BADAVG: AGGREGATION-AWARE BACKDOOR ATTACK AGAINST FEDERATED CONTRASTIVE LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Federated Learning (FL) enables collaborative model training without explicit data sharing, ensuring the participants' privacy. Unfortunately, the distributed nature of FL exposes trained models to backdoor attacks. Federated Contrastive Learning (FCL), which learns representations from unlabeled data, is often assumed to be more robust than its supervised counterpart, yet recent attacks on centralized Contrastive Learning have raised concerns about FCL's security. In this paper, we propose BadAvg, a novel backdoor attack on FCL that crafts malicious updates by modeling their propagation through federated aggregation. A malicious client running BadAvg replicates the server's aggregation of benign models via simulation, and crafts the backdoor by optimizing an adversarial loss on this differentiable simulation. We evaluate BadAvg on three downstream tasks under IID and Dirichlet-partitioned data, where a single malicious client contributes to only 0.25% of the local updates, measuring both peak effectiveness and attack persistence. BadAvg outperforms the competitors without defenses, and remains the strongest attack in the majority of the defended settings.

1 INTRODUCTION

Federated Learning (FL) trains a shared model across clients that never disclose their raw data. In the mainstream FL algorithm, FedAvg (McMahan et al., 2017), the server builds a new global model by averaging the updates that clients return. Neither the data a client holds nor the training it runs on that data is ever visible to the server. This invisibility makes FL private, but is also a source of vulnerabilities: a malicious client can freely tamper with either one, and the server has no direct way to detect tampering (Tolpegin et al., 2020). In particular, *backdoor attacks* teach the model a hidden behavior that fires only on inputs carrying a specific trigger, acting normally on everything else (Gu et al., 2019). In FL, model substitution combined with *constrain-and-scale* makes such backdoors stealthy, effective, and resilient (Bagdasaryan et al., 2020).

Federated Contrastive Learning (FCL) extends FL to train task-agnostic encoders from *unlabeled* client data (Liu et al., 2023). Since FCL exposes no labels to poison, it is generally assumed to resist backdoors better than its supervised counterpart. BadEncoder (Jia et al., 2022) has shown that assumption to be unfounded in the centralized case, which raises the question of whether FCL is any safer. Transferring such attacks to FCL is not straightforward: since a subset of clients participates in each round, a malicious client contributes in only a minority of updates, and whatever backdoor it plants must then survive repeated clean training by other benign clients. The two existing studies in this direction, BAGEL (Huang et al., 2023) and BADFSS (Zhang et al., 2024), extend BadEncoder to the unsupervised federated setting. Still, neither implements a mechanism to ensure persistence: their updates are either re-injected at every round or amplified by a large scaling factor, resulting in poor resistance to the benign aggregations that follow. Moreover, neither has been evaluated under a realistic scenario where the attacker controls a single client and must implant resistant backdoors in the presence of defenses, nor reports attack success after the attacker stops participating. Our evaluation closes this gap by continuing measurement for 100 rounds past the attacker's last round.

In this paper, we propose *BadAvg*, a novel FCL poisoning attack that integrates federated aggregation within the malicious client, achieving high stealthiness and persistence. The malicious client wraps the server's aggregation rule in a differentiable layer, called the BadAvg layer, estimates the benign contribution that will be averaged alongside its backdoor attack, and optimizes its local encoder

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

through the BadAvg layer, so that the adversarial objective is evaluated on an estimate of the model the server will deploy rather than on the local one. The layer admits different strategies for simulating the benign contribution and different constraints on update magnitude, both of which we vary in our experiments, as well as different adversarial losses, for which we use BadEncoder throughout. Differentiability further allows BadAvg to require that its update match the norm of a benign one *during* optimization rather than by post-hoc projection, which no prior FCL attack can express, letting it directly find the best update direction within the constrained budget. We evaluate BadAvg under conditions considerably more realistic and challenging than prior FCL attacks assume: our malicious client takes part in only 10 aggregations, contributing to just 0.25% of the local updates, and we measure attack success up to 100 rounds after its final participation, so that a backdoor counts as successful only if it survives extended clean training.

Overall, this work makes the following contributions: *i)* we design *BadAvg*, a novel attack which integrates federated aggregation inside the attacker's optimization through a differentiable FedAvg layer, allowing the malicious update to be held to a benign norm during optimization rather than by post-hoc projection of prior FCL attacks; *ii)* we demonstrate BadAvg's effectiveness both without defenses, and against two server-side defenses, under which it remains the strongest attack on peak success in most settings. While both defenses reduce the attacks we test, they do so to different extents: Clip+Noise consistently suppresses backdoor persistence, whereas FedTruth leaves surviving backdoors almost as durable as in the undefended setting. This strong drop in persistence under Clip+Noise is a promising result from the defenders' perspective, although further work is needed to reduce the peak ASR it still allows.

2 Related Work

Backdoor attacks on Federated Learning. Backdoor attacks (Gu et al., 2019), which train a model to predict a target class while preserving accuracy on clean inputs, are particularly dangerous because they are difficult to detect, remain silent until activated, and can survive the fine-tuning a user performs to adapt a downloaded model to a new task. FL is especially exposed, as malicious clients can enter the system and inject a backdoor into the global model, provided the attack survives aggregation. In particular, adversarial clients can exploit the linearity of FedAvg aggregation to perform *model replacement* attacks, crafting an update that, once aggregated, replaces the global model with a desired backdoored version, without degrading main-task performance (Bagdasaryan et al., 2020). All these attacks, however, are inherently supervised and cannot be directly applied to contrastive learning, where only the encoder trained in an unsupervised manner can be targeted.

Backdoor Attacks on Contrastive Learning. BadEncoder (Jia et al., 2022) is a backdoor attack targeting CL in centralized settings, which has shown that unsupervised CL does not guarantee the resilience of the model to attacks, even after fine tuning on the downstream task. The attacker optimizes a contrastive loss combining three terms:

$$\mathcal{L}_{adv} = \mathcal{L}_1 + \lambda_1 \cdot \mathcal{L}_2 + \lambda_2 \cdot \mathcal{L}_3 \tag{1}$$

where $\mathcal{L}_1$ maximizes the similarity between backdoored and reference images, $\mathcal{L}_2$ that between augmented references and clean reference features, and $\mathcal{L}_3$ maintains the similarity between the clean features of the local and global models, while λ_1, λ_2 weigh their relative contributions. Notably, $\mathcal{L}_3$ anchors the poisoned encoder to the model the attacker currently holds, which in a centralized setting is also the model that will be deployed. This approach demonstrates that backdoors embedded during unsupervised pre-training can persist across fine-tuning and affect multiple downstream tasks without ever manipulating labeled data. In federated learning, however, the attacker does not optimize the global encoder, but only a local one that eventually undergoes aggregation with multiple clean clients. Extensions of BadEncoder to FCL must therefore update the loss to operate on an estimate of the post-aggregation model rather than on the local one.

Backdoor Attacks on Federated Contrastive Learning. To the best of our knowledge, there are only two backdoor attacks on FCL algorithms: BAGEL (Huang et al., 2023) and BADFSS (Zhang et al., 2024). BAGEL directly applies BadEncoder in federated settings, where the malicious client optimizes its local adversarial model, scales the weights by a fixed factor, and sends the update to the server. BAGEL's scaling, however, produces anomalously large updates that risk detection.

BADFSS instead uses a feature-alignment method to reduce misalignment between the attacker's model and the global encoder distributed at clients.

Neither BAGEL nor BADFSS, however, implements a mechanism to ensure persistence: their updates are either re-injected at every round or amplified by a large scaling factor. BAGEL is evaluated either in a multi-shot scenario, with attacks at every round, or in a single-shot one in which three clients attack jointly every 100 rounds, introducing large update magnitudes that are easy to detect by trivial defenses. BADFSS's alignment mechanism can improve stealthiness, but prior work suggests that poisoning attacks in FL can lead to catastrophic forgetting if the impact on the global encoder is not accounted for Bagdasaryan et al. (2020). BADFSS has only been evaluated in an unrealistic scenario, where the attacker participates in every aggregation round.

Defenses against Federated Backdoors. Server-side defenses against federated backdoors fall into three broad families. *Norm-bounding and differentially private* defenses clip each update to a fixed or data-dependent norm and add Gaussian noise, directly targeting the magnitude on which model replacement relies (Sun et al., 2019). *Trust-scoring* defenses reweight clients by agreement with a reference direction, either a server-held root dataset (FLTrust (Cao et al., 2020)) or an iteratively estimated consensus (FedTruth (Ebron et al., 2023)). *Similarity-clustering* defenses, such as FLAME (Nguyen et al., 2022) and FoolsGold (Fung et al., 2020), identify malicious clients by the mutual resemblance of their updates. We evaluate one defense from each of the first two families. We do not include FLAME (Nguyen et al., 2022) as we observed its similarity-based clustering discarded benign clients under substantial statistical heterogeneity (e.g., Dirichlet partitioning).

3 Problem Formulation

3.1 FedSimCLR

We consider a federated contrastive learning setting in which a global encoder G^t is trained over communication rounds $t = 0, 1, \ldots, T$ using data distributed across a population of clients $\mathcal{N}$. At each round t, a subset of clients $\mathcal{N}^t \subset \mathcal{N}$ of fixed size $K = |\mathcal{N}^t|$ is sampled by the server. Each selected client n_i locally updates a copy of the global encoder, optimizing a contrastive learning objective (as in SimCLR (Chen et al., 2020)), producing a local update $l_i^t = L_i^t - G^t$, where L_i^t is the updated encoder. The server aggregates the received local updates using the FedAvg aggregation:

$$G^{t+1} = G^t + \frac{\eta}{K} \sum_{i \in \mathcal{N}^t} l_i^t \tag{2}$$

where η represents the learning rate of the global encoder's aggregation. At the last round, G^T can be used for multiple downstream classification tasks from a population $\mathcal{D}$: for each task D, a task-specific classifier head h_D is trained on top of the frozen encoder, forming a classifier $G^T \otimes h_D$.

3.2 Threat Model

We consider an adversarial client that aims to inject a backdoor into the global encoder during federated contrastive training. In accordance with past literature on adversarial attacks (Biggio et al., 2013; Demontis et al., 2019), we describe here the threat they pose in terms of their objective, knowledge, and capabilities.

Attacker's Goal. The attacker aims to manipulate the local training of a selected client, namely a *malicious client*, to obtain a backdoored global encoder $\hat{G}^{t+1}$ pursuing three goals. First, given a target downstream task $\bar{D}$, target class $\bar{y}_j$ and trigger ϵ, $\hat{G}$ should cause any input x containing ϵ to be classified as $\bar{y}_j$, when combined with the task-specific head $h_{\bar{D}}$, i.e., $\hat{G} \otimes h_{\bar{D}}(x \oplus \epsilon) = \bar{y}_j$. Second, the predictions on clean inputs should remain unchanged across all downstream tasks, meaning that for any task D and clean input x, $\hat{G} \otimes h_D(x)$ matches the output of the benign model $G \otimes h_D(x)$. Third, and central to this work, the attacker seeks *persistence*: the backdoor must remain effective after the attacker has stopped participating. Letting t_{last} denote the last attack round, the first condition must continue to hold for $\hat{G}^{t_{\text{last}}+k}$ for large k, where the intervening rounds contain only benign updates.

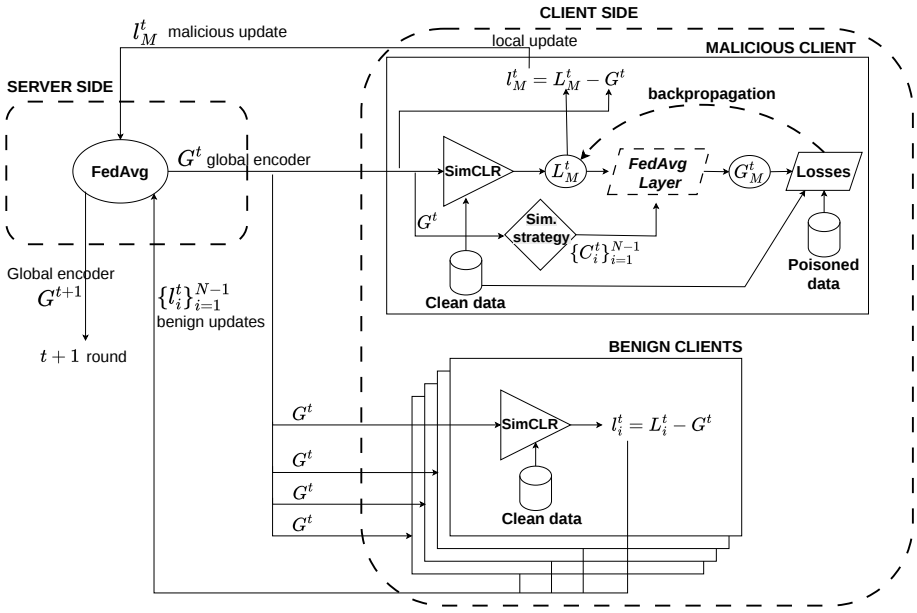


Figure 1: Overall scheme of a BadAvg attack round. Benign clients fine-tune G^t by SimCLR and produce their updates l_i^t. The attacker instead simulates the benign contribution via the FedAvg layer, obtaining an approximation G_M^t of $\hat{G}^{t+1}$, computes the BadEncoder losses on it, and backpropagates them through the layer to update L_M^t.

Attacker's Knowledge. As in the literature (Bagdasaryan et al., 2020; Jia et al., 2022), we assume that the attacker knows: 1) the number K of clients sampled in each round; 2) the learning rate η of FedAvg; 3) the target downstream task $\bar{D}$ and target class $\bar{y}_j$; 4) a dataset similar to the one available on the other participants. The attacker does not observe the private data of benign clients and does not control the task-specific classification head h_D. Since K and η may be public or inferred (Bagdasaryan et al., 2020) but are not necessarily known exactly, we study the impact of assuming a wrong value in an ablation study. Although pre-training is unsupervised, the encoder is deployed through heads on labeled data, so it is reasonable to assume the attacker knows its target and can reasonably hold images of it. Still, to preserve realism, we impose that the attacker's data is disjoint from the downstream data.

Attacker's Capabilities. The attacker controls exactly one malicious client, with full control over its local training process, and participates in federated rounds as if it were benign. The malicious client can use labeled data to compute the optimal attack, but can interfere with the global model G only by proposing updated encoder weights l_i. Moreover, since the server samples a subset $\mathcal{N}^t$ ($K = |\mathcal{N}^t|$) of the entire population $\mathcal{N}$ at each round t, the attacker participates only sporadically, may enter training after multiple rounds have elapsed, and will generally leave before it ends, so that many clean rounds occur between any two attack rounds and, crucially, after the last.

4 BADAVG

In this Section, we describe *BadAvg*, a novel FCL backdoor attack against unsupervised federated learning. BadAvg operates in a malicious client M where the attacker has full control of the learning process. The malicious client takes as input the global encoder G^t and returns the malicious update l_M^t, which the server aggregates with the other clients' benign updates to form the new backdoored global model $\hat{G}^{t+1}$. The core idea behind BadAvg is to make the federated aggregation step itself part of the attacker's optimization, so that the update is crafted to be effective *after* averaging with benign clients rather than before it.

Specifically, as illustrated in Figure 1, BadAvg introduces a differentiable *FedAvg layer* that estimates how changes to the local malicious encoder L_M^t affect the global model estimation G_M^t, so that the BadEncoder loss in equation 1 can be computed on G_M^t and backpropagated through the aggregation to optimize L_M^t with respect to the expected post-aggregation global model. Because the layer is agnostic to what is plugged into it, BadAvg is highly customizable over three main axes. The first is the *simulation strategy* $\mathcal{S}$, which estimates the aggregate contribution of the benign clients participating in the same round, a quantity the attacker cannot observe. The second is the *rescaling strategy*, which controls the magnitude of the resulting update and determines whether it passes norm-based filtering. The adversarial loss is a third, interchangeable component. Throughout this work, we instantiate it with the BadEncoder loss in equation 1. We distinguish two types of FL rounds: *clean rounds*, in which only benign clients are selected, and *attack rounds*, in which the local update of the malicious client M enters FedAvg.

4.1 Client Side: Benign client update

As shown in Figure 1, in every round t, benign clients receive the global model G^t from the server and fine-tune it on their local unlabeled data through SimCLR (Chen et al., 2020), a mainstream CL solution. SimCLR adopts the NT-Xent loss, which pulls together the representations of augmented views of the same sample and pushes apart those of different samples. After obtaining the locally updated model L_i^t, the benign client n_i computes its benign update as $l_i^t = L_i^t - G^t$ and sends it to the server for aggregation.

4.2 Client Side: Malicious client update

During an *attack round*, the malicious client M participates in the global model update, and uses BadAvg to inject the backdoor into G^{t+1}. Upon receiving G^t, it first performs a traditional SimCLR fine-tuning on its own clean data, obtaining the clean model L_M^t that serves as the starting point for the aggregation-aware update. Starting from G^t, it then constructs $K-1$ simulated benign models $\{C_i^t\}_{i=1}^{K-1}$ (which are not observable) using a *simulation strategy* $\mathcal{S}$, to be aggregated with L_M^t. Finally, it iteratively aggregates the simulated updates through the FedAvg layer, computes the adversarial loss on the aggregated encoder, and backpropagates its gradient to update L_M^t, before sending $l_M^t = L_M^t - G^t$ to the server.

The Differentiable FedAvg Layer. The core of BadAvg is a differentiable layer that reproduces the server's aggregation process inside the attacker's computational graph. Given the malicious model L_M^t and the $K-1$ benign models $\{C_i^t\}_{i=1}^{K-1}$ produced by the simulation strategy $\mathcal{S}$, the layer computes the estimated post-aggregation encoder as

$$G_M^t = G^t + \frac{\hat{\eta}}{\hat{K}}\Big[(L_M^t - G^t) + \sum_{i=1}^{\hat{K}-1}(C_i^t - G^t)\Big], \tag{3}$$

mirroring FedAvg aggregation equation 2 with the unobservable benign updates replaced by their simulated counterparts. The anchor G^t and C_i^t are fixed throughout the optimization; only L_M^t carries gradient, so $\partial G_M^t / \partial L_M^t = \hat{\eta}/\hat{K}$ and the loss computed on G_M^t propagates to the local encoder scaled by the aggregation rule itself. Equation 3 isolates the only quantity the attacker cannot observe, namely $\sum_i (C_i^t - G^t)$, which is estimated by the simulation strategy $\mathcal{S}$. Note that the attacker uses its own estimates $\hat{\eta}$ and $\hat{K}$, while the server aggregates with the true η and K regardless. We study the impact of a misestimate in the Ablation Study. The resulting G^{t+1} inherits the behavior of all participating clients, so the backdoor is transmitted only to the extent that it survives averaging with the $K-1$ benign updates, and thereafter the clean rounds in which the attacker is absent.

Simulation Strategies. A simulation strategy $\mathcal{S}$ is an estimator of $\sum_i (C_i^t - G^t)$, which must rely only on quantities the attacker can observe following the threat model described in the Problem Formulation section. It does not require being differentiable, as the simulated models enter in equation 3 as constants. Unless stated otherwise, we use *multi-clean*, which trains $K-1$ additional clean encoders from G^t on the attacker's own data with independent seeds and uses their mean and disper-

sion as the estimate. In our experiments, we compute the estimate at the first attack and reuse it to save computational cost. We discuss other approaches and the impact of the choice in the Appendix.

Rescaling Strategies. Optimizing through equation 3 can produce updates whose norm exceeds that of benign clients, since a larger $|L_M^t - G^t|$ survives the $\hat{\eta}/\hat{K}$ attenuation more easily. Where the server bounds update norms, this is a liability, so BadAvg exposes update magnitude as a second axis. The three strategies we consider all refer to the norm of the attacker's own clean fine-tuning at the current round, $\beta^t = |L^{t,\text{clean}}M - G^t|$, which is recomputed every round at no additional cost, as the simulation stage already computes its clean model at every attack round. *None* leaves the update unconstrained. *Benign* adds a differentiable penalty to the adversarial loss,

$$\mathcal{L} = \mathcal{L}_{adv} + \lambda_3 \left(\frac{|L_M^t - G^t| - \beta^t}{\beta^t} \right)^2, \tag{4}$$

so that the constraint is satisfied during optimization and the attacker spends its budget on the directions that matter. The penalty is two-sided, targeting the benign norm rather than bounding it to discourage excessive shrinkage of norms. For our experiments, we set $\lambda_3 = 1$.

Benign-scale instead projects the finished update onto the norm ball post hoc, scaling it by $\beta^t/|L_M^t - G^t|$: it satisfies the constraint exactly but discards the direction selected by the optimizer under a different budget. We test BadAvg using *none* and *benign*. In contrast, the baselines can only use *benign-scale*, as neither optimizes a differentiable model of the aggregation and so neither can express the constraint during training: BAGEL rescales its backdoor delta to β^t after fine-tuning, and Naive replaces the $\hat{K}/\hat{\eta}$ replacement factor with $\beta^t/|L_M^t - G^t|$. Post-hoc projection (*benign-scale*) selects a direction under an unconstrained budget and then shrinks it. When the direction for larger updates does not match the optimal direction for smaller ones, the implementation becomes suboptimal. The differentiable formulation of BadAvg instead allows us to incorporate the rescaling operation into the optimization itself, maintaining an optimal direction given the limited budget.

5 Experiments

In the following, we evaluate the threat posed by BadAvg to FCL over three popular datasets, comparing it against state-of-the-art competitors and baselines. To facilitate reproducibility, we include the code in the Supplementary.

We consider a FCL framework with a population of 100 clients, $K = 10$ of which are selected at each round (participation rate 10%), that train a global encoder over multiple communication rounds. We first run a variable number of clean rounds, in which only benign clients are selected for local fine-tuning and aggregation, and evaluate attacks starting at different rounds (10, 50, 100, and 200). Results for 50 and 200 are in the main text, deferring 10 and 100 to the Appendix. Over the following attack window of 100 rounds, one malicious client participates in the aggregation every 10 rounds together with 9 benign clients, for a total of 10 attacker participations. To isolate persistence from the variance of random selection, we replace stochastic selection of the attacker with a deterministic schedule matching its expected participation rate (K / N = 10%, i.e., once every 10 rounds). We then run 100 additional clean rounds to evaluate the model's resilience after the attack ends.

Overall, the attacker contributes to only 10 of the 4000 federated updates occurring during the measured window (0.25%), reflecting a practical constraint under which the malicious behavior must persist through extended phases of clean training on both sides of the attack. This contrasts with BAGEL (Huang et al., 2023) and BADFSS (Zhang et al., 2024), which evaluate their attacks in a simplified setting where the attacker participates in every aggregation round.

Model Architecture and Training. We use FedSimCLR as the target FCL algorithm, with $\lambda_1 = \lambda_2 = 1$ in BadEncoder, following the original paper (Jia et al., 2022). Benign and malicious clients train locally for 5 and 10 epochs, respectively, with a FedAvg server learning rate of $\eta = 1$. We evaluate the resulting encoder on downstream datasets by training a three-layer MLP (500, 512, 256 hidden nodes) at the server for up to 1000 epochs, with batch size 64 and learning rate 0.0001. This setting is consistent with Huang et al. (2023), enabling a fair comparison with previous work. All three attacks receive the same budget: 10 local epochs, the same 20k-image shadow set, and the same knowledge of the target task and class. The attacker trains for more epochs than genuine

clients to better implant the backdoor. The threat model grants full control of local training, and an adversary optimizing a different objective has no reason to keep benign hyperparameters. Additional architecture, hyperparameter, and trigger details are reported in the Appendix.

Datasets and Partitioning. We consider three image classification datasets: CIFAR-10 (Krizhevsky et al., 2009), STL-10 (Coates et al., 2011), and GTSRB (Stallkamp et al., 2011). Clients pretrain on CIFAR-10, and we evaluate the aggregated model's attack effectiveness across all three datasets as downstream. We partition the unlabeled pretraining data across clients under two regimes: *IID*, where each client receives a balanced training set with the same class distribution, and *Non-IID*, where data is split via a Dirichlet distribution with parameter 0.5, inducing heterogeneous local representations and larger differences among client updates.

Evaluation Metrics. We evaluate the downstream classifier fine-tuned on top of the (possibly backdoored) encoder using three metrics. *Peak ASR (P-ASR)* is the highest attack success rate reached inside the attack window, i.e., the percentage of triggered images classified as the target class; *Final ASR (F-ASR)* is the ASR measured 100 rounds after the attacker stops participating, capturing the backdoor's persistence. *Accuracy Drop (ΔAcc)* is the highest reduction in clean accuracy observed during the attack, relative to the clean reference model before the attack starts. The attacker's goal is a P-ASR (and ideally F-ASR) close to 100%, while keeping ΔAcc minimal to avoid trivial detection by defenses monitoring the model's accuracy.

Baseline and Competitors. To the best of our knowledge, only BAGEL and BADFSS target FCL. As neither algorithm's code is publicly available, we re-implemented BAGEL following the description provided in the original paper, while BADFSS could not be reproduced due to missing fundamental implementation details that are not recoverable from the paper alone. We reproduce BAGEL's attack using both its original scaling factor (100) (Huang et al., 2023) and its norm-rescaled variant (*benign-scale*), in which the scaling factor is set to match the estimated norm of benign clients' updates to survive norm-based defenses (e.g., clipping).

We also compare BadAvg against *Naive*, an original baseline we design to further confirm our method's validity, combining BadEncoder (Jia et al., 2022) with the model-replacement attack (Bagdasaryan et al., 2020). Naive applies BadEncoder to G^t to obtain a backdoored encoder G^t_M, then computes the local update that, once aggregated, reproduces G^t_M as the post-aggregation global encoder. Unlike BadAvg, which makes only the minimum changes required to survive aggregation and can leverage a custom loss to evade defenses, Naive directly solves for the update forcing the desired global encoder, making it useful to isolate the improvement brought by BadAvg's aggregation-aware (*FedAvg layer*) design. Even for Naive, we adopt both the non-rescaled and the *benign-scale* variants. Note that Naive generalizes BadEncoder by adding the direct rescaling operation needed to work in a federated learning setting and, when needed, the scaling to bypass the defenses.

5.1 Results

The comprehensive results are summarized in Table 1, which reports the peak ASR (P-ASR), the ASR 100 rounds after the attacker stops participating (F-ASR), and the highest accuracy drop observed during the attack (ΔAcc). We compare each attack's better-performing variant.

Regarding accuracy, most attacks remain highly stealthy, with ΔAcc typically within a few percentage points across datasets and settings. A notable exception is BAGEL in its unconstrained configuration, where the lack of norm control leads to substantial accuracy degradation, exceeding 40 percentage points in several settings; this drop highlights the importance of the norm-rescaling procedure for preserving stealthiness. Without defenses, BadAvg reaches ≥ 98.8 P-ASR in all 12 settings, reaching the best P-ASR and best F-ASR in 10 of 12 settings. Against Naive, which shares BadAvg's loss and scale, unconstrained BadAvg wins P-ASR in 12/12 undefended settings. Under FedTruth, which does not bound norms, BAGEL's 100× scaling is most persistent (7/12). Under Clip+Noise, BadAvg instead achieves the highest final ASR in most unblocked settings.

The attack-start round affects persistence significantly more than peak effectiveness. On STL-10 and CIFAR-10 under Non-IID, BadAvg's F-ASR is near zero when the attack begins at round 50 but exceeds 95% at round 200, while on GTSRB it stays above 95% in both cases. This suggests that convergence plays a significant role. Early in training, the model is still moving fast enough that 100

Table 1: **Attack outcomes across datasets, distributions, and defenses.** P-ASR = peak ASR inside the attack window (%); F-ASR = ASR 100 rounds after the attacker stops (%); ΔAcc = highest clean-accuracy drop observed during the attack (pp). ✗ = blocked (peak below the 60% viability threshold). **Bold** marks the best value in each column within a dataset/distribution block: highest for P-ASR and F-ASR, *lowest* for ΔAcc (a stealthier attack perturbs clean accuracy less). Ties are bolded jointly.

		start 50									start 200								
		No Defense			Clip+Noise			FedTruth			No Defense			Clip+Noise			FedTruth		
Attack	Norm	P-ASR	F-ASR	ΔAcc	P-ASR	F-ASR	ΔAcc	P-ASR	F-ASR	ΔAcc	P-ASR	F-ASR	ΔAcc	P-ASR	F-ASR	ΔAcc	P-ASR	F-ASR	ΔAcc
GTSRB — IID																			
BadAvg	unconstr.	**99.9**	**95.0**	11.2	✗	✗	✗	97.6	23.3	**7.5**	99.9	**97.1**	11.6	✗	✗	✗	99.1	86.0	8.7
	rescaled	99.4	71.7	12.9	**98.8**	14.5	10.4	99.1	60.2	9.9	99.6	76.6	13.7	**99.3**	**41.9**	12.0	**99.4**	68.1	13.0
BAGEL	unconstr.	99.0	49.6	30.0	✗	✗	✗	**99.4**	**86.0**	10.5	**100.0**	96.8	47.1	✗	✗	✗	99.2	**88.8**	10.0
	rescaled	97.9	61.4	**9.5**	90.6	10.9	**8.8**	98.0	64.7	8.4	98.3	79.4	**8.5**	93.7	31.8	8.8	98.4	71.9	**7.8**
Naive	unconstr.	98.7	79.9	11.7	91.3	**14.9**	**8.8**	98.4	70.3	8.7	98.9	88.0	8.7	92.7	32.9	9.3	98.7	74.9	**7.8**
	rescaled	97.8	73.0	9.6	91.0	13.3	8.9	98.0	68.0	8.3	98.5	79.6	8.9	91.7	27.4	**8.6**	98.3	74.2	**7.8**
GTSRB — Non-IID (Dirichlet)																			
BadAvg	unconstr.	**99.8**	**97.0**	11.0	✗	✗	✗	95.3	15.3	**7.4**	99.8	**96.3**	**9.9**	✗	✗	✗	99.1	**89.4**	10.9
	rescaled	98.0	15.5	13.3	**98.6**	**7.2**	11.9	**98.6**	20.5	13.9	99.8	38.2	16.7	**97.7**	**14.0**	13.8	**99.5**	12.2	14.8
BAGEL	unconstr.	98.2	77.1	33.6	✗	✗	✗	98.0	**77.2**	15.1	**100.0**	94.9	51.5	✗	✗	✗	98.7	88.3	15.0
	rescaled	94.2	11.3	**10.9**	84.1	5.7	**9.6**	94.3	16.8	11.4	96.6	24.2	11.1	75.9	8.9	10.0	97.3	36.3	**9.0**
Naive	unconstr.	96.4	24.0	14.2	91.5	5.5	10.3	96.7	28.4	10.9	95.2	51.2	11.6	86.4	9.0	11.2	98.4	76.6	11.3
	rescaled	93.9	15.7	11.2	84.5	6.7	9.7	94.4	33.2	11.4	96.2	28.3	10.9	75.3	12.7	**9.6**	97.7	42.2	**9.0**
CIFAR-10 — IID																			
BadAvg	unconstr.	**99.0**	**96.7**	1.5	✗	✗	✗	**98.2**	**77.6**	**1.0**	**99.6**	**97.3**	3.5	✗	✗	✗	**95.4**	44.4	2.3
	rescaled	66.8	34.3	1.9	68.2	24.7	1.4	80.4	48.4	2.1	95.2	70.5	3.4	73.8	38.0	3.1	92.2	60.2	3.0
BAGEL	unconstr.	80.8	64.0	43.4	✗	✗	✗	87.1	73.5	2.9	98.0	53.5	44.2	✗	✗	✗	91.9	**65.1**	3.0
	rescaled	70.3	47.0	1.1	**87.0**	20.4	**0.5**	72.6	52.1	1.1	81.5	69.1	2.5	82.2	23.9	2.8	77.6	37.9	2.6
Naive	unconstr.	69.3	45.7	2.1	83.2	22.2	0.7	71.5	60.5	**1.0**	88.3	73.9	2.2	80.0	36.1	2.5	69.4	47.2	2.7
	rescaled	88.9	27.3	**0.4**	80.7	**25.0**	3.2	79.4	48.9	3.6	96.6	41.2	**0.8**	**91.9**	**41.4**	**0.1**	89.6	51.9	**1.1**
CIFAR-10 — Non-IID (Dirichlet)																			
BadAvg	unconstr.	**98.8**	**1.6**	**2.1**	✗	✗	✗	75.9	19.1	2.0	**99.8**	**95.1**	4.0	65.7	20.8	3.4	**99.2**	94.0	2.8
	rescaled	✗	✗	✗	✗	✗	✗	✗	✗	✗	84.4	35.2	3.8	77.7	**49.1**	2.9	98.4	69.9	2.7
BAGEL	unconstr.	76.0	0.4	44.4	✗	✗	✗	75.2	16.7	5.8	93.2	81.1	51.9	✗	✗	✗	96.8	**95.5**	3.4
	rescaled	✗	✗	✗	✗	✗	✗	✗	✗	✗	95.8	45.7	3.7	88.1	30.2	3.2	98.7	77.5	2.8
Naive	unconstr.	✗	✗	✗	✗	✗	✗	✗	✗	✗	94.6	51.8	4.0	86.7	31.0	3.4	95.5	61.7	3.0
	rescaled	✗	✗	✗	**75.8**	**14.9**	**0.2**	**94.8**	**28.4**	**0.3**	98.7	39.0	**1.9**	**88.7**	20.1	**2.2**	**99.2**	67.0	**2.5**
STL-10 — IID																			
BadAvg	unconstr.	**99.9**	6.0	**2.0**	✗	✗	✗	98.8	12.8	**2.1**	**99.9**	**96.6**	1.8	86.0	17.9	1.5	99.7	**85.7**	**1.0**
	rescaled	99.4	31.9	3.0	**98.6**	17.4	2.7	**99.4**	26.4	2.8	99.8	80.2	2.0	**99.8**	**51.2**	1.8	**99.8**	44.5	1.6
BAGEL	unconstr.	89.6	15.5	37.7	✗	✗	✗	92.3	19.9	3.3	99.3	0.5	43.2	✗	✗	✗	99.3	76.8	1.9
	rescaled	93.0	27.2	2.8	88.3	21.8	2.7	94.4	**30.9**	2.4	99.6	69.8	1.6	97.3	28.7	1.5	99.5	48.1	1.1
Naive	unconstr.	94.9	**33.1**	3.4	83.7	25.7	2.6	97.3	27.7	2.5	99.5	71.5	**1.4**	96.7	24.2	1.7	99.1	53.2	1.5
	rescaled	91.1	23.5	2.6	84.0	**34.5**	**2.4**	96.8	23.7	2.2	99.5	71.2	1.5	96.8	32.1	**1.1**	98.0	42.9	1.1
STL-10 — Non-IID (Dirichlet)																			
BadAvg	unconstr.	**99.8**	0.4	2.9	✗	✗	✗	**99.0**	70.9	**2.0**	**99.9**	**99.3**	**1.6**	76.7	19.5	2.6	**99.8**	95.4	2.6
	rescaled	87.6	26.0	3.8	74.6	**20.2**	4.7	95.9	28.0	4.7	**99.9**	74.8	2.2	**95.1**	**37.4**	2.6	**99.8**	84.2	**2.3**
BAGEL	unconstr.	91.3	24.6	39.0	✗	✗	✗	83.0	**72.6**	6.3	99.6	5.9	43.7	✗	✗	✗	97.5	**96.6**	3.1
	rescaled	83.7	25.4	**1.8**	**80.2**	16.9	2.5	92.9	40.6	2.7	99.2	76.7	2.2	90.5	22.9	2.8	99.3	78.5	**2.3**
Naive	unconstr.	83.4	**26.9**	3.0	74.9	14.4	2.5	93.0	45.1	3.2	99.7	89.2	2.1	90.5	18.4	2.8	99.7	87.9	2.8
	rescaled	83.8	19.9	**1.8**	75.1	13.3	**2.4**	93.4	37.4	2.5	99.4	81.5	2.4	88.3	24.8	**2.2**	99.3	80.8	2.7

clean rounds dilute the backdoor; after convergence, the same rounds barely perturb it. In this more realistic regime, BadAvg leads F-ASR in 13 of 18 settings.

Under *Clip+Noise* defense, unconstrained BadAvg and BAGEL are frequently reduced to a P-ASR below the viability threshold (60%), confirming that norm-based filtering is an effective first line of defense against attacks whose required scaling produces anomalously large updates, especially when the attack starts early (round 50) and the model has not converged yet. Rescaling the attack's update norm is frequently effective at bypassing Clip+Noise: among the rescaled variants, BadAvg generally achieves the highest P-ASR and, in most settings, also the highest F-ASR, though Naive's rescaled or even unconstrained variant occasionally survives comparably or better on individual dataset-distribution combinations (e.g., STL-10 IID at start 50). Even in these cases, BadAvg (rescaled) typically remains competitive with or superior to Naive's unconstrained configuration, reinforcing that the aggregation-aware design of BadAvg provides a consistent advantage over naively combining BadEncoder with model replacement.

FedTruth offers comparatively weaker protection than Clip+Noise: unlike Clip+Noise, almost no configuration under FedTruth falls below the viability threshold, and F-ASR values under FedTruth

Table 2: Ablation for BadAvg on STL-10: effect of misestimating the server learning rate η (true value $\eta = 1$) and the number of clients per round K (true value $K = 10$).

	Start 50						Start 200					
	IID			Non-IID			IID			Non-IID		
Param.	ΔAcc	P-ASR	F-ASR	ΔAcc	P-ASR	F-ASR	ΔAcc	P-ASR	F-ASR	ΔAcc	P-ASR	F-ASR
$\eta = 0.5$	6.39	61.48	53.16	6.25	39.35	34.71	5.50	85.92	60.02	2.86	89.72	76.20
$\eta = 1.5$	2.82	99.04	31.81	3.78	96.03	19.09	1.97	99.61	37.61	2.80	99.61	60.74
$K = 5$	2.68	98.90	31.88	3.11	97.58	28.38	1.62	99.75	70.61	2.95	99.59	80.30
$K = 20$	6.81	59.48	29.76	7.72	35.36	27.64	7.37	60.74	11.85	3.16	85.09	64.46

are, in most settings, noticeably higher than the corresponding values under Clip+Noise, indicating that a larger fraction of the attack survives the defense.

5.1.1 Defense Mechanisms

To further study the threat posed by BadAvg, we evaluate its effectiveness against two defenses adopted in the FCL literature. The first, *Clip+Noise* (Sun et al., 2019), operates on the server and combines weight clipping with noise injection: any client update whose $L2$ norm exceeds a given threshold is clipped, after which all updates are perturbed with Gaussian noise before weighted aggregation. The second, FedTruth (Ebron et al., 2023), works by iteratively estimating a global model from the weighted average of all client updates. The clients' weight decreases with their distance from the estimated global model, progressively moving toward convergence. This aims to reduce the importance of backdoors while partially preserving the outliers' contributions.

6 Ablation Study

BadAvg relies on estimating two server-side parameters that are not directly observable by the attacker: the aggregation learning rate η and the number of clients selected per round K. In practice, the attacker can only obtain an approximate estimate of these values, so we study the effect of over- and underestimating η (true value $\eta = 1$) and K (true value $K = 10$) for the BadAvg attack on STL-10 without defenses. The results in Table 2 show that underestimating η leads to a drop in P-ASR, especially in the Non-IID case at start 50, while overestimating η has a much smaller impact and, in several settings, even slightly improves P-ASR while keeping ΔAcc comparable or lower. In contrast, underestimating the number of selected clients ($K = 5$) has only a minor effect on P-ASR and, in most settings, yields the best or near-best F-ASR among the tested configurations, while overestimating it ($K = 20$) degrades P-ASR.

These results suggest a practical guideline for an attacker facing uncertainty about the true server-side parameters: it is preferable to assume a larger learning rate and a smaller client-selection size than the true values, rather than the opposite.

7 Conclusions

In this work, we introduced *BadAvg*, a novel aggregation-aware backdoor attack against FCL. BadAvg simulates the federated aggregation process directly into the attack optimization loop through the *FedAvg layer*, allowing the attack to be optimized against the post-aggregation model at the server. This enables BadAvg to account for the effect of benign clients' contributions and produce backdoors that survive multiple benign aggregation rounds. Our experiments showed that BadAvg can effectively poison FCL even when controlling a very small portion of the model's updates, outperforming all baselines and competitors in the undefended setting and remaining the strongest attack in the majority of the defended ones.

This work emphasizes the need for more robust defenses to make FCL safe against advanced poisoning attacks. Future work can extend BadAvg by investigating strategies to better estimate the server-side hyperparameters or the defense mechanism adopted by the FL framework. More crucially, BadAvg and future attacks should be incorporated as stress-testing tools for FCL frameworks, across different models, settings, and defenses, to evaluate model robustness and defense effectiveness against adversaries in this field.

AI USE STATEMENT

In this work, we used generative AI tools to assist with code development and polish the writing. We have not used generative AI tools for ideation, drafting the paper, or verifying the validity of our findings. We have reviewed all AI-assisted work, manually reviewing the code before employing it, verifying the results, and validating the correctness of the draft. Additionally, we ensured no plagiarism by limiting the agents to simple stylistic improvements to text we wrote. The authors verified and tested LLM-generated code for correctness, and validated every result against a problem-informed analysis. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This paper proposes a novel attack against federated contrastive learning that can overwhelm the system when no defense is in place and threaten the tested defenses. We acknowledge that malicious actors can in principle use the ideas behind BadAvg to threaten valid systems. However, we believe that finding and understanding vulnerabilities in federated contrastive learning before it becomes widespread is crucial to avoid zero-day attacks. The security of federated learning has improved over the years through this type of stress testing; this paper is one of the first to port the same principle to the contrastive learning setting.

REPRODUCIBILITY STATEMENT

In line with ICLR's open science principles, we made all our code fully reproducible. We have attached the code and results as supplementary material and will make them publicly available on GitHub when the paper is accepted. The experiments are seeded, their setting fully explained, and they can be independently re-run. Finally, all the datasets used are publicly available.

REFERENCES

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

Battista Biggio, Giorgio Fumera, and Fabio Roli. Security evaluation of pattern classifiers under attack. *IEEE transactions on knowledge and data engineering*, 26(4):984–996, 2013.

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

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, pp. 1597–1607. PmLR, 2020.

Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the 14th international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2011.

Ambra Demontis, Marco Melis, Maura Pintor, Matthew Jagielski, Battista Biggio, Alina Oprea, Cristina Nita-Rotaru, and Fabio Roli. Why do adversarial attacks transfer? explaining transferability of evasion and poisoning attacks. In *28th USENIX security symposium*, 2019.

Sheldon C Ebron, Meiying Zhang, and Kan Yang. Identifying the truth of global model: A generic solution to defend against byzantine and backdoor attacks in federated learning (full version). *arXiv preprint arXiv:2311.10248*, 2023.

Clement Fung, Chris JM Yoon, and Ivan Beschastnikh. The limitations of federated learning in sybil settings. In *23rd International symposium on research in attacks, intrusions and defenses (RAID 2020)*, pp. 301–316, 2020.

Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 7:47230–47244, 2019. doi: 10.1109/ACCESS.2019.2909068.

Yao Huang, Kongyang Chen, Jiannong Cao, Jiaxing Shen, Shaowei Wang, Yun Peng, Weilong Peng, and Kechao Cai. Bagel: Backdoor attacks against federated contrastive learning. *arXiv preprint arXiv:2311.16113*, 2023.

Jinyuan Jia, Yupei Liu, and Neil Zhenqiang Gong. Badencoder: Backdoor attacks to pre-trained encoders in self-supervised learning. In *2022 IEEE Symposium on Security and Privacy (SP)*, pp. 2043–2059. IEEE, 2022.

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

Zhenbing Liu, Fengfeng Wu, Yumeng Wang, Mengyu Yang, and Xipeng Pan. Fedcl: Federated contrastive learning for multi-center medical image classification. *Pattern Recognition*, 2023.

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

Thien Duc Nguyen, Phillip Rieger, Huili Chen, Hossein Yalame, Helen Möllering, Hossein Fereidooni, Samuel Marchal, Markus Miettinen, Azalia Mirhoseini, Shaza Zeitouni, et al. {FLAME}: Taming backdoors in federated learning. In *31st USENIX Security Symposium*, pp. 1415–1432, 2022.

Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. The German Traffic Sign Recognition Benchmark: A multi-class classification competition. In *IEEE International Joint Conference on Neural Networks*, pp. 1453–1460, 2011.

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

Vale Tolpegin, Stacey Truex, Mehmet Emre Gursoy, and Ling Liu. Data poisoning attacks against federated learning systems. In *European symposium on research in computer security*, pp. 480–501. Springer, 2020.

Jiale Zhang, Chengcheng Zhu, Di Wu 0050, Xiaobing Sun, Jianming Yong, and Guodong Long. Badfss: Backdoor attacks on federated self-supervised learning. In *IJCAI*, pp. 548–558, 2024.

A Appendix

A.1 Implementation Details

A.1.1 Encoder and Downstream Classifier

The image encoder follows a SimCLR-style design (Chen et al., 2020) adopted from BadEncoder (Jia et al., 2022): a ResNet-18 backbone $f(\cdot)$ with a projection head $g(\cdot)$ used only for the contrastive loss. The head is discarded at evaluation time, so clean accuracy and attack success rate are always measured on features from $f(\cdot)$.

Following common practice for low-resolution inputs, we adapt the backbone's initial layers, which in the standard ResNet are designed for 224×224 images and downsample the input by a factor of four before the first residual stage. We replace the initial 7×7/stride-2 convolution with a 3×3/stride-1 convolution and remove the subsequent max-pooling layer, so that 32×32 inputs reach the first residual stage at full spatial resolution. Consistently, all images are resized to 32×32 before being fed to the encoder, both during federated pre-training on CIFAR-10 and at evaluation time on the downstream tasks, whose native resolutions differ (96×96 for STL-10, variable for GTSRB).

Downstream evaluation trains a lightweight MLP ($512 \to 256 \to$ num_classes) from scratch on frozen encoder features. This head is local to the evaluation procedure and is never part of the federated model nor transmitted to the server.

Table 3: Encoder and downstream classifier configuration.

Component	**Configuration**
Backbone $f(\cdot)$	ResNet-18, adapted stem
Feature dimension	512-d, ℓ_2-normalized
Projection head $g(\cdot)$	Linear(512→512, no bias) → BatchNorm → ReLU → Linear(512→128), ℓ_2-normalized
Downstream head	MLP $512 \to 256 \to$ num_classes, trained from scratch on frozen features

A.1.2 Federated Protocol

Clients pre-train on CIFAR-10, which also serves as the attacker's shadow dataset. The client pool comprises 100 simulated clients, of which $K = 10$ participate per round. On clean rounds, all 10 are drawn uniformly at random from the full pool; on attack rounds, the malicious client's fixed identity is forced into the participant list and the remaining 9 slots are sampled uniformly from the other 99 clients. Which rounds are attack rounds follows a deterministic schedule (Table 4) rather than depending on the attacker being drawn. All runs use a fixed client-selection seed (42) for reproducibility. Non-IID partitions use a label-skewed Dirichlet allocation with $\alpha = 0.5$.

Table 4: Federated round schedule. The attack start round t_0 varies across experiments ($t_0 \in \{10, 50, 100, 200\}$).

Setting	**Value**
Clean warm-up	Rounds 1–t_0
Attack window	Rounds t_0–t_0+100
Poisoning cadence	Every 10 rounds (10 attempts)
Post-attack observation	100 clean rounds
Aggregation	FedAvg, $\eta = 1$
Local epochs / round	5 (10 for the attacker on poisoning rounds)

Defenses are applied server-side on every round: *Clip+Noise* discards updates whose norm exceeds $10\times$ the median client-update norm, clips survivors to the median, and adds DP-calibrated Gaus-

sian noise ($\varepsilon = 10$, $\delta = 10^{-3}$, noise multiplier 0.01); *FedTruth* performs coordinate-descent truth discovery over client updates using cosine distance with a collapse-resistant negative-log weighting, capped at 30 iterations.

A.1.3 Optimization Hyperparameters

The attacker fine-tunes a local copy of the global encoder with the three-term BadEncoder loss $\mathcal{L} = \mathcal{L}_1 + \lambda_1\mathcal{L}_2 + \lambda_2\mathcal{L}_3$, where $\mathcal{L}_1$ pulls triggered-image embeddings toward a fixed reference image, $\mathcal{L}_2$ stabilizes the reference embedding under augmentation, and $\mathcal{L}_3$ preserves clean-image embeddings. BadAvg additionally passes the resulting update through the differentiable FedAvg layer, and applies a norm-regularization term ($\lambda_3 = 1.0$) pulling the submitted update's magnitude toward an estimate of a benign update's norm.

Table 5: Optimization hyperparameters. lr = learning rate, wd = weight decay.

Stage	**Settings**
Local pre-training	NT-Xent, τ=0.5; Adam, lr 10^{-3}, wd 10^{-6}; batch 256
Attack optimization	SGD, lr 10^{-3}, momentum 0.9, wd 5×10^{-4}; λ_1=λ_2=1.0; batch 512; 20k shadow images
Downstream head	Adam, lr 10^{-4}; batch 64; up to 1000 epochs

To keep evaluation tractable, clean-round classifier heads are warm-started from the previous evaluation and fine-tuned for 50 epochs, whereas poisoning-round heads are always cold-trained for the full budget, so attack-round estimates are unbiased by warm-starting.

A.1.4 Trigger and Target

The trigger is a solid white 10×10 patch overwriting the bottom-right corner of the 32×32 input, applied identically at train and test time, following the BadEncoder convention (Jia et al., 2022). The attack targets a single fixed reference image per downstream task rather than a class prototype: the trigger is optimized to match that image's embedding, so the downstream classifier assigns any triggered input the reference's label. Since the encoder is always pre-trained on CIFAR-10, the CIFAR-10 and STL-10 conditions share the same reference image and target class (class 9, "truck"); GTSRB targets class 12 ("priority road").

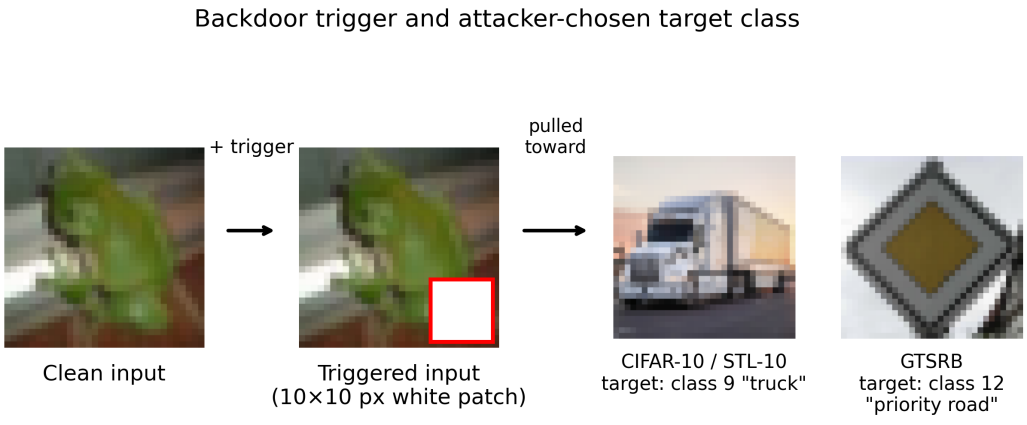


Figure 2: Left to right: a clean input; the same input with the trigger applied (red outline for illustration only); and the fixed reference image defining the target class.

A.1.5 Compute and Software

All experiments run on a single NVIDIA RTX 6000 Ada Generation GPU (48 GB), with no data- or model-parallelism, using PyTorch 2.6.0, torchvision 0.21.0, and Python 3.11.

A.2 Simulation Strategies

A simulation strategy $\mathcal{S}$ is an estimator of $\sum_{i=1}^{K-1}(C_i^t - G^t)$, the aggregate contribution of the benign clients participating in the same round, and must rely only on quantities the attacker can observe. All strategies produce a pair (μ, σ) of elementwise mean and per-parameter standard deviation describing a *single* estimated benign update, used identically regardless of strategy: the population estimate $\hat{\mathcal{S}} = (\hat{K} - 1)\,\mu$ enters the differentiable FedAvg layer as a constant, with a noise term scaled elementwise by $\sqrt{\hat{K} - 1}\,\sigma$ added on top and redrawn at every training batch. The adversarial loss is therefore evaluated against a new realization of the estimated aggregate at each step, so σ acts as an implicit regularizer favouring updates that survive a distribution of realistic benign aggregates rather than a single point estimate. We implement three strategies, differing only in how (μ, σ) are obtained.

Self-copy. The attacker reuses the clean local update $L_M^t - G^t$ it already computes each round, taking it as the single sample $\mu = L_M^t - G^t$ with an a-priori dispersion $\sigma = \sigma_{\text{rel}}\,|L_M^t - G^t|$ (relative noise scale $\sigma_{\text{rel}} = 0.1$) rather than one measured from data. It requires no knowledge beyond the attacker's own update and costs nothing, since L_M^t is already computed as part of the malicious client update. However, replicating a single update coherently $\hat{K} - 1$ times assumes every benign client moves the model in the same direction, so the estimate grows linearly where $\hat{K} - 1$ independent updates would grow as $\sqrt{\hat{K} - 1}$, overstating the true benign sum. The bias can be even worse under Dirichlet partitioning, where the attacker's own direction is least representative of the population.

Global-delta. Rather than synthesizing an estimate, this strategy inverts the server's public history. Since $G^{t-1} = G^{t-2} + \frac{\eta}{K}\sum_i (C_i^{t-2} - G^{t-2})$, the realized aggregate at the previous round is recovered by inversion as $\frac{\hat{K}}{\hat{\eta}}(G^{t-1} - G^{t-2})$, with the attacker's own contribution subtracted whenever it participated at $t-2$, and μ obtained by rescaling to a single client. As only one realized sample is available, σ is again set heuristically, to a fixed fraction (0.3) of the estimate's magnitude. The strategy is free, requiring no additional local training, and its μ is unbiased under vanilla FedAvg with correctly estimated $\hat{\eta}$ and $\hat{K}$, since the realized transition is the true benign sum. However, this strategy has multiple limitations: *i)* it needs the two most recent global models, so it is unusable on the first attack round; *ii)* $\hat{\eta}$ enters the inversion itself and not only the aggregation step, making the strategy most sensitive to a misestimate of η; *iii)* it degrades under active defenses, which reshape the aggregate into something other than the plain average the inversion assumes.

Multi-clean. The attacker trains $K - 1$ additional local encoders on its own data, all starting from the current global model G^t and each with an independent random seed governing data ordering and augmentation, so the runs differ in optimization noise. Writing $\delta^{(j)} = C^{(j)} - G^t$ for the resulting update vectors, μ and σ are their elementwise mean and standard deviation across $j = 1, \ldots, K-1$, computed separately for every parameter coordinate. This is the only strategy whose σ is measured rather than assumed, so the perturbation applied during the attack optimization reflects the dispersion actually observed across training runs. Since each copy is fine-tuned from G^t, the estimate is implicitly anchored to the round at which it was computed; once the global model has converged and the changes between subsequent rounds are small, it can therefore be computed once and reused across attack rounds, paying the $K - 1$ additional local trainings a single time. Before convergence this reuse is not optimal, and the estimate should be recomputed at each attack round. Its residual limitation is that all $K-1$ draws come from the attacker's own partition, so they capture optimization noise but not genuine cross-client data heterogeneity.

A.2.1 Comparison of the Strategies

Table 6 compares the three strategies on STL-10, under both distributions, both attack-start rounds, and all three defenses.

Table 6: **BadAvg simulation strategies on STL-10.** How the attacker estimates the benign clients' updates inside its FedAvg layer: *self-copy* reuses its own clean local update for every other client, *global-delta* reconstructs the benign mean from two consecutive global models, *multi-clean* averages several locally trained clean models. P-ASR = peak ASR inside the attack window (%); F-ASR = ASR 100 rounds after the attacker stops (%); ΔAcc = highest clean-accuracy drop observed during the attack (pp). ✗ = blocked (peak below the 60% viability threshold). **Bold** marks the best value in each column within a distribution block: highest for P-ASR and F-ASR, *lowest* for ΔAcc (a stealthier attack perturbs clean accuracy less). Ties are bolded jointly.

		start 50									start 200								
		No Defense			Clip+Noise			FedTruth			No Defense			Clip+Noise			FedTruth		
Simulation	Norm	P-ASR	F-ASR	ΔAcc	P-ASR	F-ASR	ΔAcc	P-ASR	F-ASR	ΔAcc	P-ASR	F-ASR	ΔAcc	P-ASR	F-ASR	ΔAcc	P-ASR	F-ASR	ΔAcc
STL-10 — IID																			
self-copy	unconstr.	99.8	**98.5**	3.4	✗	✗	✗	93.2	19.1	**2.1**	99.8	77.2	2.0	81.6	17.1	1.7	99.4	75.4	1.3
	rescaled	98.5	17.0	4.8	92.0	12.9	3.6	98.5	19.5	3.7	99.7	49.3	1.5	99.0	48.4	1.1	99.4	22.9	1.3
global-delta	unconstr.	99.8	28.6	3.5	✗	✗	✗	98.5	22.4	**2.1**	**99.9**	87.7	1.8	89.3	31.2	**0.9**	**99.8**	32.8	**0.8**
	rescaled	99.1	21.9	5.0	97.6	**17.8**	3.5	98.8	**32.1**	3.7	99.7	57.7	**1.1**	**99.8**	47.7	1.7	99.6	17.0	1.5
multi-clean	unconstr.	**99.9**	6.0	**2.0**	✗	✗	✗	98.8	12.8	**2.1**	**99.9**	**96.6**	1.8	86.0	17.9	1.5	99.7	**85.7**	1.0
	rescaled	99.4	31.9	3.0	**98.6**	17.4	**2.7**	**99.4**	26.4	2.8	99.8	80.2	2.0	**99.8**	**51.2**	1.8	**99.8**	44.5	1.6
STL-10 — Non-IID (Dirichlet)																			
self-copy	unconstr.	99.7	96.2	3.0	✗	✗	✗	97.7	69.9	2.3	99.8	**99.3**	2.2	✗	✗	✗	99.5	93.1	2.4
	rescaled	80.7	19.5	**2.9**	64.9	13.3	**3.5**	93.4	30.5	3.9	99.4	70.7	1.9	98.9	**50.0**	3.3	99.7	71.8	3.0
global-delta	unconstr.	**99.8**	**96.6**	3.0	✗	✗	✗	**99.1**	**79.6**	2.4	**100.0**	**99.3**	2.4	79.5	17.4	2.7	**99.8**	**97.1**	2.6
	rescaled	91.5	30.6	**2.9**	**92.5**	18.1	**3.5**	98.8	41.5	3.6	99.8	80.3	1.8	**99.1**	32.3	2.7	**99.8**	87.2	**2.2**
multi-clean	unconstr.	**99.8**	0.4	**2.9**	✗	✗	✗	99.0	70.9	**2.0**	99.9	**99.3**	**1.6**	76.7	19.5	**2.6**	**99.8**	95.4	2.6
	rescaled	87.6	26.0	3.8	74.6	**20.2**	4.7	95.9	28.0	4.7	99.9	74.8	2.2	95.1	37.4	**2.6**	**99.8**	84.2	2.3

The choice of $\mathcal{S}$ has almost no effect on whether the attack lands: P-ASR exceeds 95% for all three strategies in nearly every unblocked configuration, and the cells blocked by Clip+Noise coincide across strategies. Differences in ΔAcc are likewise small, remaining within roughly 1–2 pp of each other and not systematically favoring any strategy. What the estimate affects is instead the backdoor's persistence after the attacker stops participating.

Persistence is best compared on the rescaled rows, where the submitted update has a magnitude comparable to a benign one. There, multi-clean is the strongest or tied-strongest choice in most settings at both start rounds (e.g., F-ASR of 31.9% against 21.9% and 17.0% under IID No Defense at start 50, and 80.2% against 57.7% and 49.3% at start 200), consistent with its estimate being built from several independently trained clean models rather than from a single replicated update or a single realized transition. The F-ASR is instead comparable across all attacks on the unconstrained rows, where the large perturbation budget allowed reduces the impact of BadAvg's rescaling strategy.

A.3 Results for Early and Mid Attack-Start Rounds

Table 7 reports the two attack-start rounds deferred from the main text, $t_0 = 10$ and $t_0 = 100$, using the same metrics, attacks, and defenses as the main results.

We can notice that the starting round affects persistence far more than it affects whether the attack lands at all. P-ASR remains high across nearly all undefended settings at both start rounds, frequently above 95%, confirming that implanting the backdoor is comparatively easy regardless of how converged the encoder is. F-ASR, in contrast, clearly separates the two regimes: at $t_0 = 10$ the backdoor is almost entirely diluted by the subsequent clean rounds, with F-ASR in the single digits for most GTSRB and STL-10 configurations, whereas at $t_0 = 100$ a substantial fraction survives (e.g., BadAvg rescaled reaches an F-ASR of 76.4% on GTSRB IID, against 6.1% at $t_0 = 10$). This is consistent with the results in the main text for $t_0 \in \{50, 200\}$ and supports the same interpretation: while the model is still changing rapidly, benign updates dilute the backdoor faster than it can consolidate.

The defenses behave as in the main results. Clip+Noise blocks unconstrained BadAvg and BAGEL in most settings at both start rounds, and the effect is most systematic at $t_0 = 10$, where benign updates are themselves unstable and large-norm malicious contributions are easiest to isolate. FedTruth again offers weaker protection, leaving several configurations viable, though with consistently reduced persistence relative to the undefended case. Finally, unconstrained BAGEL shows large accuracy drops in several settings (up to 44.3 pp on GTSRB Non-IID and 44.0 pp on STL-10 Non-IID),

Table 7: **Attack outcomes for the early and mid attack-start rounds.** P-ASR = peak ASR inside the attack window (%); F-ASR = ASR 100 rounds after the attacker stops (%); ΔAcc = highest clean-accuracy drop observed during the attack (pp), floored at 0. For each metric-setting combination, the best-performing attack is highlighted in bold (ΔAcc: lowest value). ✗ = blocked (peak below the 60% viability threshold).

		start 10									start 100								
		No Defense			Clip+Noise			FedTruth			No Defense			Clip+Noise			FedTruth		
Attack	Norm	P-ASR	F-ASR	ΔAcc	P-ASR	F-ASR	ΔAcc	P-ASR	F-ASR	ΔAcc	P-ASR	F-ASR	ΔAcc	P-ASR	F-ASR	ΔAcc	P-ASR	F-ASR	ΔAcc
GTSRB — IID																			
BadAvg	unconstr.	**99.8**	**65.0**	0.7	✗	✗	✗	84.7	5.0	**1.2**	99.9	**97.4**	7.6	✗	✗	✗	98.9	71.6	**4.9**
	rescaled	99.3	6.1	1.6	**99.1**	3.6	1.2	99.4	4.0	1.5	99.8	76.4	13.1	**99.2**	**51.4**	9.8	**99.7**	71.6	10.5
BAGEL	unconstr.	✗	✗	✗	✗	✗	✗	**99.9**	5.4	1.4	**100.0**	94.7	39.8	✗	✗	✗	99.1	**90.4**	9.7
	rescaled	97.4	6.9	**0.0**	82.3	3.9	**0.4**	93.4	7.6	1.4	98.8	74.3	7.3	95.2	17.1	6.3	98.6	70.1	5.8
Naive	unconstr.	99.2	5.8	0.2	83.1	**4.4**	0.5	98.5	**9.6**	1.7	99.2	82.0	9.8	93.7	11.6	6.8	99.0	74.7	6.6
	rescaled	96.9	5.0	0.0	84.2	3.3	**0.4**	94.5	8.5	1.5	98.9	70.7	**7.2**	91.8	9.3	**6.2**	98.7	73.2	6.0
GTSRB — Non-IID (Dirichlet)																			
BadAvg	unconstr.	**99.5**	3.2	3.0	✗	✗	✗	81.5	3.6	**0.4**	**99.9**	**95.8**	9.3	✗	✗	✗	98.7	33.9	9.6
	rescaled	97.8	4.0	4.0	**98.8**	4.0	1.3	**98.8**	3.6	2.7	99.0	25.3	16.2	**98.7**	6.1	13.8	**99.0**	42.9	14.3
BAGEL	unconstr.	✗	✗	✗	✗	✗	✗	✗	✗	✗	97.9	54.5	44.3	✗	✗	✗	92.0	48.8	15.6
	rescaled	86.9	**5.6**	**2.1**	77.8	3.5	**0.0**	89.3	6.2	0.5	97.1	29.7	**8.9**	89.3	**19.4**	9.3	97.0	**54.3**	**7.5**
Naive	unconstr.	96.2	4.4	3.0	77.4	3.4	1.9	90.2	4.2	1.0	97.2	42.5	11.4	93.8	8.2	9.2	98.1	33.8	11.4
	rescaled	86.8	5.4	**2.1**	79.7	**4.0**	0.2	90.5	**7.5**	0.5	97.3	18.5	9.1	87.4	5.1	**8.4**	96.9	47.7	**7.5**
CIFAR-10 — IID																			
BadAvg	unconstr.	**99.7**	**93.9**	4.1	✗	✗	✗	89.6	**42.8**	3.6	**99.6**	**53.5**	0.9	✗	✗	✗	88.3	**63.1**	0.4
	rescaled	82.0	14.2	3.3	71.9	13.2	4.7	72.4	25.6	4.2	65.3	47.0	0.7	71.0	29.5	0.6	75.5	44.1	0.6
BAGEL	unconstr.	✗	✗	✗	✗	✗	✗	✗	✗	✗	97.5	0.5	43.1	✗	✗	✗	82.3	9.4	1.6
	rescaled	69.5	28.4	3.3	69.1	11.0	4.0	75.5	30.3	4.0	63.8	47.8	0.7	75.2	25.8	0.5	82.5	34.6	0.4
Naive	unconstr.	98.5	24.0	1.8	82.8	18.3	0.6	77.6	31.3	0.0	96.3	11.1	0.2	95.3	**45.1**	0.0	**98.0**	39.6	0.0
	rescaled	97.3	29.5	**0.0**	60.4	20.0	**0.4**	86.2	27.8	**0.0**	96.6	30.6	**0.0**	**95.9**	31.7	**0.0**	92.4	43.7	**0.0**
CIFAR-10 — Non-IID (Dirichlet)																			
BadAvg	unconstr.	**99.6**	**29.5**	5.7	✗	✗	✗	73.6	**23.6**	5.1	**99.6**	5.3	1.3	✗	✗	✗	98.5	26.4	1.5
	rescaled	✗	✗	✗	✗	✗	✗	✗	✗	✗	72.2	27.1	1.4	**93.0**	**27.5**	1.2	70.2	11.6	1.7
BAGEL	unconstr.	69.7	14.2	33.4	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	80.0	22.3	4.1
	rescaled	✗	✗	✗	✗	✗	✗	✗	✗	✗	61.0	29.4	1.2	67.9	19.0	1.0	71.6	23.4	0.9
Naive	unconstr.	✗	✗	✗	✗	✗	✗	**88.1**	12.8	0.0	89.5	25.7	0.7	84.3	13.8	0.0	**98.7**	**43.6**	0.3
	rescaled	✗	✗	✗	✗	✗	✗	77.9	19.4	**0.0**	84.3	19.9	**0.1**	84.0	20.8	**0.0**	97.9	42.9	**0.0**
STL-10 — IID																			
BadAvg	unconstr.	**99.9**	**78.2**	2.0	✗	✗	✗	93.9	**41.4**	1.9	**99.9**	2.3	1.0	72.7	25.3	0.7	99.6	19.5	**0.1**
	rescaled	98.4	11.8	2.4	**93.9**	13.0	1.2	**98.5**	19.0	**1.4**	99.4	**68.9**	0.8	**99.8**	37.9	0.3	**99.8**	17.3	0.9
BAGEL	unconstr.	60.7	14.6	28.4	✗	✗	✗	84.7	27.7	1.9	97.2	0.3	43.6	✗	✗	✗	99.3	23.3	1.8
	rescaled	97.5	21.2	**1.5**	70.7	14.6	**1.1**	93.5	30.3	2.0	98.0	36.9	0.1	96.6	**45.6**	**0.1**	99.4	20.1	0.4
Naive	unconstr.	93.6	19.3	2.8	71.9	17.1	1.4	95.8	34.2	1.9	99.1	46.5	0.5	96.9	20.1	0.5	99.6	34.8	0.2
	rescaled	97.0	16.8	**1.5**	82.0	**18.4**	1.2	92.5	27.5	2.1	98.6	50.6	**0.0**	97.1	30.4	**0.1**	99.4	**40.0**	0.6
STL-10 — Non-IID (Dirichlet)																			
BadAvg	unconstr.	**99.8**	1.8	1.5	✗	✗	✗	**92.2**	35.8	2.3	**99.9**	**98.5**	2.2	✗	✗	✗	**99.7**	89.6	**1.6**
	rescaled	89.9	16.2	2.3	**80.5**	18.2	2.7	84.6	20.1	**1.9**	94.8	50.9	1.8	**93.5**	**30.2**	2.4	98.7	67.8	2.3
BAGEL	unconstr.	72.2	**30.5**	26.1	✗	✗	✗	70.6	**51.1**	4.5	96.8	41.8	44.0	✗	✗	✗	96.3	**93.9**	3.8
	rescaled	✗	✗	✗	✗	✗	✗	74.5	20.2	2.7	97.2	59.1	1.3	82.0	23.5	**1.1**	96.1	63.0	**1.6**
Naive	unconstr.	61.0	24.1	1.6	61.7	**19.8**	**2.6**	82.3	24.1	2.7	97.8	78.2	1.6	85.9	23.2	1.2	97.4	77.9	2.1
	rescaled	62.4	19.8	**1.3**	✗	✗	✗	71.9	21.5	2.8	97.5	63.3	**1.2**	83.8	24.7	1.2	96.4	66.5	1.7

confirming that its fixed scaling factor trades stealthiness for magnitude, whereas the rescaled variants keep ΔAcc within a few percentage points throughout.