mathcal{W} = \{\theta_{t-10}, \dots \theta_{t-1}\}$. Specifically, during each federated communication round, we record the global model updates within a sliding time window sized w and maintain a collection of historical global models $\mathcal{W}$. Once the number of stored models in $\mathcal{W}$ reaches a predefined limit, we maintain a collection of historical global models in $\mathcal{W}$. The parameter variations across all historical models are evaluated before local training begins. Let Θ denote the coordinate indices of the global model parameters θ, where $\theta_{l,k}$ represents the kth neuron parameter in the lth layer of the model. For each parameter coordinate (l, k), $\theta_{\mu,(l,k)}$ and $\theta_{\sigma,(l,k)}$ denote the mean and standard deviation of $\theta_{l,k}$, respectively, which is computed over all models within $\mathcal{W}$. For the convenience of reading, we omit the subscripts l and k in the following derivations. The corresponding formulation is as follows:

$$\begin{aligned} \theta_\mu &= \frac{\sum_t^{\mathcal{W}} \theta_{t,(k,l)}}{|\mathcal{W}|}, \\ \theta_\sigma &= 1/|\mathcal{W}| \sqrt{\sum_t^{|\mathcal{W}|} (\theta_t - \theta_\mu)}, \end{aligned} \tag{2}$$

where $|\mathcal{W}|$ denotes the number of models contained in the historical model set $\mathcal{W}$. Based on the standard deviation of each global model parameter coordinate computed above, we identify the parameter coordinates whose update magnitudes remain within a small deviation range, as formulated below:

$$\Theta_{\theta_{l,k} \le s} = \mathbb{1}(\theta_{\sigma,(l,k)} < s), \tag{3}$$

where $\Theta_{\theta_{l,k} \le s}$ denotes the result of the indicator function $\mathbb{1}(\cdot)$ applied to $\theta_{\sigma,(l,k)}$ with a threshold s, which is an empirically set threshold. This threshold s is used to identify parameters whose variations remain within a stable range across the historical global models, which is represented by $\Theta_{\theta_{l,k} \le s}$. Thus, $\Theta_{\theta < s}$ thus represents the set of historically stable parameters within the window w. Subsequently, we shift our focus to the stable parameters in the current round of updates.

Current Stable Parameters. We consider the gradient g_i of the local model computed on the local dataset of each malicious client u_i. The gradient g_i reflects the magnitude of parameter updates contributed by a single client. DualVeil leverages collusion among malicious clients to jointly evaluate smaller gradient regions across their models to mitigate the influence of varying local data volumes. A smaller gradient indicates that the corresponding parameter has lower importance to the main task, and thus, it can be a reference to identify the current stable parameter coordinates Θ_g associated with the backdoor parameters. In the tth communication round, the average update gradient $\overline{g}_{t,(l,k)}$ of the global model parameters θ_{t-1} is computed as follows:

$$\overline{g}_{t,(l,k)} = \sum_i^{\mathcal{P}} \frac{g_{i,(l,k)}}{|\mathcal{P}|}, \tag{4}$$

where $\mathcal{P}$ denotes the set of malicious clients and $|\mathcal{P}|$ denotes its size. Eq. (4) computes the average gradient $\overline{g}_{t,(l,k)}$ for the (l, k)th parameter across all malicious clients. We then use this average gradient as a threshold to identify the coordinates of model parameters with small gradients:

$$\Theta_{g_t \le \overline{g}_t} = \mathbb{1}(g_{t,(l,k)} < \overline{g}_{t,(l,k)}), \tag{5}$$

where $\Theta_{g_t} \le \overline{g}_t$ represents the coordinates of model parameters with smaller gradient update magnitudes in the current round training. Upon

obtaining historical stable parameter coordinates $\Theta_{\theta<s}$ and the current stable parameter coordinates $\Theta_{g_t \leq \overline{g}_t}$, DualVeil leverages the update direction of the previous trigger sensitive parameters Δ_{t-1} to perform maximization-constrained optimization on the local malicious model.

Persistence Parameter Optimization. Persistence parameter optimization utilizes the stable parameter coordinates $\Theta_{\theta_{l,k}\leq s}$ and $\Theta_{g_t \leq \overline{g}t}$ obtained from the previous two steps as coordinate constraints to maximize the optimization of backdoor parameters. During federated aggregation, updates to the global model minimally impact on the historically stable parameters that correspond to $\Theta_{\theta_{l,k}\leq s}$. Similarly, during local training, the parameters associated with $\Theta_{g_t \leq \overline{g}_t}$ undergo relatively minor changes during local training. Using the local dataset $\mathcal{D}_i$, we minimize the interference with aligning the corrdinate poisitions of $\Theta_{\theta_{l,k}\leq s}$ and $\Theta_{g_t \leq \overline{g}t}$ to derive the coordinate set Θ_{backdoor}, which maximizes the persistence of the backdoor parameters. The formulation can be given as:

$$\Theta_{backdoor} = \Theta_{\theta_{l,k}\leq s} \wedge \Theta_{g_t \leq \overline{g}_t}, \tag{6}$$

where $\wedge$ denotes the logical AND operator, indicating that only parameters satisfying both the conditions are selected. Upon obtaining the candidate coordinate positions $\Theta_{backdoor}$, we compute the parameter update direction for the local model parameters using images stamped with the trigger:

$$sign(\nabla_{\theta_{t-1}} \mathcal{L}(x, \hat{y}, \Delta_{t-1}))$$

where $\mathcal{L}$ represents the loss function for the backdoor task. This update direction is a constraint for optimizing the backdoor parameters. We introduce an optimization control parameter δ to control the update magnitude. The change in backdoor parameters corresponding to $\Theta_{backdoor}$ can be formulated as follows:

$$\theta_t = \theta_{t-1} - \delta \cdot sign(\nabla_{\theta_{t-1}} \mathcal{L}(x, \hat{y}, \Delta_{t-1})) \cdot \Theta_{backdoor}, \tag{7}$$

where Δ_{t-1} denotes the trigger optimized in the $(t-1)$th communication round, $\hat{y}$ denotes the backdoor target label, and $\nabla_{\theta_{t-1}} \mathcal{L}$ represents the gradient of the model parameters. The function $sign(\cdot)$ denotes the sign operation, and δ denotes the step size for updating the backdoor parameters. The update is performed on a hypersphere centered at the current model parameters θ_{t-1}.

3.4. Invisible optimization

We design an internal minimization process that iteratively updates the trigger by reducing the discrepancy between clean and poisoned samples, enabling the derivation of an imperceptible trigger Δ. Similar to the persistence optimizer, this optimization is performed on the local dataset $\mathcal{D}_i$, ensuring that the trigger adapts to the client's data distribution. Specifically, the previous trigger Δ_{t-1} is optimized with respect to either the previous global model θ_{t-1} or the persistence-optimized model θ'_t, producing the updated trigger Δ_t depending on whether the current federated round employs persistence optimization. The optimized trigger Δ_t is subsequently embedded into the local model $\theta_{i,t}$, and the resultant backdoored model is uploaded to the server during communication round t.

Trigger Optimization. At the beginning of the attack, each malicious client randomly initializes its trigger $\Delta_{0,i}$ during the preparation phase, before federated training. In the tth communication round, malicious clients collaboratively perform invisible optimization on the previously shared trigger Δ_{t-1}, generating the updated trigger $\Delta_{i,t}$ for local training.

$$\Delta_{i,t} = \Delta_{t-1} - \eta \times \nabla_{\Delta_{i,t-1}} \mathcal{L}(\mathcal{D}_i, \hat{y}, \theta_i) + \mathcal{L}_{\mathcal{I}}(x, \hat{x}), \tag{8}$$

where $\Delta_{i,t}$ denotes the trigger optimized by client u_i during the tth communication round, η refers to the learning rate fortrigger optimization, and $\nabla_{\Delta_{i,t-1}} \mathcal{L}$ represents the gradient of the loss with respect to the trigger. $\mathcal{L}_{\mathcal{I}}$ denotes the ℓ_1 loss that measures the perceptual difference between the clean image x and the poisoned sample $\hat{x}$. The loss is defined as follows:

$$\mathcal{L}_{\mathcal{I}} = \frac{1}{H \times W} \sum_{i}^{H} \sum_{j}^{W} |x_{i,j} - \hat{x}_{i,j}|, \tag{9}$$

where H, W denote the height and width of the image, respectively.

Backdoor Inject. In this stage, each malicious client constructs poisoned (backdoor) training samples from its local dataset $\mathcal{D}_i$. We partition $\mathcal{D}_i$ into mini-batches $\mathcal{B}_j$, where $j = 1, 2, \ldots, |\mathcal{D}_i|/b$ and b denotes the batch size. For each batch $\mathcal{B}_j$, a fraction r (the poisoning rate) of samples are selected and stamped with the trigger to from the poisoned batch $\mathcal{B}_j^{posi}$. Let $S_j \subset \mathcal{B}_j$ be the set of selected sample swith $|S_j| = \lfloor r \cdot |\mathcal{B}_j| \rfloor$. Then

$$\mathcal{B}_j^{\text{pos}} = \left(\mathcal{B}_j \setminus S_j\right) \cup \{(x + \Delta, \hat{y}) \mid x \in S_j\},$$

where Δ denotes the trigger and $\hat{y}$ denotes the backdoor target label. The poisoned batches $\mathcal{B}_j^{\text{pos}}$ are used during local training to inject the backdoor into the model. The process of stamping the trigger onto a clean image is given by:

$$\hat{x} = \text{clip}\big(x + \Delta_{i,t}, \min(x), \max(x)\big), \tag{10}$$

where clip$(\cdot)$ enforces valid pixel bounds (e.g. $[0, 1]$ or $[0, 255]$ based on image normalization).

Each malicious client updates its local parameters by minimizing the training loss on the poisoned batch to inject the backdoor while preserving the main-task accuracy. Simply,

$$\min_{\theta_{i,t}} \mathcal{L}\big(\mathcal{B}_j^{\text{pos}}; \theta_{i,t}\big), \tag{11}$$

and a single gradient-descent update can be expressed as

$$\theta_{i,t} = \theta'_{i,t} - \lambda \nabla_\theta \mathcal{L}\big(\mathcal{B}_j^{\text{pos}}; \theta'_{i,t}\big), \tag{12}$$

where λ denotes the learning rate for model updates, and $\theta'_{i,t}$ denotes the initialization used for the local update (either the persistence optimized local model or the previous global model θ_{t-1}, depending on whether persistence optimization is applied and whether the historical window $\mathcal{W}$ has reached size w).

3.5. Collusion attack

The collusion process of DualVeil comprises two stages. The first stage occurs during the malicious client gradient aggregation step in Eq. (4). The second stage takes place immediately before server-side aggregation in the current communication round t. All malicious clients $u_i, i \in \mathcal{P}$ communicate before the global model aggregation to improve attack effectiveness. Each client shares the locally obtained candidate trigger $\Delta_{i,t}$ with the malicious clients set $\mathcal{P}$, and an optimal trigger for the current round is selected from these candidates. Speciffically, each malicious client u_i evaluates each candidate trigger $\Delta_{j,t}, j \in \mathcal{P}$ on its local dataset $\mathcal{D}_i$ and then shares the evaluation results with the other malicious clients. Each client contributes to the attack success rate of each candidate trigger on its local data. These values are accumulated across the malicious set to form an aggregate score Q_p for each candidate p.

$$Q_p = \sum_{i}^{\mathcal{P}} eval(\mathcal{D}_i, \Delta_{i,t}, \theta_t), p = 1, 2, \ldots, |\mathcal{P}|, \tag{13}$$

where $eval(\cdot)$ returns the backdoor task accuracy on a specified dataset and Δ_t denotes the model used for evaluation in round t. The candidate with the highest aggregate score is selected as the global trigger for round t:

$$\begin{aligned} &\hat{i} = \arg\max_{i \in \mathcal{P}}(Q_i), \\ &\Delta_t \leftarrow \Delta_{\hat{i},t}, \end{aligned} \tag{14}$$

Table 1
Detailed settings for federated learning.

Dataset	Model	Features	batch size b	lr	Dirichlet α	Epoch
CIFAR-10	VGG11	3× 32 × 32	64	0.01	0.9	2
CIFAR-10	ResNet18	3× 32 × 32	64	0.01	0.9	2
TinyImageNet	ResNet18	3× 64 × 64	64	0.01	0.9	2

where $\hat{i}$ indexes the trigger that achieves the maximal accumulated success across malicious clients and $\Delta_{\hat{i},t}$ is selected as the global trigger for this round.

4. Experiments

4.1. Experimental setup

The Detail of Federated Learning. In our federated learning setup, a total of $|\mathcal{N}| = 1000$ clients independently train their local models for a fixed number of epochs using their respective local dataset, and subsequnetly communicate with the server. During each communication round, the server randomly selects $|S| = 10$ client model updates from the 1000 clients to aggregate the global model. The aggregation step employs an aggregate learning rate $lr_{agg} = 0.5$ to control the update magnitutde of the global model. To evaluate performance under small-scale scenarios, we also conduct experiments with only $|\mathcal{N}| = 20$ clinets. Detailed experimental settings are summarized in Table 1.

Datasets. We consider two widely used benchmark datasets for image classification: 10-class color image dataset CIFAR-10 [42] and 200-class dataset TinyImageNet [43]. The CIFAR-10 dataset consistes of 50,000 training images and 10,000 testing images. with each image having a size of 32 × 32. TinyImageNet contains 100,000 images of 200 classes (500 for each class) downsized to 64 × 64 colored images. Each class has 500 training images, 50 validation images and 50 test images. For both datasets, we adopt a setting of non-IID data distribution. Unless otherwise specified, we use Dirichlet sampling with $\alpha = 0.9$ to simulate non-IID distributions. Among them, the degree of non-IID increases as α decreases.

Models and Training Rounds. We conduct experiments using VGG11 and ResNet18 on the two datasets. Specifically, both VGG11 and ResNet18 are evaluated on CIFAR-10, while only ResNet18 is used for TinyImageNet For experiments with 20 clients, models are trained from scratch for 400 federated communication rounds on CIFAR-10 and 100 rounds on TinyImageNet, and the results are recorded. For experiments with 1000 clients, we continue training a pre-trained models: the VGG11(pre-trained for 600 rounds) is further trained for 1400 additional rounds, totaling 2000 rounds. Similarly, the ResNet18 model (pre-trained for 1900 rounds) is further trained for 600 additional rounds, reaching a total of 2500 rounds.

For fair comparison, all comparative methods are evaluated under the same experimental settings: they use identical pre-training schedules, local-training hyperparameters, data partitions, and aggregation protocols. Specifically, for local training, to maintain attack consistency, malicious clients are trained with a learning rate of $lr_p = .002$ for 5 epochs, while benign clients are trained with $lr = 0.01$ for 2 epochs. If a baseline requires any special configuration, we explicitly describe it in the corresponding experimental section.

Baseline and Attack Setting. We compare DualVeil against five representative backdoor attacks: DBA [44], CerP [45], Neurotoxin [20], F3BA [21], and IBA [22]. For DBA, the trigger is partitioned into four complementary patches so that every group of four colluding malicious clients can assemble a complete trigger. For IBA, we adopt a UNet as the trigger generator, which is trained locally on a malicious client to produce sample specific invisible triggers.

For DualVeil, to minimize the impact of Persistence (parameter) Optimization on clean accuracy, we perform only 200 federated communication rounds dedicated to parameter optimization. The historical global-model window size is set to $w = 10$. Parameter-selection and update-control hyperparameters are set as $s = 5 \times 10^{-4}$ and $\delta = 1 \times 10^{-4}$, respectively. During trigger optimization, the trigger learning rate is $\eta = 0.01$, and the perturbation budget for the trigger is $\epsilon = 8/255$.

Moreover, during local training, the trigger is applied to 5 images per batch, with a batch size of $b = 64$. For ease of comparison, in the experiments reported in Tables 2 and 3, malicious clients use a learning rate of $lr_p = .002$ to train for $e = 5$ epochs. In other experiments, unless otherwise specified, both malicious and benign clients perform local training with the same learning rate $lr = 0.01$ and number of epochs $e = 2$. We emphasize that for federated learning with 20 clients, the model is trained from scratch. In contrast, for federated learning with 1000 clients, training is conducted on a pretrained model obtained after 1900 epochs in the federated learning setting. In this pretrained model, the configurations are the same as previously described, and since malicious clients do not need to inject backdoors, their local training follows the same settings as benign clients, using a learning rate of $lr = 0.01$ and $e = 2$ epochs.

Defense Method in Federated Learning. To evaluate the robustness of DualVeil, we consider five representative aggregation based defense mechanisms: FedAvg [1], Deepsight [24], FoolsGold [25], Mkrum [27], and RLR [26]. Each of these defenses employs distinct strategies to mitigate malicious updates in federated learning, as described in Section 2.1. Below, we provide the implementation details of these aggregation algorithms. In Deepsight, 256 noise images are randomly generated to simulate real inputs. The neuro activation differences between the global model and each local model are then evaluated to identify and filter out potentially malicious updates. In RLR, the symbolic threshold is set to $r = 2$. Specifically, the parameter directions (denoted as $+1$ or -1) for the same coordinate across clients are cumulatively summed. If the final sum exceed 2, the output direction is set to $+1$, otherwise, it is set to -1. In **Mkrum**, the server aggregates the global models of the first $|S| - f$ clients with the smallest ℓ_2-norm distances, where $|S|$ is the number of selected clients in each FedAvg communication round and f denotes the maximum number of malicious clients that Mkrum can tolerate.

Metric. The effectiveness of the proposed attack method is evaluated in terms of Accuracy (Acc) and Attack Success Rate (Asr). Accuracy measures is the proportion of clean samples that are correctly classified into their true categories in the test dataset. Asr represents the proportion of non-target samples that are misclassified into the specified target category after the trigger is added. Formally, Acc and the Acc and Asr are computed as follows:

$$\begin{aligned} Acc &= \frac{\sum^{\mathcal{D}_{test}} \mathbb{1}(\mathcal{F}(x_{clean}), y)}{|\mathcal{D}_{test}|} \times 100\%, \\ Asr &= \frac{\sum^{\mathcal{D}_{test/(x,\hat{y})}} \mathbb{1}(\mathcal{F}(x_{clean} + \Delta), \hat{y})}{|\mathcal{D}_{test/(x,\hat{y})}|} \times 100\%, \end{aligned} \tag{15}$$

where $\mathbb{1}(\cdot)$ denotes the indicator funciton and $\mathcal{F}$ denotes mapping of the Deep Nueral Network (DNN) from input to output. Here, x_{clean} denotes a clean input images, Δ is the backdoor trigger, y is the true label of an input, and $\hat{y}$ denotes the backdoor target for the triggered input $x_{clean} + \Delta$. Let $|\mathcal{D}_{test}|$ be the test dataset. We denote by $\mathcal{D}_{test/(x,\hat{y})}$ the subset of the test dataset obtained by excluding samples corresponding to the pair $(x, \hat{y})$ (i.e., samples of the backdoor target class used for evaluation). In addition to reporting Acc and Asr, we evaluate the persistence of the backdoor after the attack is stopped. The persistence is quantified by monitoring the ASR over subsequent training epochs; the decay

Table 2
Performance comparison between DualVeil and five state-of-the-art attack methods under 20 clients. Malicious clients locally train for 5 epochs with $lr_p = .002$, while benign clients train for 2 epochs with $lr = 0.01$. All models are trained from scratch for 400 rounds on CIFAR10 and 100 rounds on TinyImageNet.

Datasets	Models	Defenses	NoAtt	DBA		CerP		Neurotoxin		F3BA		IBA		DualVeil	
			Acc	Acc	Asr	Acc	Asr	Acc	Asr	Acc	Asr	Acc	Asr	Acc	Asr
CIFAR-10	VGG11	FedAvg	82.69	82.62	96.36	81.39	97.44	81.86	92.73	82.62	99.97	81.18	93.11	82.62	96.48
		Deepsight	82.07	82.32	98.38	77.56	97.08	80.49	97.81	82.24	100.00	80.34	94.02	81.77	92.76
		Foolsgold	82.47	82.67	96.58	81.21	97.78	82.04	95.28	83.22	100.00	80.24	93.78	82.92	98.67
		Mkrum	78.22	80.96	96.17	80.19	96.92	80.11	95.30	81.15	99.93	78.01	88.51	80.88	98.97
		RLR	80.52	82.14	96.37	81.85	95.41	82.34	94.03	82.53	100.00	80.69	96.97	81.90	98.30
	ResNet18	FedAvg	77.97	75.78	97.76	76.10	98.68	78.05	97.81	77.39	100.00	74.95	94.24	76.18	99.21
		Deppsight	77.81	77.45	98.73	76.20	83.37	77.73	97.06	77.23	100.00	74.09	98.36	77.58	99.25
		Foolsgold	76.83	77.05	98.67	76.61	91.13	77.96	97.27	77.99	100.00	74.47	98.86	77.22	99.70
		Mkrum	73.67	71.69	98.97	73.69	98.14	74.85	97.48	75.33	02.63	74.72	77.56	75.61	99.56
		RLR	77.26	76.20	97.53	76.52	98.54	77.92	96.16	77.46	100.00	72.33	98.42	76.85	98.82
Tiny-ImageNet	ResNet18	FedAvg	41.24	39.40	25.66	35.48	92.60	38.18	22.45	39.78	33.70	36.20	70.05	38.42	79.85
		Deppsight	39.32	38.54	14.45	32.78	94.93	38.98	24.24	38.72	12.36	37.32	69.12	38.24	74.01
		Foolsgold	38.61	36.72	19.15	33.30	92.74	38.34	21.66	37.34	21.42	36.92	82.35	38.46	79.65
		Mkrum	33.24	33.44	68.38	33.72	00.28	35.66	19.93	34.70	00.26	27.32	66.87	33.92	93.48
		RLR	34.40	32.94	58.67	34.14	86.09	36.88	13.76	38.04	34.07	35.16	75.13	34.06	87.49

Table 3
Performance comparison between DualVeil and five state-of-the-art attack methods under 1000 clients. Malicious clients locally train for 5 epochs with $lr_p = .002$, while benign clients train for 2 epochs with $lr = 0.01$. On VGG11, training continues from a pre-trained model after 600 rounds to 2000 rounds, and on ResNet18, from 1900 rounds to 2500 rounds. Compared with Table 2, the only differences lie in the increased number of clients and the use of models pre-trained for 1900 rounds under the same settings, which highlights the performance of DualVeil in large-scale federated learning.

Datasets	Models	Defenses	NoAtt	DBA		CerP		Neurotoxin		F3BA		IBA		DualVeil	
			Acc	Acc	Asr	Acc	Asr	Acc	Asr	Acc	Asr	Acc	Asr	Acc	Asr
CIFAR-10	VGG11	FedAvg	79.93	77.06	90.33	78.27	85.93	79.52	83.15	78.72	98.77	79.11	57.14	79.51	90.94
		Deppsight	79.30	79.33	91.03	79.18	84.62	84.47	87.62	78.81	99.30	79.04	50.25	79.85	86.93
		Foolsgold	78.15	78.93	84.44	78.16	86.78	83.46	88.95	79.43	98.83	79.34	14.50	78.61	84.96
		Mkrum	79.97	79.06	85.86	79.14	86.76	79.31	81.55	79.42	98.06	79.08	45.17	79.20	89.62
		RLR	78.00	79.06	88.97	79.20	84.73	80.16	80.43	78.21	98.37	79.32	48.23	79.35	90.76
	ResNet18	FedAvg	77.72	76.86	87.46	76.33	95.00	77.32	83.24	69.98	94.68	77.76	65.37	76.95	96.66
		Deppsight	77.66	76.56	88.78	76.41	94.17	77.05	82.54	69.35	80.53	76.74	57.90	76.89	97.36
		Foolsgold	76.83	77.05	26.93	76.97	77.64	77.07	79.81	69.06	81.35	76.86	26.98	77.49	89.16
		Mkrum	77.36	77.31	24.14	76.99	67.15	77.07	77.07	77.03	10.50	76.78	56.64	77.42	94.88
		RLR	77.57	76.94	77.50	77.04	93.55	77.12	80.81	71.28	91.24	77.52	51.74	77.06	96.56
Tiny-ImageNet	ResNet18	FedAvg	34.60	34.22	88.88	34.38	87.51	34.26	91.87	28.30	97.10	33.50	25.90	34.50	96.46
		Deppsight	33.96	33.28	88.44	33.38	85.92	33.50	90.53	32.82	99.59	33.78	40.50	33.58	97.80
		Foolsgold	33.92	33.66	87.59	33.90	00.58	33.16	91.33	32.42	99.61	34.48	21.04	35.28	97.28
		Mkrum	31.60	34.94	93.82	30.84	95.31	31.54	94.55	31.10	07.73	33.88	36.06	35.58	98.79
		RLR	33.90	33.54	90.81	33.28	92.16	33.24	89.10	33.46	99.95	32.82	27.31	33.24	98.23

Table 4
Performance comparison (mean ± std) of different attacks under varying numbers of clients.

Clients	Models	DBA		CerP		Neurotoxin		F3BA		IBA		DualVeil	
		Acc	Asr	Acc	Asr	Acc	Asr	Acc	Asr	Acc	Asr	Acc	Asr
20	VGG11	77.53 (±0.27)	98.27 (±0.04)	77.22 (±0.42)	98.14 (±0.23)	77.46 (±1.44)	97.34 (±0.26)	77.54 (±0.31)	99.77 (±0.39)	77.54 (±0.29)	95.15 (±5.67)	77.67 (±0.18)	98.36 (±0.16)
	ResNet18	75.84 (±0.60)	95.85 (±0.79)	75.67 (±0.70)	98.27 (±0.88)	72.28 (±1.87)	89.72 (±2.31)	65.07 (±6.53)	97.67 (±1.28)	75.38 (±0.06)	68.18 (±30.49)	76.23 (±0.94)	98.16 (±1.57)
1000	VGG11	81.85 (±1.06)	95.79 (±1.82)	82.24 (±0.07)	95.93 (±1.67)	82.78 (±0.11)	94.61 (±0.65)	82.19 (±0.22)	97.45 (±1.68)	82.05 (±0.99)	82.13 (±21.39)	82.86 (±0.37)	95.77 (±2.44)
	ResNet18	73.22 (±9.20)	92.05 (±3.27)	77.48 (±0.45)	88.09 (±2.44)	79.36 (±0.74)	87.82 (±1.20)	78.86 (±0.30)	96.18 (±2.80)	76.96 (±3.07)	80.15 (±4.96)	78.56 (±0.68)	93.87 (±0.61)

of the backdoor is characterized both by the reduction in ASR and by the number of epochs (or rounds) during which the backdoor remains effective above a chosen threshold.

4.2. Experimental results

Compared Attack Baselines. In Tables 2 and 3, we comprehensively compare DualVeil with five representative backdoor attacks under various aggregation strategies. Table 2 reports results for the small-scale setting with 20 clients, among which 20% are malicious. In each communication round, the server randomly selects 10 clients for aggregation. Under this setting, all attacks achieve over 90% Asr on the CIFAR-10, which is largely attributable to the relatively abundant local data held by each client that mitigates the decentralization effect. Additionally, in Table 4, we conduct multiple runs for different attacks under the FedAvg baseline, reporting the mean and standard deviation over three independent trials for each method and corresponding model architecture.

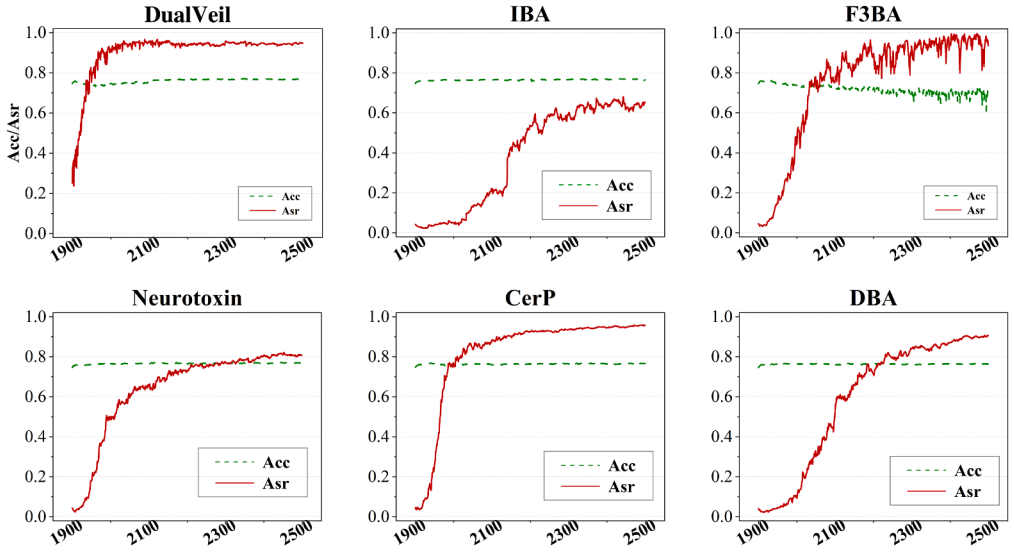


Fig. 3. Different attacks on the performance of FedAvg in each federated learning communication round. The performance of different attack method in FedAvg with each federated learning communication round. The malicious clients local train with $lr_p = .002$ for 5 epochs and use constraints $\epsilon = 8/255$ to generate invisible trigger Δ.

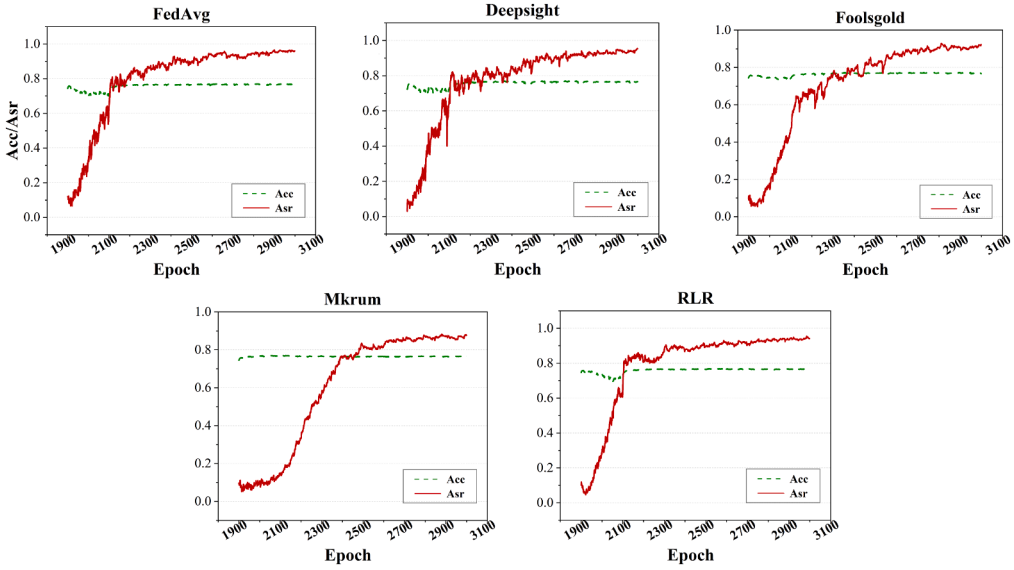


Fig. 4. DualVeil performance with an invisible trigger Δ under a constraint of $\epsilon = 4/255$ is evaluated under different defense mechanisms. Unlike the settings in Fig. 3, here we report the results obtained when both malicious and benign clients perform local training with a learning rate of $lr = 0.01$ for 2 epochs.

Notably, F3BA attains a particularly high Asr on CIFAR-10 by directly optimizing a fixed 3×3 patch as the trigger. However, such explicit trigger patterns are easily perceptible and therefore risk compromising stealth in practice.

In contrast, Table 3 presents a more challenging setting with 1000 clients, where 20% are malicious and only 10 clients are randomly selected per round. In this large-scale and highly decentralized regime, where each client holds far less data, the effectiveness of backdoor attacks generally degrades, as reflected by reduced Asr values.

Despite the increased difficulty, DualVeil consistently demonstrates strong attack performance across different model architectures and federated settings. Specifically, on CIFAR-10 with VGG11, DualVeil attains Asr values ranging from 84.96% to 98.97%, while with ResNet18, the range is 89.16% to 99.70%. This robustness arises from the optimization-driven stealthy trigger mechanism, which allows DualVeil to preserve high effectiveness and adaptability under diverse federated configurations.

Attack Performance Per Communication Round. Fig. 3 illustrates the performance of both the backdoor and main tasks during each communication round of federated learning under different attack methods. The results are evaluated in a setting with 20 clients, where malicious clients perform local training with a learning rate of $lr_p = .002$ for 5 epochs, and benign clients use a learning rate of $lr = 0.001$ for 2 epochs. The perturbation constraint $\epsilon = 8/255$ is applied to generate the imperceptible trigger Δ. As shown, DualVeil, F3BA, and CerP all achieve strong backdoor performance. However, compared to DualVeil, F3BA slightly impacts the main task after the attack converges, while CerP exhibits slower backdoor convergence. These results further highlight the superiority of DualVeil in backdoor attacks.

As shown in Fig. 4, under DualVeil with a trigger constraint of $\epsilon = 4/255$, Acc and Asr are evaluated across different aggregation algorithms. In this setting, malicious clients are trained for 2 epochs with a learning rate of $lr_p = .01$. This configuration differs from that in Fig. 3, where $\epsilon = 8/255$ was used. On one hand, the smaller trigger constraint

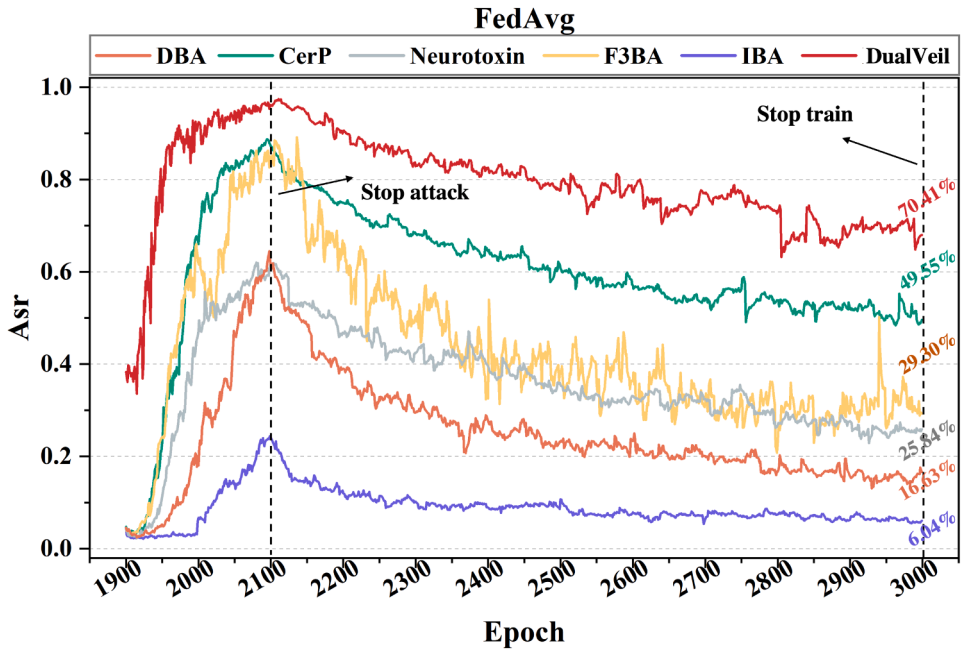


Fig. 5. The persistence of backdoor tasks. The malicious client stops attacking after 200 rounds and conducts normal training like a benign client (using clean data). In each attack method, the malicious clients train the local model for epochs $e = 5$ with a learning rate $lr_p = .002$. Additionally, DualVeil applies a constrain of $\epsilon = 8/255$ to the trigger.

reduces the strength of the trigger; on the other hand, the fewer local training epochs for malicious clients result in slower convergence of the backdoor attack.

We conclude that metrics related to DualVeil's Asr depend on the trigger constraint ϵ and the number of local training epochs performed by malicious clients. However, these factors also affect the stealthiness of the trigger samples and the concealment of the backdoor within the model.

Backdoor Task Persistence. In this experiment, we evaluate the persistence of backdoor behaviors induced by DualVeil. We start from a pre-trained model that has been trained for 1900 federated rounds. We then perform 200 rounds of backdoor attacks and subsequently continue training under benign (normal) conditions.

As shown in Fig. 5, after the attack is stopped the federated training proceeds. By round 3,000, among the six attack methods considered, DualVeil exhibits the smallest degradation in attack performance and maintains the highest Asr in the global model.

Table 5 reports the attack performance of the six methods both at the end of the attack phase and after continued benign training. After 800 rounds of benign training following the attack, DualVeil's ASR drops from 95.76% to 70.41%, a reduction of 25.45%. By contrast, although CerP and F3BA achieve relatively high ASRs immediately after the attack (e.g., 87.21% and 84.96%, respectively), their ASRs decrease by 37.66% and 55.66% during subsequent benign training.

These results demonstrate that collusion enabled parameter selection effectively identifies backdoor relevant parameters that are less tied to the main task. By maximizing and reinforcing such parameters, DualVeil substantially extends the persistence of the backdoor behavior, making the attack more durable under continued benign training.

These results demonstrate that the collusion based parameter selection effectively identifies backdoor relevant parameters that are less associated with the main task. By maximizing and reinforcing these parameters, DualVeil substantially extends the persistence of the backdoor behavior, making the attack more durable under continued benign training.

Invisible Trigger. In this section, we visualize and evaluate the effectiveness and stealthiness of the generated triggers. As shown in Fig. 7,

Table 5
2100 rounds and 3000 rounds of Asr (backdoor attack started in 1900 rounds, and training resumed normally after stopping the attack in 2100 rounds).

Dataset	Method	Attack End rounds 2100 round	Training End Round 3000 round
CIFAR-10	CerP	87.21%	49.55%
	BDA	61.34%	16.63%
	Neurotoxin	61.17%	25.84%
	IBA	21.82%	6.04%
	F3BA	84.96%	29.30%
	DualVeil	95.76%	70.41%

we present a visual comparison between clean and poisoned samples on the Tiny-ImageNet dataset. The first and third rows show the clean samples, while the second and fourth rows display the corresponding poisoned samples with the embedded triggers. The triggers generated by DualVeil are visually indistinguishable from the clean samples, clearly demonstrating their stealthiness.

To further investigate imperceptibility of the triggers. Fig. 8 visualizes the results on the CIFAR-10 dataset under a tighter perturbation constraint of $\epsilon = 4/255$. For each example, we sequentially present (from left to right) the clean image, its corresponding Grad-CAM heatmap, the poisoned image, and the Grad-CAM of the poisoned sample. The visual differences between the clean and poisoned images are visually negligible, confirming that the trigger introduces minimal perceptual distortion. Moreover, the Grad-CAM results reveal a distinct shift in the model's attention regions after trigger insertion , activating unrelated spatial regions and thereby successfully triggering the backdoor behavior.

Fig. 6 visualizes clean images and backdoored images generated by various attacks. For CerP, we use a trigger with the same shape as the initial DBA trigger for visualization purposes. The initial trigger for F3BA is similar to that of Neurotoxin, but after several rounds of optimization the patch evolves into random noise. In the case of IBA, the trigger is generated by training a separate UNet for each local client, which is constrained by the amount of local data available. As a result, the trigger

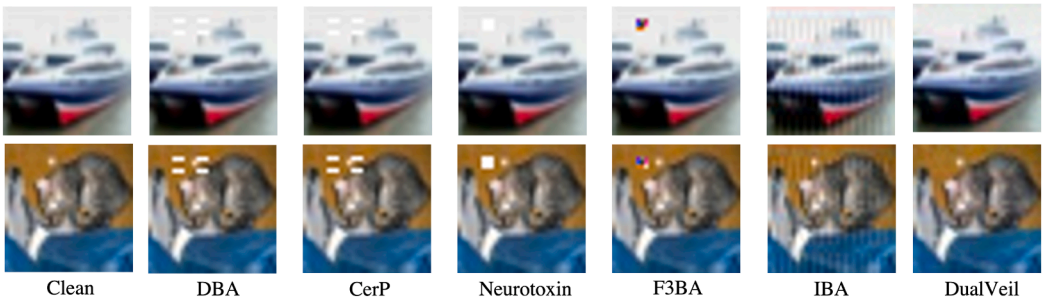


Fig. 6. Table visualizes clean samples and backdoored samples under various attack methods.

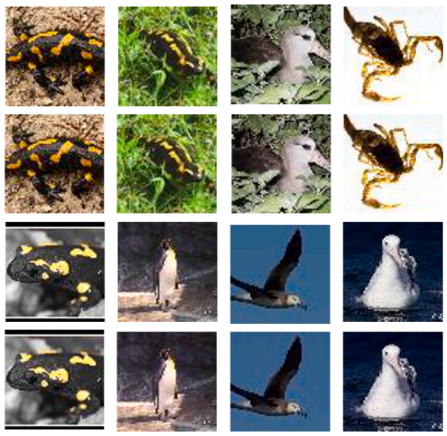


Fig. 7. Visualization of poisoned samples on Tiny-ImageNet.

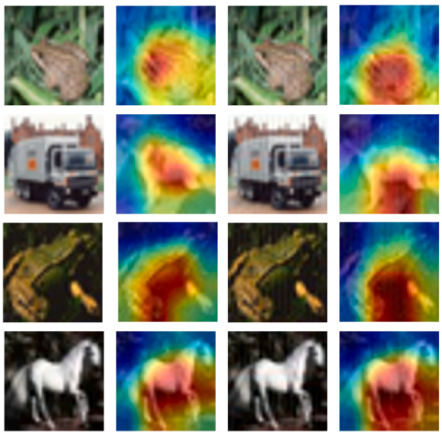


Fig. 8. Visualization results of DualVeil t rigger Δ at constrain $\epsilon = 4/255$.

Table 6
SSIM and PSNR comparison of poisoned samples generated by different backdoor attack methods.

Metrics	DBA	CerP	NeurotoXin	F3BA	IBA	DualVeil
SSIM	0.9219	0.9219	0.9455	0.9594	0.9432	0.9485
PSNR	26.6524	26.6500	28.8274	28.2587	31.2850	32.1910

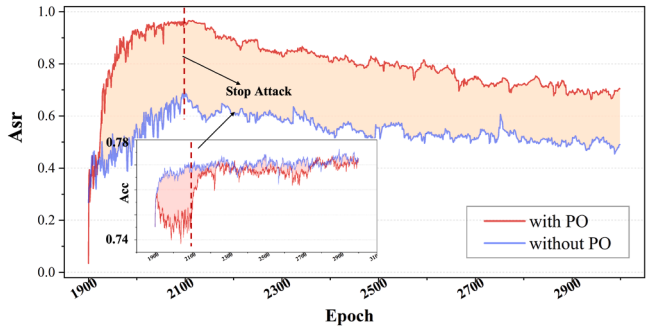


Fig. 9. The impact of with or without a PO on the persistence of backdoor tasks in the global model. Malicious clients start DualVeil attack at 1900 round, using $lr_p = .002$ for 5 epochs of training. Stop attacking after 200 rounds and use clean data for normal training to 3000 round.

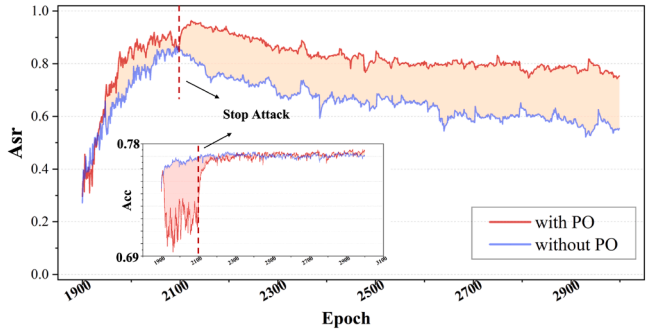


Fig. 10. The impact of with or without a PO on the persistence of backdoor tasks in the global model. Malicious clients start DualVeil attack at 1900 round, using $lr_p = .01$ for 2 epochs of training. Stop attacking after 200 rounds and use clean data for normal training to 3000 round.

for each image varies slightly. Our proposed attack introduces subtle, visually imperceptible perturbations while ensuring a high attack success rate. Table 6 presents quantitative evaluations of poisoned samples generated by different attack methods in terms of SSIM (Structural Similarity Index) and PSNR (Peak Signal-to-Noise Ratio). SSIM measures the perceptual similarity between the original and poisoned images, while PSNR quantifies overall image fidelity. Our method achieves higher image fidelity and lower perceptual distortion, demonstrating the effectiveness and stealthiness of the proposed trigger design.

Ablation Experiments. To evaluate the contribution of the ronePersistence Optimization (PO) to extending the persistence of the backdoor, we evaluate the backdoor task persistence under two hyperparameter settings. Fig. 9 corresponds to malicious clients training for 5 local epochs with a learning rate of $lr_p = .002$, while Fig. 10 corresponds to training for 2 local epochs with $lr_p = .01$.

From Fig. 9, at the end of the attack phase (round 2100), the Asr increases from 68.18% to 95.76%, an improvement of 27.58%, whereas the main task accuacy (Acc) decreases only slightly from 76.82% to 75.50% (a decrease of 1.32%). After continued benign training until round 3000, the main task accuracy remains stable (about 77.26%) with or without ronePO, while the Asr rised from 49.04% to 70.41% (an improvement of 21.37%)

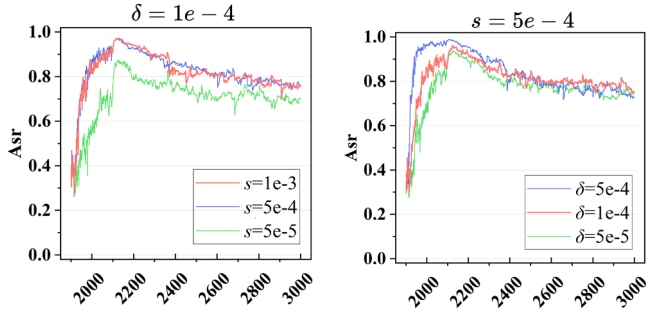


Fig. 11. Impact of hyper-parameters δ and s in DualVeil.

In Fig. 10, at round 2100 the Asr rises from 85.35% to 89.17%, an increase of 3.82%, while the accuracy decreases by 4.35%. By round 3000, the Asr decreases from 70.41% to 55.36%, a drop of 15.05%, whereas the attack accuracy changes only slightly from 77.02% to 77.44%, showing minimal difference. The orange shaded area in Fig. 10 highlights that, after stopping the attack at the round 2100, the performance gap between runs with and without parameter injection gradually became evident. This observation further underscores the contribution of the ronePO in enhancing attack accuracy and prolonging the backdoor persistence.

Through the above comparison, we find that at the round 2100, the influence of the ronePO on Asr is more pronounced in this setting. This can be attributed to the fact that malicious clients perform more local training, which increases the disparity between the global and local models, thereby reducing the effectiveness of backdoor attacks that rely solely on optimized triggers.

Moreover, our empirical results indicate that the persistence of the backdoor task continues to extend even after the Parameter Optimization phase has ceased. As training progresses, the impact of parameter optimization on the main task's performance becomes nearly negligible.

Hyper-Parameters. In the section, we evaluate the hyper-parameters of the ronePO. Fig. 11 show DualVeil performance under variations of the hyper-parameter while holding either s or δ fixed. We observe that small changes in δ have a limited effect on extending longevity of the backdoor: as the number of federated communication rounds increases, updates constrained to the δ-sphere do not substantially prolong backdoor lifetime. However, δ has a more pronounced effect on Asr. For example, in Fig. 11a during the update between the 1900 and the 2100 round, larger values of δ increase DualVeil's Asr, consistent with the intuition that parameter updates on the δ-sphere can move model parameters in direction that reinforce the backdoor. Note that excessively large δ may also increase the detectability of the backdoored model by defense mechanisms.

In contrast, Fig. 11b highlights the important of the parameter selection threshold s for extending backdoor persistence. Larger threshold values tend to further enhance persistence. Our Empirical results indicate only a small difference between $s = 5e^{-4}$ and $s = 1e^{-4}$ was minimal, suggesting that choosing $s = 5e^{-4}$ is sufficient for identifying neuron parameters that are insensitive to the main task.

Trigger and Backdoor Activation. Because the trigger is obtained by optimizing a loss function, we are concerned that it might capture class-specific features rather than model-specific vulnerabilities. This would contradict the desired behavior of backdoor, which should be effective primarily on the backdoored model and not generalize across unrealated. To examine this, we evaluate the generalizability of the learned trigger under two federated learning settings with 20 and 1000 clients, respectively. In both setting, 20% of the clients are malicious. In these experiments the malicious clients participate in the local training but do not contribute malicious updates to the server aggregation (they receive the distributed global model but refrain from uploading poisoned updates), which isolates the trigger's effect from model poisoning.

Table 7
Acc and Asr in a 20-client federated learning setting on CIFAR-10, where malicious clients are excluded from model aggregation.

Clients	Models	0 round		100 round	
		Acc	Asr	Acc	Asr
20	ResNet18	10.00%	00.00%	75.65%	02.31%
	VGG11	10.00%	00.63%	81.43%	02.01%

Table 8
Acc and Asr in a 1000-client federated learning setting on CIFAR-10, where malicious clients are excluded from model aggregation.

Clients	Models	1900 round		2000 round	
		Acc	Asr	Acc	Asr
1000	ResNet18	74.09%	03.33%	75.01%	02.52%
	VGG11	71.09%	00.38%	74.05%	02.88%

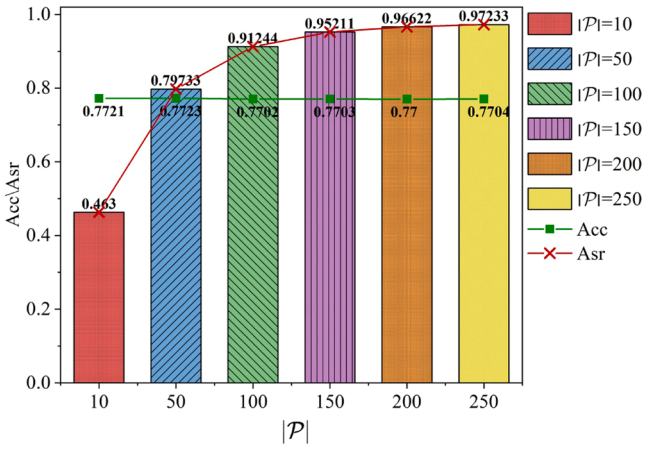


Fig. 12. Impact of the malicious clients number in DualVeil.

This setup allows us to distinguish whether the trigger exploits model specififc vulnerabilities or merely encodes class-specific patterns. As shown in Tables 7 and 8, across 100 communication rounds the attack success rate remains low when malicious updates are not aggregated, indicating that the learned trigger does not generalize across models and is not simply a class based exploit. Morevoer, As shown in Fig. 10 , when only 10 malicious clients are present among 1000 participants, the backdoor attack still attains an ASR exceeding 46%, demonstrating that DualVeil can successfully inject backdoors even when the fraction of malicious clients is very small.

The Impact of the Number of Malicious Clients. In Fig. 12, we investigate the effect of varying the number of malicious clients on the attack performance. Specifically, we limit the number of participating malicious clients to $|\mathcal{P}| = \{10, 50, 100, 150, 200, 250\}$ out of a total of 1000 clients. We record the backdoor Asr at round 2500. The results show that even with only 10 malicious clients, the backdoor task can still be successfully injected, achieving an Asr of 46.3%. This finding demonstrates that DualVeil remains effective in injecting persistent backdoors into the global model even when the proportion of malicious clients is extremely small.

The Implact of the Non-I.I.D. We further analyze the influence of non-I.I.D. data distributions on the performance of DualVeil, as illustrated in Fig. 13. Using Dirichlet sampling to simulate data heterogeneity among 1000 clients, we observe that when Dirichlet concentration parameter α is sufficiently small, some randomly selected clients may receive no data samples. Although this reduces the number of clients effectively

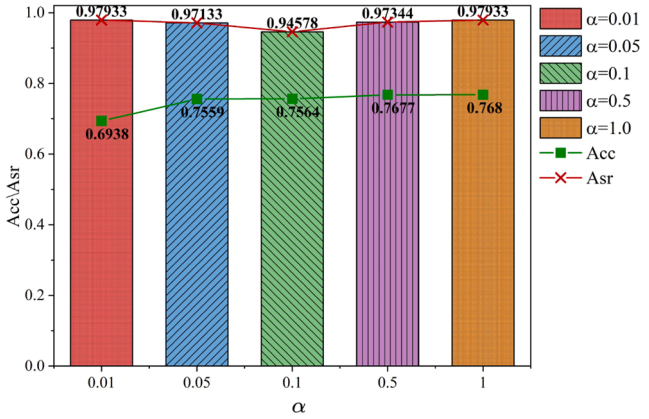


Fig. 13. Impact of dirichlet parameters α in DualVeil.

participating in training, it provides a realistic representation of highly non-I.I.D. federated settings.

Computational and Communication Cost Analysis DualVeil introduces additional local computation compared with methods that use handcrafted triggers (e.g., DBA, Cerp, NeurotoXin), since it requires per-client trigger optimization. This extra cost is borne only by malicious clients and therefore does not increase the computational burden on benign participants or the central server. In terms of resource footprint, DualVeil is comparable to other optimization-based backdoor attacks such as F3BA and IBA. However, unlike IBA, which relies on a trainable generator (for example, a U-Net) to synthesize triggers, DualVeil optimizes a parameterized trigger in the form of a single image. As a result, DualVeil avoids the memory and computation overhead associated with training an auxiliary generator, reducing both space and compute requirements. Concretely, the computational cost of DualVeil scales with the size of the parameterized trigger (for example, CIFAR-10 triggers have size $3 \times 32 \times 32$), and in our implementation the per-step gradient updates operate directly on these image parameters. Overall, while DualVeil requires more local computation than handcrafted-trigger methods, its overhead is modest and comparable to other optimization-based attacks, and it benefits from a simpler optimization target compared with generator-based approaches.

For communication cost, DualVeil does not change the transmission of model parameters between clients and the server compared to FedAvg; model uploads and downloads follow the same protocol and incur the same bandwidth cost. The additional communication overhead introduced by DualVeil stems solely from the exchange related to the collusive trigger selection: malicious clients publish their locally optimized triggers (each trigger is a single image) and evaluate triggers received from other malicious clients, then vote to select the best trigger. Since a trigger is the size of a single image, this overhead is negligible compared to model updates (which are typically on the order of megabytes). Therefore, the incremental communication cost of DualVeil is minimal and does not materially affect the overall communication budget of the federated training process.

5. Conclusion

In this paper, we present DualVeil, a backdoor attack framework for federated learning that achieves both persistence and invisibility through dual optimization. DualVeil decouples the correlation between the main task and backdoor parameters while minimizing the visual detectability of the trigger, thereby producing stealthy yet highly effective backdoors. Extensive experiments demonstrate the limitations of existing defense mechanisms in federated learning, particularly under scenarios involving a large number of benign clients and only a few malicious ones. For future work, unlike this study, which focuses on the impacts of parameter and trigger injection attacks, researchers are encouraged to investigate advanced defense strategies, roneincluding pre-aggregation filtering and post-aggregation mitigation. In addition, potential countermeasures specifically designed to detect or disrupt the dual optimization and client-collusion mechanisms in DualVeil could provide valuable directions for strengthening federated learning security. Furthermore, future efforts could improve the scalability and efficiency of defense mechanisms in large-scale federated systems.

CRediT authorship contribution statement

Maozhen Zhang: Writing – original draft, Visualization, Resources, Project administration, Methodology; **Mengnan Zhao:** Writing – review & editing, Validation, Supervision; **Wei Wang:** Formal analysis, Conceptualization; **Bo Wang:** Methodology, Investigation, Funding acquisition.

Data availability

Data will be made available on request.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work is supported by the Application Fundamental Research Project of Liaoning Province (No. 2025JH2/101330123), the Open Research Fund of the National Key Laboratory of Cyber Space Behavior Cognition Technology (No. CBCT2024KF02) and the National Natural Science Foundation of China (No. 62372452).

References

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, B.A. y Arcas, Communication-efficient learning of deep networks from decentralized data, in: Artificial Intelligence and Statistics, 2017, pp. 1273–1282.
[2] W. Ma, X. Wu, S. Zhao, T. Zhou, D. Guo, L. Gu, Z. Cai, M. Wang, FedSH: towards privacy-preserving text-based person re-identification, IEEE Trans. Multimed. 26 (2023) 5065–5077.
[3] Q. Xu, R. Zhang, Y. Zhang, Y.-Y. Wu, Y. Wang, Federated adversarial domain hallucination for privacy-preserving domain generalization, IEEE Trans. Multimed. 26 (2023) 1–14.
[4] R. Zhang, J. Mao, H. Wang, B. Li, X. Cheng, L. Yang, A survey on federated learning in intelligent transportation systems, IEEE Trans. Intell. Veh. 10 (2024) 3043–3059.
[5] S. Rani, A. Kataria, S. Kumar, P. Tiwari, Federated learning for secure IoMT applications in smart healthcare systems: a comprehensive review, Knowl. Based Syst. 274 (2023) 110658.
[6] P. Chatterjee, D. Das, D.B. Rawat, Federated learning empowered recommendation model for financial consumer services, IEEE Trans. Consum. Electron. 70 (2023). 2508–2516.
[7] D. Zhan, K. Xu, X. Liu, T. Han, Z. Pan, S. Guo, Practical clean-label backdoor attack against static malware detection, Comput. Secur. 150 (2025) 104280.
[8] K. Grosse, T. Lee, B. Biggio, Y. Park, M. Backes, I. Molloy, Backdoor smoothing: demystifying backdoor attacks on deep neural networks, Comput. Secur. 120 (2022) 102814.
[9] Q. Ren, Y. Zheng, C. Yang, Y. Li, J. Ma, Shadow backdoor attack: multi-intensity backdoor attack against federated learning, Comput. Secur. 139 (2024) 103740.
[10] S. Lu, R. Li, W. Liu, X. Chen, Defense against backdoor attack in federated learning, Comput. Secur. 121 (2022) 102819.
[11] Q. Li, W. Chen, X. Xu, Y. Zhang, L. Wu, Precision strike: precise backdoor attack with dynamic trigger, Comput. Secur. 148 (2025) 104101.
[12] A. Sharma, N. Marchang, A review on client-server attacks and defenses in federated learning, Comput. Secur. (2024) 103801.
[13] T. Gu, K. Liu, B. Dolan-Gavitt, S. Garg, BadNets: evaluating backdooring attacks on deep neural networks, IEEE Access 7 (2019) 47230–47244.
[14] G. Fields, M. Samragh, M. Javaheripi, F. Koushanfar, T. Javidi, Trojan signatures in DNN weights, in: International Conference on Computer Vision, 2021, pp. 12–20.
[15] B. Wang, F. Yu, F. Wei, Y. Li, W. Wang, Invisible intruders: label-consistent backdoor attack using re-parameterized noise trigger, IEEE Trans. Multimed. 26 (2024) 10766–10778.
[16] K. Gao, J. Bai, B. Wu, M. Ya, S.-T. Xia, Imperceptible and robust backdoor attack in 3D point cloud, IEEE Trans. Inf. Forensics Secur. 19 (2023) 1267–1282.

[17] G. Wang, H. Ma, Y. Gao, A. Abuadbba, Z. Zhang, W. Kang, S.F. Al-Sarawi, G. Zhang, D. Abbott, One-to-multiple clean-label image camouflage (OmClic) based backdoor attack on deep learning, Knowl. Based Syst. 288 (2024) 111456.
[18] N. Rodríguez-Barroso, E. Martínez-Cámara, M.V. Luzón, F. Herrera, Backdoor attacks-resilient aggregation based on robust filtering of outliers in federated learning for image classification, Knowl. Based Syst. 245 (2022) 108588.
[19] U. Zukaib, X. Cui, Mitigating backdoor attacks in federated learning based intrusion detection systems through neuron synaptic weight adjustment, Knowl. Based Syst. 314 (2025) 113167.
[20] Z. Zhang, A. Panda, L. Song, Y. Yang, M. Mahoney, P. Mittal, R. Kannan, J. Gonzalez, Neurotoxin: durable backdoors in federated learning, in: International Conference on Machine Learning, 2022, pp. 26429–26446.
[21] P. Fang, J. Chen, On the vulnerability of backdoor defenses for federated learning, in: AAAI Conference on Artificial Intelligence, 37, 2023, pp. 11800–11808.
[22] T.D. Nguyen, T.A. Nguyen, A. Tran, K.D. Doan, K.-S. Wong, IBA: towards irreversible backdoor attacks in federated learning, Adv. Neural Inf. Process. Syst. 36 (2024) 66364–66376.
[23] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, Y. Gao, A survey on federated learning, Knowl. Based Syst. 216 (2021) 106775.
[24] P. Rieger, T.D. Nguyen, M. Miettinen, A.-R. Sadeghi, Deepsight: mitigating backdoor attacks in federated learning through deep model inspection, arXiv preprint arXiv:2201.00763 (2022).
[25] C. Fung, C.J.M. Yoon, I. Beschastnikh, The limitations of federated learning in sybil settings, in: International Symposium on Research in Attacks, Intrusions and Defenses, 2020, pp. 301–316.
[26] M.S. Ozdayi, M. Kantarcioglu, Y.R. Gel, Defending against backdoors in federated learning with robust learning rate, in: AAAI Conference on Artificial Intelligence, 35, 2021, pp. 9268–9276.
[27] P. Blanchard, E.M. El Mhamdi, R. Guerraoui, J. Stainer, Machine learning with adversaries: byzantine tolerant gradient descent, in: I. Guyon, U.V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems, 30, Curran Associates, Inc., 2017.
[28] Y. Li, Y. Jiang, Z. Li, S.-T. Xia, Backdoor learning: a survey, IEEE Trans. Neural Netw. Learn. Syst. 35 (1) (2022) 5–22.
[29] A. Khaddaj, G. Leclerc, A. Makelov, K. Georgiev, H. Salman, A. Ilyas, A. Madry, Rethinking backdoor attacks, in: International Conference on Machine Learning, PMLR, 2023, pp. 16216–16236.
[30] M. Saremi, M. Khalooei, R. Rastgoo, M. Sabokrou, Projan: a probabilistic trojan attack on deep neural networks, Knowl. Based Syst. 304 (2024) 112565.
[31] X. Chen, C. Liu, B. Li, K. Lu, D. Song, Targeted backdoor attacks on deep learning systems using data poisoning, arXiv preprint arXiv:1712.05526 (2017).
[32] T. Wang, Y. Yao, F. Xu, S. An, H. Tong, T. Wang, An invisible black-box backdoor attack through frequency domain, in: European Conference on Computer Vision, Springer, 2022, pp. 396–413.
[33] Y. Li, Y. Li, B. Wu, L. Li, R. He, S. Lyu, Invisible backdoor attack with sample-specific triggers, in: International Conference on Computer Vision, 2021, pp. 16463–16472.
[34] W. Jiang, H. Li, G. Xu, T. Zhang, Color backdoor: a robust poisoning attack in color space, in: Conference on Computer Vision and Pattern Recognition, 2023, pp. 8133–8142.
[35] Y. Zeng, W. Park, Z.M. Mao, R. Jia, Rethinking the backdoor attacks' triggers: a frequency perspective, in: International Conference on Computer Vision, 2021, pp. 16473–16481.
[36] T. Huynh, D. Nguyen, T. Pham, A. Tran, COMBAT: alternated training for effective clean-label backdoor attacks, in: AAAI Conference on Artificial Intelligence, 38, 2024, pp. 2436–2444.
[37] H. Zhang, J. Jia, J. Chen, L. Lin, D. Wu, A3FL: adversarially adaptive backdoor attacks to federated learning, in: A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, S. Levine (Eds.), Advances in Neural Information Processing Systems, 36, Curran Associates, Inc., 2023, pp. 61213–61233.
[38] M. Ilić, M. Ivanović, V. Kurbalija, A. Valachis, Towards optimal learning: investigating the impact of different model updating strategies in federated learning, Expert Syst. Appl. 249 (2024) 123553.
[39] D.C. Nguyen, M. Ding, P.N. Pathirana, A. Seneviratne, J. Li, H.V. Poor, Federated learning for internet of things: a comprehensive survey, IEEE Commun. Surv. Tutorials 23 (3) (2021) 1622–1658.
[40] D. Zeng, S. Liang, X. Hu, H. Wang, Z. Xu, Fedlab: a flexible federated learning framework, J. Mach. Learn. Res. 24 (100) (2023) 1–7.
[41] M. Alam, E. Sarkar, M. Maniatakos, PerDoor: persistent backdoors in federated learning using adversarial perturbations, 2023, pp. 1–6.
[42] A. Krizhevsky, G. Hinton, et al., Learning multiple layers of features from tiny images (2009).
[43] Y. Le, X. Yang, Tiny imagenet visual recognition challenge, CS 231N 7 (7) (2015) 3.
[44] C. Xie, K. Huang, P.-Y. Chen, B. Li, DBA: distributed backdoor attacks against federated learning, in: International Conference on Learning Representations, 2019.
[45] X. Lyu, Y. Han, W. Wang, J. Liu, B. Wang, J. Liu, X. Zhang, Poisoning with cerberus: stealthy and colluded backdoor attack against federated learning, in: AAAI Conference on Artificial Intelligence, 37, 2023, pp. 9020–9028.