Your Neighbors Know: Leveraging Local Neighborhoods for Backdoor Detection in Decentralized Learning

Sayan Biswas[1] **Antoine Boutet**[2] **Davide Frey**[3] **Romaric Gaudel**[3]

Rachid Guerraoui[1] **Maxime Jacovella**[1,*] **Anne-Marie Kermarrec**[1]

Dimitri Lerévérend[3] **François Taïani**[3]

Martijn de Vos[1]

[1]EPFL [2]Inria, INSA Lyon, CITI [3]Univ. Rennes, Inria, CNRS, IRISA

Authors are listed in alphabetical order.

Abstract

Decentralized learning (DL) is an emerging machine learning paradigm where nodes collaboratively train models without a central server. However, the collaborative nature of DL makes it vulnerable to backdoor attacks, where a model is taught to behave normally on standard inputs while executing hidden, malicious actions when encountering data with specific triggers. Backdoor attacks in DL remain understudied and existing defenses often overlook DL constraints. We introduce ARGUS, a novel backdoor detection framework native to DL that requires neither a central coordinator nor prior knowledge of the trigger. In ARGUS, honest nodes locally analyze received model updates to identify potential backdoor triggers. Nodes then collectively share their triggers with their neighbors and use a structural similarity metric to separate true backdoors from false alarms induced by data heterogeneity. A key insight is that false positive triggers exhibit inconsistencies across participants while true positive ones show consistent patterns. Model updates that fail this collaborative test are rejected, and persistently malicious senders are eventually evicted. We provide the first theoretical convergence guarantees for a DL-specific backdoor detection mechanism, showing that filtering out suspicious model updates with high probability preserves a convergence rate comparable to standard DL. We implement and evaluate ARGUS on three standard datasets and against three state-of-the-art baselines. Across settings, ARGUS reduces attack success rates by up to 90 points compared to no defense, while preserving model utility within 5 percentage points of an omniscient oracle. Furthermore, the effectiveness of ARGUS compared to baselines improves as data heterogeneity increases.

1 Introduction

Decentralized learning (DL) is a collaborative machine learning (ML) approach that enables a set of nodes to train an ML model collectively without the need for any central server [1, 2, 3]. In each round of DL, nodes independently train their model on their local private datasets and exchange the

*Corresponding author: maxime.jacovella@epfl.ch.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

resulting model updates with their neighbors according to a topology. Each node then aggregates the received updates locally, and the resulting model is used as a starting point for the next round. This process repeats until the models converge. Because DL requires no central coordinator, it is inherently scalable and fault-tolerant, and removes the need to trust any single party. DL is increasingly adopted in high-stakes domains such as healthcare, energy and finance [2, 4].

Unfortunately, the collaborative nature of DL makes it vulnerable to attacker nodes attempting to undermine model training by injecting adversarial updates that hurt model utility for other nodes [5, 6]. One such example is *backdoor attacks*, by which attacker nodes manipulate their local models to misclassify inputs containing a specific trigger pattern, redirecting them to a target label of their choice. This is done without compromising the model's accuracy on normal, clean inputs [7]. In DL, such backdoor triggers are continuously injected and spread across the network through seemingly legitimate model updates. Figure 1 illustrates the effectiveness of backdoor attacks in DL with 16 nodes and just one attacker (see Appendix G.1 for further experimental details) in a Non-Independent and Non-Identically Distributed (NIID) setting on the CIFAR-10 dataset. The models of honest nodes are significantly compromised, reaching over 60% average Attack Success Rate (ASR) and thus showing that backdoors remain a serious threat in DL. The average ASR for nodes directly connected to the attacker is even more pronounced, ultimately exceeding 80%. While backdoor attacks and defenses have been extensively studied in federated learning (FL) [8, 9], these solutions are largely inapplicable in DL because of the absence of a central server. Therefore, backdoor defense in DL remains unsolved.

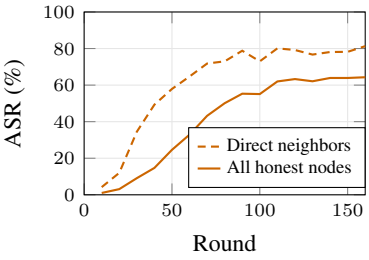

Figure 1: The average Attack Success Rate (ASR) of the backdoor attack in a 16-node network with one attacker, using the CIFAR-10 dataset in a NIID setting.

We address this shortcoming and introduce ARGUS, a novel framework for backdoor detection in DL that operates without any central coordinator and without requiring prior knowledge of the backdoor trigger. With ARGUS, nodes detect and filter backdoored model updates before they are aggregated into local models, and eventually identify and evict attacker nodes from participating. Figure 2 shows the overall workflow of ARGUS as executed by an honest node. ARGUS works in two phases: *(i)* a *local trigger detection* phase (Step 3), in which each node independently attempts to reverse-engineer a backdoor trigger from each received update, using only their local dataset; and *(ii)* a *collaborative verification* phase (Steps 4–5), in which nodes cross-validate their recovered triggers with the ones found by their neighbors. Depending on the similarity between triggers, incoming updates are accepted or rejected (Step 6). This collaborative verification phase is crucial: data heterogeneity causes local detection to produce some false positives, and those triggers recovered from honest nodes are structurally inconsistent across detecting nodes whereas true backdoor triggers are not, allowing ARGUS to cleanly separate the two. By leveraging the collaborative nature of DL, ARGUS effectively suppresses backdoor attacks without compromising model utility. We implement ARGUS and conduct extensive experiments on three standard vision datasets (CIFAR-10, FEMNIST, and TinyImageNet), against three state-of-the-art baselines (MULTI-KRUM, a Byzantine-robust aggregation, and the two existing backdoor defenses targeting DL: BADFL and the clipping defense of Syros et al. [10]), and across varying levels of data heterogeneity, network topologies, and attacker node counts. We find that ARGUS reduces the ASR from over 70% to below 7% across all settings, while retaining test accuracy on non-backdoored samples within 5 percentage points of an omniscient oracle defense.

Contributions. In summary, our main contributions are:

1. We introduce ARGUS, the first backdoor detection framework where nodes in DL reverse-engineer and collaborate to identify potential backdoor triggers (Section 3).
2. We provide a convergence analysis of ARGUS, showing a competitive convergence rate alongside a characterization of the effect of false positives introduced by ARGUS (Section 4). To the best of our knowledge, this is the first analysis for DL with *asymmetric* link failure that captures the impact of the failure probability in the convergence bound.
3. We implement ARGUS and evaluate it on three standard datasets and against three state-of-the-art baselines (Section 5). Our experiments demonstrate the effectiveness of ARGUS in detecting backdoor attacks while preserving model utility, especially in NIID settings.

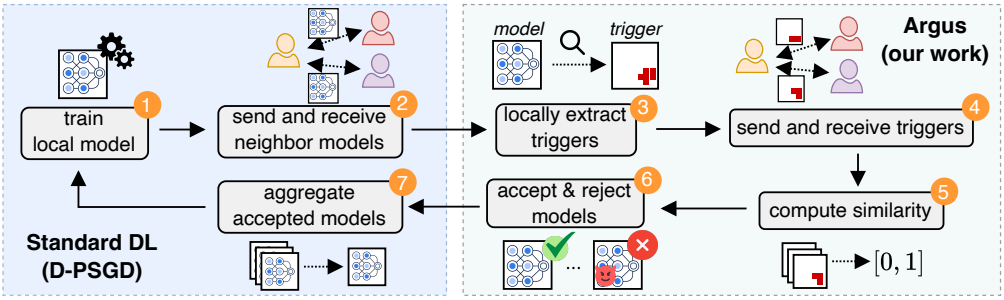

Figure 2: The workflow of ARGUS during a single round in DL as executed by an honest node.

2 Background and problem formulation

We now introduce the standard DL algorithm and backdoors, and then elaborate on the shortcomings of existing approaches that defend against backdoor attacks in collaborative ML algorithms.

Decentralized learning. A set of n nodes collaboratively trains a model f_θ to minimize the global risk $\mathcal{L}(\theta) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{(x,y)\sim\mathcal{P}_i}\ell(f_\theta(x), y)$, where ℓ is the point-wise loss and each node i holds a local dataset $\mathcal{D}_i$ drawn from $\mathcal{P}_i$, with $\mathcal{P}_1, \ldots, \mathcal{P}_n$ *a priori* NIID. Communication follows an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ in synchronous rounds. At round t, every node i, owning a model θ_i^t, performs an arbitrary number G of stochastic gradient descent (SGD) steps on θ_i^t to obtain $\theta_i^{t+1/2}$ (Step 1 in Figure 2), sends it to its neighbors $N(i)$ (Step 2), and aggregates the received updates (Step 7) as $\theta_i^{t+1} = \sum_j \mathcal{W}_{ij}\theta_j^{t+1/2}$ with $\mathcal{W}$ a row-stochastic gossip matrix consistent with $\mathcal{G}$ ($\mathcal{W}1 = 1$ and $\mathcal{W}_{ij} \neq 0 \iff \{i,j\} \in \mathcal{E} \cup \{(i,i)\}$). This follows the standard D-PSGD algorithm [1].

In this paper, we consider a K-class image classification task with an input space $\mathcal{X} \subseteq \mathbb{R}^{C\times H\times W}$, where C, H, and W denote the number of channels, height, and width of the input images, respectively. The output space is defined over K discrete classes as $\mathcal{Y} = \{1, \ldots, K\}$.

Backdoor attacks. A backdoor attack [7, 11] aims to obtain a model θ^b that retains a high clean accuracy (*i.e.*, accuracy on clean data) but predicts a fixed target label t^* on any input to which a predefined trigger τ^* has been applied: $\mathbb{P}_{(x,y)\sim\mathcal{P}}[f_{\theta^b}(x) = y] \approx 1$ and $\mathbb{P}_{(x,y)\sim\mathcal{P}}[f_{\theta^b}(x \oplus \tau^*) = t^*] \approx 1$ where $x \oplus \tau^*$ denotes applying τ^* to x. Following [7, 12, 13, 14], we focus on *spatially localized*, patch-like triggers occupying a contiguous region of at most $p \cdot H \cdot W$ pixels with $p \ll 1$. We consider a non-contiguous trigger in Appendix G.3 and discuss semantic backdoors [11] in Section 6.

Shortcomings of existing solutions. Many works address backdoor attacks in FL, where a central server aggregates client updates. This is done via robust aggregation [15, 16, 13, 17], server-side anomaly detection [18, 19, 20, 21, 22, 23, 24, 25], training-time hardening [26], or trigger-based detection [27, 28, 29, 30, 31, 32] (see Appendix B for a thorough discussion). All of them rely in some way on a central orchestrator with a global view of every update of each round. Removing this assumption is far from trivial: each honest node now only sees its own neighborhood and the local statistics are biased by its own data, which starves distance-based aggregators, weakens clusterings, and breaks voting quorums. DETRIGGER [32] is the most adaptable approach for decentralization and tries to reverse-engineer a candidate trigger from incoming model updates. It inspires our local-detection step (Section 3.2.1), but we show in Section 5.3 that, if applied without any adaptation, it leads to excessive rejections and sharp accuracy drops in NIID settings. In DL, on the other hand, prior work has mostly focused on Byzantine robustness [6, 33] and privacy [34, 35, 36]. BADFL [37] is framed as a backdoor defense for DL, but actually only considers the untargeted gradient-perturbation attack of [38], a model poisoning attack. To the best of our knowledge, only Syros et al. [10] (referred to hereafter as P2PCD for brevity) explicitly aims for backdoor robustness in fully decentralized settings. It proposes a two-norm clipping scheme that applies separate clipping bounds to peer updates and to the local model, thus not actually detecting backdoors but rather trying to limit their spread. However, P2PCD heavily relies on well-chosen clipping thresholds, with no heuristic given to set them. Crucially, DL opens an opportunity: an attacker's update is observed by *all* its neighbors, who can collaborate to detect backdoors. ARGUS explores this avenue.

Algorithm 1: ARGUS: Protocol executed by node i at round t

Input: Local model θ_i^t, trust states $\{\mathcal{S}_j\}_{j\in N(i)}$, learning rate η, no. SGD steps G, thresholds γ, ξ, min. confirmations κ

Compute gradient g_i^t on a batch from $\mathcal{D}_i$; Update $\theta_i^{t+1/2} \leftarrow \theta_i^t - \eta g_i^t$ // G local SGD steps

Send $\theta_i^{t+1/2}$ to $N(i)$; receive $\{\theta_j^{t+1/2}\}_{j\in N(i)}$ // Communication

▷ *Stage 1 - Backdoor mitigation*

$\mathcal{A}_i^t \leftarrow \emptyset$ // Set of accepted neighbors

foreach $j \in N(i)$ *s.t.* $\mathcal{S}_j \neq$ EJECTED **do**

  $(\text{flagged}, \hat{\tau}_j^i) \leftarrow \text{LOCALDETECT}(\theta_j^{t+1/2}, \gamma)$ // Section 3.2.1

  $\text{reject} \leftarrow \text{flagged} \wedge \text{COLLABORATIVEVERIFY}(j, \hat{\tau}_j^i, \xi, \kappa)$ // Section 3.2.2

  if *not* reject *and* $\mathcal{S}_j =$ TRUSTED **then** $\mathcal{A}_i^t \leftarrow \mathcal{A}_i^t \cup \{j\}$

  $\mathcal{S}_j \leftarrow \text{UPDATETRUSTSTATE}(\mathcal{S}_j, \text{reject})$ // Section 3.2.3

▷ *Stage 2 - Average over accepted neighbors only*

$\theta_i^{t+1} \leftarrow \frac{1}{|\mathcal{A}_i^t|+1}\left(\theta_i^{t+1/2} + \sum_{j\in\mathcal{A}_i^t} \theta_j^{t+1/2}\right)$ // Re-scaled averaging

return θ_i^{t+1} and updated $\{\mathcal{S}_j\}_{j\in N(i)}$

3 Design of ARGUS

We now first formalize the system and threat model in Section 3.1 and then outline the ARGUS workflow in Section 3.2.

3.1 System and threat model

A subset $\mathcal{M} \subset \mathcal{V}$ of nodes is malicious, sharing a common target label t^* and backdoor trigger τ^*. We refer to the nodes in $\mathcal{M}$ as *attacker nodes*. They use a single, shared trigger and target label for their backdoor, a standard assumption in this domain [11]. Attacker nodes know each other, as well as both the communication graph $\mathcal{G}$ and the algorithm implemented by ARGUS, but do not know the honest nodes' local data. They comply with the protocol from an external perspective: they participate in every communication round and never abstain from sending a model update. Internally, however, they may deviate from honest training to pursue their goal, *i.e.*, they may modify their local training data and training procedure. They may also downscale or disregard some of the updates they receive to preserve the backdoor, as integrating them all is known to weaken the attack's effectiveness [11].

Let $\mathcal{H} := \mathcal{V}\backslash\mathcal{M}$ denote the set of *honest nodes*, who adhere to the ARGUS protocol and do not know which node is a potential attacker. Honest nodes know the *local* graph topology up to distance 2 and, hence, may communicate lightweight information with neighbors of their neighbors. This is a common assumption in DL literature [39, 40, 41]. We also assume that, for every node $i \in \mathcal{V}$, at least $\kappa + 1$ of its neighbors are honest, where κ will control the minimum number of confirmations required for collaborative verification (Section 3.2.2). This ensures the collaborative step always has enough witnesses. Note that this assumption is local, and so is the impact of its violation. We explore adversarial settings where this assumption does not hold in Appendix G.6.

3.2 ARGUS workflow

ARGUS provides a three-component defense against backdoor attacks in DL. Its architecture is summarized in Figure 2 and the full per-round protocol is available in Algorithm 1. At a high level, ARGUS comprises three key components: *(i)* **local trigger detection** (Step 3 in Figure 2), in which each honest node independently scans its neighbors' updates for potential backdoors, returning a binary decision on whether or not to reject the update and a possible trigger pattern; *(ii)* **collaborative verification** (Steps 4–6 in Figure 2), in which flagged updates are cross-validated by querying the sender's other neighbors and comparing their recovered triggers via a structural similarity metric; and *(iii)* **a trust state machine** (omitted from Figure 2 for brevity), used by honest nodes to maintain a per-neighbor trust level and eject nodes detected as persistently malicious, thus reducing compute overhead. We note that ARGUS is modular: the exact local detection and collaborative verification implementations may be altered without changing the overall protocol. We describe below a possible

instantiation. Similarly, ARGUS is agnostic to the underlying DL algorithm and can be layered on top of any gossip-based training procedure. We instantiate it on D-PSGD [1] throughout this work, as it is the most widely adopted baseline in the DL literature.

3.2.1 Local trigger detection

When node i receives a model update $\theta_j^{t+1/2}$, it first tries to reverse-engineer, for each output label $y \in \mathcal{Y}$, a small input perturbation that misclassifies inputs to class y using a small validation set $\mathcal{D}_i^{val} \subseteq \mathcal{D}_i$. For this, we adopt the per-update recovery step of DETRIGGER [32] proposed for FL, but adapt it to a decentralized setting. This proceeds in three steps: *(i) Initial candidate:* for each source label $z \in \mathcal{Y} \setminus \{y\}$ and $(x, z) \in \mathcal{D}_i^{val}$, we compute the input-layer gradient $g_y(x, z) = \nabla_x[f_\theta(x)_y - f_\theta(x)_z]$, average it across these examples, min-max normalize to $[-1, 1]$ and select the source label whose average has highest energy to obtain an initial trigger $\hat{\tau}^0$ with associated binary mask m_y; *(ii) Optimizing the trigger:* we perform R projected gradient steps: $\hat{\tau}^{(r+1)} = \text{clip}\left[\hat{\tau}^{(r)} - \tilde{\eta}\tanh\left(\nabla_{\hat{\tau}}\mathcal{L}_{CE}(f_\theta(x \oplus \hat{\tau}^{(r)}), y)\right)\right]_{[-1,1]} \odot m_y$ where $\odot$ is the Hadamard product; *(iii) Flagging suspicious updates:* we compute the resulting trigger's ASR on $\mathcal{D}_i^{val}$, return the target label with the highest ASR and the corresponding $\hat{\tau}_j^i$ if its ASR exceeds a threshold γ.

It is noteworthy that a recovered trigger might not correspond to an actual backdoor trigger. Because false positives (FP) will be filtered by collaborative verification, γ can be set aggressively low. Similarly, $|\mathcal{D}_i^{val}|$ can be chosen small, or step *(i)* can be performed on a small subset of it (in our experiments in Section 5, $\mathcal{D}_i^{val}$ contains at most one sample per class).

Local detection alone is not enough. A node whose local data is dominated by class y naturally produces updates that shift the decision to y. A neighbor may interpret this change as a backdoor targeting y, producing many FP detections under NIID data (see ablation in Section 5.3 where local detection alone achieves low ASR but severely damages accuracy when heterogeneity increases). Crucially, these False Positive (FP)-triggers are *inconsistent across detecting nodes*: each detecting node's input gradient depends on its local data, so two neighbors of the same node end up with two independent random-looking triggers. True backdoors, in contrast, are *structurally consistent* since detecting nodes are tracing the same implanted trigger.
This is formulated in the insight below:

Insight 1 (Structural similarity of recovered triggers)**.** *Let $\hat{\tau}_a$ and $\hat{\tau}_b$ be triggers reverse-engineered by two honest nodes given the* same *update.* ***(TP)*** *If the update contains a genuine backdoor with trigger τ^*, then $\hat{\tau}_a$ and $\hat{\tau}_b$ will tend to be structurally similar to each other (and to τ^*).* ***(FP)*** *Otherwise, $\hat{\tau}_a$ and $\hat{\tau}_b$ will likely appear as independent random masks.*

Figure 3 motivates this insight on CIFAR-10 with two attacker nodes trying to inject a 3×3 bottom-right square backdoor: the pixels in TP triggers concentrate near the true backdoor while those in FP triggers are scattered and look random.

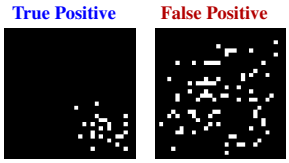

Figure 3: Example True Positive (TP) and FP reverse-engineered triggers from CIFAR-10 when the real backdoor trigger is a bottom-right 3×3 pixel square.

3.2.2 Collaborative verification

When node i flags an update from neighbor j, it asks j's neighbors to confirm. Each $\ell \in N(j)\backslash\{i\}$ either replies with its own candidate $\hat{\tau}_j^\ell$ or with NOT-SUSPICIOUS. Node i rejects the update if, and only if, at least κ peers returned a trigger $\hat{\tau}_j^\ell$ with $\texttt{sim}(\hat{\tau}_j^i, \hat{\tau}_j^\ell) \geq \xi$, where $\texttt{sim}$ is a similarity metric.

Trigger similarity metric. Comparing raw triggers pixel-by-pixel is unreliable because candidate triggers may differ in exact pixel positions and intensity even when sharing the same spatial structure. ARGUS assumes a similarity metric that *(i)* is sensitive to high-level spatial structure, *(ii)* can abstract away from background noise and intensity scaling, and *(iii)* provides a principled way to set the threshold ξ independently of the ground-truth backdoor trigger. We instantiate $\texttt{sim}$ as the Structural similarity (SSIM) [42] of the two trigger energy maps after retaining only the top-k highest-energy pixels. We provide the full mathematical definition of the trigger similarity metric in Appendix C.1.

Resilience to attackers. Attacker nodes may try to interfere in two ways: *(i) Shielding a fellow attacker node* by sending a NOT-SUSPICIOUS message or a random trigger when queried about it. Detection still succeeds as long as at least κ of the other neighbors confirm, which is possible since the threat model gives at least $\kappa + 1$ honest neighbors. *(ii) Framing an honest node* by attempting to confirm an FP trigger. This gives a lower bound on κ, above the expected number of attacker nodes.

3.2.3 Trust state machine

The collaborative verification protocol involves nodes detecting backdoors independently every round. To leverage the history of collaborative decisions, each node in ARGUS maintains a per-neighbor trust state machine. In addition to reducing computations for nodes that are already known to be backdoored, this helps reduce the impact of occasional misses in backdoor detection. Each node i maintains a state $\mathcal{S}_j \in \{\text{TRUSTED}, \text{SUSPECTED}, \text{EJECTED}\}$ per neighbor j. j moves from TRUSTED to SUSPECTED after k_1 consecutive confirmed rejections by i (its updates are then rejected by default), and is permanently EJECTED if at least k_2 further rejections occur in the next k_3 rounds (otherwise it returns to TRUSTED). More details and a visual representation are available in Appendix C.2.

3.3 Setting the ARGUS hyperparameters

ARGUS hinges on two main hyperparameters, both related to the collaborative verification phase: *(i)* the minimum number of required similarity confirmations κ, and *(ii)* the similarity threshold ξ. κ can be set directly from the graph degree and the threat model, given the expected minimum number of honest neighbors. ξ is more sensitive since setting it too high lets backdoors pass, while too low rejects honest updates. Crucially, we show that ξ can be calibrated *before training* using only the image dimensions (H, W), the clipping size k, the SSIM window size w and the model architecture, with no access to the trigger or training data. Motivated by Insight 1, we model FP triggers as independent samples from a Gaussian random field with correlation σ on the $H \times W$ grid, where σ depends solely on the model architecture. We set ξ as a quantile of the resulting null SSIM distribution, estimated by Monte Carlo sampling over $M = 10^4$ pairs. This is a one-time, offline cost that merely takes a few seconds on CPU. On CIFAR-10 with our parameters ($H = W = 32, k = 51, w = 11, \sigma = 2$), we obtain $\xi \approx 0.42$. Per-dataset values are in Table 3 and the full derivation in Appendix D.

Figure 4 validates this calibration: across all rounds and pairs of detecting nodes flagging the same source, FP and TP similarities are cleanly separated by $\xi = 0.42$, with limited overlap.

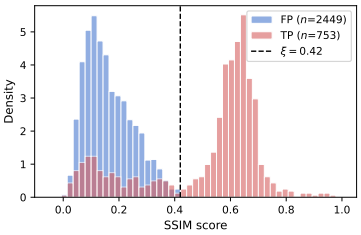

Figure 4: **Empirical FP vs. TP trigger similarities** on CIFAR-10 ($\alpha = 0.5$, $m = 2$ attacker nodes, $n = 16$ nodes) and calibrated threshold $\xi = 0.42$.

4 Convergence analysis

We now analyze the convergence of ARGUS. We proceed through the following steps. First, we make a simple Independent and Identically Distributed (IID) model for accept/reject events in ARGUS (Assumption 2). Then, we provide theoretical guarantees on malicious and honest node ejection probabilities (Proposition 4). We then highlight a gossip contraction factor induced by randomized filtering (Proposition 6). Finally, we use this contraction factor with a standard DL convergence analysis to derive convergence guarantees for ARGUS (Theorem 7) and an end-to-end guarantee (Corollary 8). For brevity, the proofs for the node-ejection bounds (Section 4.1) are postponed to Appendix I, and the proofs of the convergence-rate properties (Section 4.2) to Appendix H.

4.1 Node ejection bounds

Intuitively, if all nodes reject all external updates then learning fails, while if malicious updates are often accepted then backdoors persist. We model this trade-off with a two-rate assumption.

Assumption 2 (IID two-rate detection model)**.** *For $i \in \mathcal{H}, j \in N(i)$, let $\delta_{i,j}^t := \mathbb{I}_{\{j \in \mathcal{A}_i^t\}}$ denote whether node i accepts the update from node j at round t (and set $\delta_{i,i}^t := 1$). Assume the random*

variables $\{\delta_{i,j}^t\}$ are independent over t and satisfy $\delta_{i,j}^t \sim \mathrm{Bern}(p_{\mathrm{fn}})$ $(j \in \mathcal{M})$ and $\delta_{i,j}^t \sim \mathrm{Bern}(1 - p_{\mathrm{fp}})$ $(j \in \mathcal{H})$, where p_{fn} is the per-round probability of accepting a malicious update and p_{fp} is the per-round probability of rejecting an honest update.

Remark 3. *In practice, model rejections are temporally correlated due to the inherent dependence of model updates over time. However, when a backdoor is detected from an honest node, we observe random behavior that aligns well with this assumption (*cf.*, Figure 3). In addition, our threat model assumes that attacker nodes may behave arbitrarily; thus, attackers could send uncorrelated updates, which is handled by Assumption 2. Moreover, Appendix I.3 relaxes independence to a first-order Markov model: at the correlations measured in our experiments (*0.09 *for honest and* 0.31 *for attacker neighbors), the ejection bounds below remain under* 1%*.*

Assumption 2 allows us to derive bounds on the probability of ejecting honest and attacker nodes:

Proposition 4. *In any round T, for thresholds $k_1, k_2 \leq k_3$, let $\pi(p) := p^{k_1} \sum_{r=k_2}^{k_3} \binom{k_3}{r} p^r (1-p)^{k_3-r}$, and define A_T^{hon} (resp. B_T^{mal}) as the event where an honest node reaches (resp. an attacker node does not reach) the state* EJECTED *by the end of round T. With Assumption 2 and $q_{\mathrm{fn}} := 1 - p_{\mathrm{fn}}$, we have*

$$\mathbb{P}(B_T^{\mathrm{mal}} = 1) \leq (1 - p_{\mathrm{fn}}\pi(q_{\mathrm{fn}}))^{\lfloor T/(k_1+k_3+1) \rfloor}\,; \mathbb{P}(A_T^{\mathrm{hon}} = 1) \leq \max\{T - k_1 - k_2 + 1, 0\}\,\pi(p_{\mathrm{fp}}).$$

The first bound is stated for $k_2 = 1$, as in our experiments; Appendix I.3 covers all thresholds.

Remark 5. *In our experiments on CIFAR-10, we obtain $p_{\mathrm{fp}} = 1.1\%$ and $p_{\mathrm{fn}} = 20.0\%$. After* 50 *iterations, this upper-bounds the probability that a node has ejected a given honest neighbor by* 0.02%*, and the probability that it has not ejected a given attacker neighbor by* 33.7%*; the tighter bound of Appendix I.3 brings the latter below* 0.01%*. Moreover, the values of k_1, k_2, and k_3 can be modified to adjust these probabilities depending on the use case.*

4.2 Convergence rates

Using Proposition 4, we can show that, with high probability, attacker nodes are removed after a few rounds while honest nodes remain in the system. After these few rounds, ARGUS behaves like standard DL with random link failures. Similar scenarios have been studied in the literature [43], but without considering rescaling of the received models as we do in Stage 2 of Algorithm 1, and those scenarios fail to derive the impact of the failure probability on the convergence term.

Let $(S^t)_{i,j} := {}^{\delta_{i,j}^t}/_{(1+|\mathcal{A}_i^t|)}$, where $\delta_{i,j}^t$ is as defined in Assumption 2. Thus, S^t is the random mixing matrix induced by Stage 2 of Algorithm 1, obtained by masking the base gossip matrix $\mathcal{W}$ according to rejected updates and re-normalizing. Using this notation, we analyze how link failures affect the spectral gap, a key quantity controlling the convergence of decentralized optimization algorithms [3, 44]:

Proposition 6 (Spectral gap of $\mathbb{E}\big((S^t)^\top S^t\big)$)**.** *For an* undirected *and d-regular communication graph $\mathcal{W}$, if the entries of S^t satisfy Assumption 2, let $\phi_{p_{\mathrm{fp}}}(\nu) := (a - cd) + 2b\,\nu + c\,\nu^2$ and $\psi_{p_{\mathrm{fp}}}(\mu) := \phi_{p_{\mathrm{fp}}}((d+1)\mu - 1)$ for some constants a, b, c (*cf.*, Equation* (4) *to* (6)*) depending on p_{fp}. Let $1 = \mu_1 \geq \mu_2 \geq \cdots \geq \mu_n$ be the eigenvalues of $\mathcal{W}$. Then, we obtain the following identities:*

$$\lambda_1\big(\mathbb{E}[(S^t)^\top S^t]\big) = 1 \quad \textit{and} \quad \rho := \lambda_2\big(\mathbb{E}[(S^t)^\top S^t]\big) = \max\big\{\psi_{p_{\mathrm{fp}}}(\mu_2)\,,\ \psi_{p_{\mathrm{fp}}}(\mu_n)\big\} \in [0, 1).$$

We can now state the main convergence result of ARGUS by combining the mean-square contraction factor ρ from Proposition 6 with the standard descent-consensus decomposition used in DL convergence analysis.

Theorem 7 (Convergence rate of ARGUS)**.** *Consider Assumption 2, $G = 1$ and that all attackers have been ejected while honest neighbors remain* TRUSTED*. Moreover, assume that the setting of Proposition 6 holds, and that mixing matrices S^t are independent of the gradients computed in the same round. Assume each local objective $\mathcal{L}_i$ is L-smooth and that SGD steps satisfy $\forall x \in \mathbb{R}^d, t \in \mathbb{N} : \mathbb{E}\big\|g_i^t(x) - \nabla\mathcal{L}_i(x)\big\|^2 \leq \sigma^2$ and $\frac{1}{n}\sum_{i=1}^n \big\|\nabla\mathcal{L}_i(x) - \nabla\mathcal{L}(x)\big\|^2 \leq \varsigma^2$. Define $\Delta_0 := \mathcal{L}\left(\overline{\theta}^0\right) - \mathcal{L}^*$. Then, for a constant stepsize η (Equation* (12)*), and constants C_ρ and $\beta(d, p_{\mathrm{fp}})$, $\frac{1}{n}\sum_{i\in\mathcal{V}} \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\left[\|\nabla\mathcal{L}(\theta_i^t)\|^2\right]$ is bounded by:*

$$\mathcal{O}\left(\frac{L}{T}\Delta_0 + \sqrt{\frac{L\Delta_0 C_\rho \beta(d, p_{\mathrm{fp}})(\varsigma^2 + \sigma^2)}{nT}} + \sqrt[3]{\frac{L^2\Delta_0^2 C_\rho(\varsigma^2 + \sigma^2)}{T^2}}\right).$$

Theorem 7 shows how the probability p_{fp} of rejecting honest updates affects the convergence rate, captured by the variables C_ρ and $\beta(d, p_{\text{fp}})$ (defined in Equation (9) and (16) for brevity).

Finally, both results combine into a guarantee that covers the whole run, attack phase included:

Corollary 8 (End-to-end convergence under attack, informal)**.** *Fix $\delta \in (0,1)$ and $\epsilon > 0$. Under the assumptions of Proposition 4 and Theorem 7, and if the loss and the disagreement of honest models remain bounded while attackers are present (Assumption 19), there exist $T_{\text{ej}} = \mathcal{O}(\log(m/\delta))$ and $T_{\text{cv}} = \mathcal{O}(1/(\delta^2\epsilon^2))$ such that, running* ARGUS *for $T = T_{\text{ej}} + T_{\text{cv}}$ rounds,*

$$\mathbb{P}\left(\frac{1}{|\mathcal{H}|\,T_{\text{cv}}}\sum_{i\in\mathcal{H}}\sum_{t=T_{\text{ej}}}^{T-1}\left\|\nabla\mathcal{L}(\theta_i^t)\right\|^2 \leq \epsilon\right) \;\geq\; 1-\delta.$$

The confidence $1-\delta$ and the accuracy ϵ can thus be set independently. The closed-form version, proof, and the case of honest neighbors leaving TRUSTED are in Appendix J. In particular, the formal version of Corollary 8 lies in Corollary 21.

5 Experimental evaluation

We next present our experimental evaluation and answer the following two questions: **(Q1)** How effective is ARGUS at detecting backdoors across datasets and compared to baselines (Section 5.2)? **(Q2)** What does collaborative verification add beyond local detection alone, in terms of balancing attack suppression and model utility, using the CIFAR-10 dataset (Section 5.3)?

We provide additional experiments in Appendix G where we further analyze the effectiveness of ARGUS, *e.g.*, by experimenting with different trigger shapes, a moving trigger designed to evade detection, different topologies, network sizes and attacker node counts. We also discuss the computational and communication overhead of ARGUS.

5.1 Experimental setup

We now outline the key aspects of our experiment setup and provide additional setup details in Appendix E. We also make our code available for reproducibility.[2]

Datasets, models and topologies. We evaluate ARGUS on CIFAR-10 [45] with a ResNet-8 model [46], FEMNIST [47] with a 2-layer Convolutional Neural Networks (CNN) model [48] (two 5×5 convolutional blocks followed by 2×2 max-pooling and a 512-unit fully connected layer) and TinyImageNet [49] with a ResNet-18 model, covering image sizes 32×32, 28×28 and 64×64 and 10, 62 and 200 classes respectively. The default topology is a 3-regular graph with $n = 16$ nodes, but we vary both the degree and the graph family in Appendix G.5. In line with recent work [50, 51], we realize label heterogeneity across nodes with the Dirichlet distribution parametrized by $\alpha > 0$: a lower α represents higher heterogeneity. We consider $\alpha \in \{0.25, 0.5, \infty\}$ on CIFAR-10 and $\alpha = 1$ on TinyImageNet. FEMNIST incorporates real-world heterogeneity through its feature distribution.

Attack configuration. We consider $m \in \{1, 2, 3, 4\}$ attacker nodes sharing the same target label and backdoor trigger. They poison a random fraction of their training set by adding the backdoor and changing the target label, and ignore incoming updates to preserve the backdoor [11]. Attacker nodes know the defense: during collaborative verification, they shield other attacker nodes and try to frame honest nodes by sending a random trigger. The default trigger is a 3×3 (6×6 for TinyImageNet) bottom-right-corner square. We experiment with different trigger characteristics in Appendix G.3.

Baselines. We compare ARGUS against: *(i)* NO DEFENSE, the standard D-PSGD algorithm without any defense, *(ii)* ORACLE, an idealized defense that knows the attacker nodes and rejects only their updates. It is not practical but gives an upper bound on achievable clean accuracy with ASR≈ 0, *(iii)* MULTI-KRUM [15], a Byzantine-robust aggregation method commonly used as a baseline in backdoor defenses [17, 29, 25], applied locally over each neighborhood, *(iv)* BADFL [37], an approach introduced as a backdoor defense for DL, but evaluated only against untargeted poisoning attacks (see Section 2) that we re-implement and evaluate against true backdoors, and *(v)* P2PCD (Peer-to-Peer Clipping Defense), the two-norm clipping defense proposed for DL [10], which applies

[2] Code available at `https://github.com/sacs-epfl/argus`.

Table 1: **Main results.** Clean Test Accuracy (CA) (↑ is better) and Attack Success Rate (ASR) (↓ is better) with $m = 2$ attacker nodes out of $n = 16$ nodes, reporting mean and std over 3 seeds.

	CIFAR-10, $\alpha = 0.25$		CIFAR-10, $\alpha = 0.5$		FEMNIST		TinyImageNet	
Defense	CA [%]	ASR [%]	CA [%]	ASR [%]	CA [%]	ASR [%]	CA [%]	ASR [%]
ORACLE	48.4±1.3	0.1±0.0	54.7±0.8	0.1±0.0	73.5±0.4	0.0±0.0	48.2±0.7	0.0±0.0
NO DEFENSE	35.3±2.2	78.0±6.2	43.2±1.3	72.4±4.7	69.4±0.6	99.8±0.1	28.3±0.3	46.9±2.7
BADFL	30.4±2.1	50.4±5.2	37.4±1.2	45.7±5.4	69.2±0.5	99.8±0.1	27.1±0.3	45.6±3.1
P2PCD	31.0±2.0	29.3±8.0	38.4±0.7	24.9±12.2	68.0±0.5	57.7±2.8	20.0±1.0	9.4±1.2
MULTI-KRUM	41.5±3.5	17.8±8.5	47.4±0.5	15.1±11.0	**73.2±0.4**	**0.0±0.0**	37.3±0.6	**0.0±0.0**
ARGUS (Ours)	**45.6±2.7**	**6.9±3.2**	**53.3±1.0**	**1.7±1.0**	72.8±0.4	**0.0±0.0**	**43.3±2.0**	**0.0±0.0**

a tighter clipping norm to peer updates than to local ones, and skips clipping during the early *agreement phase*.

More details about the baselines and their hyperparameters are provided in Appendix F.

Metrics. We evaluate all methods using the following two metrics, averaged across honest nodes: *(i)* Clean Test Accuracy (CA), measuring model utility on non-backdoored inputs; and *(ii)* Attack Success Rate (ASR), the fraction of non-target test images misclassified as t^* when the trigger is applied, excluding images already misclassified without it. We also report the Rejection Rate (Rej. Rate), the fraction of peer updates rejected during training, and for ARGUS specifically, the per-update False Positive Rate (FPR) and True Positive Rate (TPR) to characterize detection quality. All experiments are run with 3 random seeds and we report aggregated values and associated standard deviations.

5.2 The effectiveness of ARGUS against baselines

Table 1 reports the CA and ASR with $m = 2$ attackers in a 16-node 3-regular graph, for all three datasets. For CIFAR-10, we include two NIID heterogeneity levels ($\alpha = 0.25$ and $\alpha = 0.5$) and report IID results in Appendix G.2. Across every configuration, ARGUS stays within 5 percentage points of the ORACLE baseline's CA, demonstrating that backdoor filtering does largely preserve model utility. By contrast, BADFL and P2PCD lose up to 20 CA percentage points compared to ORACLE, while MULTI-KRUM, the strongest competitor, is 10.9 points below ORACLE on TinyImageNet (while showing minimal loss on the easier FEMNIST task). On the attack side, ARGUS reduces ASR to below 7% on CIFAR-10 and to near-zero on FEMNIST and TinyImageNet. MULTI-KRUM matches ARGUS on the latter two, which we attribute to its distance-based rejection: attackers do not integrate honest updates in order to preserve their backdoor, so their model updates drift away, which is exactly what MULTI-KRUM is built to detect. Under heterogeneity, however, MULTI-KRUM performs worse and leaves a residual ASR of 15.1% on CIFAR-10 ($\alpha = 0.5$) (vs. 1.7% for ARGUS) and 17.8% for $\alpha = 0.25$. Overall, ARGUS is the only defense that simultaneously achieves low ASR and near-ORACLE CA across all evaluated settings, and its margin over the strongest baseline widens as data heterogeneity increases.

5.3 The contribution of collaborative verification

We now isolate the contribution of collaborative verification by introducing LOCAL DEF., an ARGUS variant that rejects every update flagged by local detection, without cross-validating with neighbors. Figure 5 reports the CA (left), rejection rate of all model updates (middle) and ASR (right) on CIFAR-10 for three heterogeneity levels: IID ($\alpha \to \infty$), moderate ($\alpha = 0.5$) and high ($\alpha = 0.25$). With NO DEFENSE, the backdoor reaches high ASR across all settings ($> 70\%$), confirming the intrinsic vulnerability of DL to backdoors. Its CA remains low as malicious updates deviate from others because attacker nodes do not incorporate the update they receive to preserve the backdoor (see Section 5.1). LOCAL DEF. successfully suppresses the backdoor, leading to a near-zero ASR, but its rejection rate is excessively high, especially in NIID settings: 43.5% with IID data and over 80% under high heterogeneity. As discussed in Section 3.2.1, data heterogeneity introduces FPs, resulting in accuracy loss: for $\alpha = 0.25$, LOCAL DEF. is 13 CA points lower than ORACLE. ARGUS' collaborative verification step helps close that gap across heterogeneity levels.

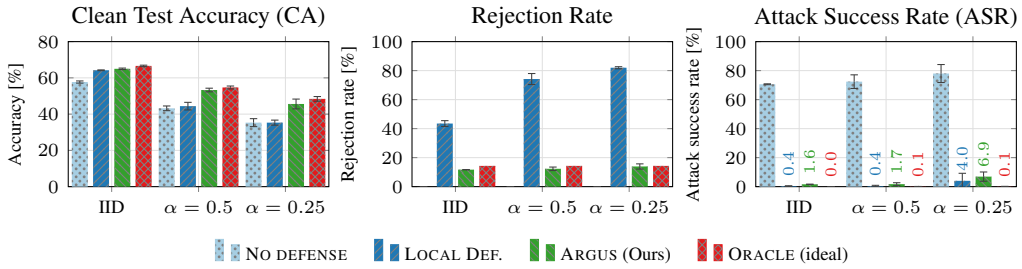

Figure 5: **Local detection alone is unsatisfactory.** The CA (left, ↑ is better), rejection rate (middle, ↓ is better) and ASR (right, ↓ is better) for CIFAR-10 with $m = 2$ attackers out of $n = 16$ nodes, and for varying heterogeneity levels. We consider different baseline settings and report std over 3 seeds.

6 Conclusions

We presented ARGUS, a novel backdoor detection framework for DL that requires neither a central coordinator nor prior knowledge of the trigger. ARGUS combines local trigger reverse-engineering with collaborative cross-validation among neighboring nodes to cleanly separate true backdoors from false positives induced by data heterogeneity. We provide the first convergence analysis for a DL-native backdoor defense that captures the impact of rejecting updates, modeled as asymmetric link failures, within the convergence bound. Our experimental evaluation highlights that across three datasets and three baselines, ARGUS reduces attack success rates to below 7% while preserving clean accuracy within 5 points of an omniscient oracle. Thus, ARGUS is a key step toward backdoor-resilient DL.

Limitations and future work. We identify two limitations motivating future work. First, rejected updates may still carry a useful learning signal. Collaborative verification already limits this loss (for $\alpha = 0.25$ on CIFAR-10, it lowers the rejection rate from 82% with LOCAL DEF. to 14%, on par with ORACLE; Figure 5). However, *beyond rejection*, a collaboratively verified trigger could be used to *filter* or *unlearn* the backdoor from an update rather than just discarding it, thereby retaining its useful information. Feasibility in a centralized [12] and FL setting [52, 53] has been shown. Second, *semantic backdoor triggers* [11] fall outside the considered patch model: our local detection searches for a small additive input-space perturbation, whereas semantic triggers are neither additive nor directly comparable by structural similarity. Studying these triggers in DL, even just their applicability, remains an open problem.

Acknowledgments

The work of the co-authors affiliated with EPFL has been funded by the Swiss National Science Foundation, under the project "FRIDAY: Frugal, Privacy-Aware and Practical Decentralized Learning", SNSF proposal No. 10.001.796, as well as the project "TruBrain", proposal No. 20CH21_218778. This work has also been supported in part by the Groupe La Poste, sponsor of the Inria Foundation, in the framework of the FedMalin Inria Challenge, and by the ANR 22-PECY-0002 IPOP (Interdisciplinary Project on Privacy) project of the Cybersecurity PEPR.

References

[1] Xiangru Lian, Ce Zhang, Huan Zhang, Cho-Jui Hsieh, Wei Zhang, and Ji Liu. Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent. *Advances in Neural Information Processing Systems*, 30, 2017. URL `https://proceedings.neurips.cc/paper_files/paper/2017/file/f75526659f31040afeb61cb7133e4e6d-Paper.pdf`.

[2] Enrique Tomás Martínez Beltrán, Mario Quiles Pérez, Pedro Miguel Sánchez Sánchez, Sergio López Bernal, Gérôme Bovet, Manuel Gil Pérez, Gregorio Martínez Pérez, and Alberto Huertas Celdrán. Decentralized federated learning: Fundamentals, state of the art, frameworks, trends, and challenges. *IEEE Communications Surveys & Tutorials*, 25(4):2983–3013, 2023.

[3] Martijn De Vos, Sadegh Farhadkhani, Rachid Guerraoui, Anne-marie Kermarrec, Rafael Pires, and Rishi Sharma. Epidemic Learning: Boosting Decentralized Learning with Randomized Communication. *Advances in Neural Information Processing Systems*, 36:36132–36164, December 2023.

[4] Chamani Shiranthika, Parvaneh Saeedi, and Ivan V Bajić. Decentralized learning in healthcare: a review of emerging techniques. *IEEE Access*, 11:54188–54209, 2023.

[5] Carmela Troncoso, Marios Isaakidis, George Danezis, and Harry Halpin. Systematizing decentralization and privacy: Lessons from 15 years of research and deployments. *Proceedings on Privacy Enhancing Technologies*, 2017(4):404–426, October 2017. ISSN 2299-0984. doi: 10.1515/popets-2017-0056. URL http://dx.doi.org/10.1515/popets-2017-0056.

[6] Cheng Fang, Zhixiong Yang, and Waheed U. Bajwa. Bridge: Byzantine-resilient decentralized gradient descent. *IEEE Transactions on Signal and Information Processing over Networks*, 8: 610–626, 2022. doi: 10.1109/TSIPN.2022.3188456.

[7] 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.

[8] Xueluan Gong, Yanjiao Chen, Qian Wang, and Weihan Kong. Backdoor attacks and defenses in federated learning: State-of-the-art, taxonomy, and future directions. *IEEE Wireless Communications*, 30(2):114–121, 2022.

[9] Zhaozheng Li, Jiahe Lan, Zheng Yan, and Erol Gelenbe. Backdoor attacks and defense mechanisms in federated learning: A survey. *Information Fusion*, 123:103248, 2025.

[10] Georgios Syros, Gokberk Yar, Simona Boboila, Cristina Nita-Rotaru, and Alina Oprea. Backdoor attacks in peer-to-peer federated learning. *ACM Trans. Priv. Secur.*, 28(1), December 2024. ISSN 2471-2566. doi: 10.1145/3691633. URL https://doi.org/10.1145/3691633.

[11] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In Silvia Chiappa and Roberto Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 2938–2948. PMLR, 26–28 Aug 2020. URL https://proceedings.mlr.press/v108/bagdasaryan20a.html.

[12] Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y. Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 707–723, 2019. doi: 10.1109/SP.2019.00031. URL https://ieeexplore.ieee.org/document/8835365.

[13] Chulin Xie, Minghao Chen, Pin-Yu Chen, and Bo Li. CRFL: certifiably robust federated learning against backdoor attacks. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, Proceedings of Machine Learning Research, pages 11372–11382. PMLR, 2021. URL http://proceedings.mlr.press/v139/xie21a.html.

[14] Bochuan Cao, Jinyuan Jia, Chuxuan Hu, Wenbo Guo, Zhen Xiang, Jinghui Chen, Bo Li, and Dawn Song. Data free backdoor attacks. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 23881–23911. Curran Associates, Inc., 2024. doi: 10.52202/079017-0753. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/2a7e91c6e4b68325d9884a7469804837-Paper-Conference.pdf.

[15] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/f4b9ec30ad9f68f89b29639786cb62ef-Paper.pdf.

[16] Dong Yin, Yudong Chen, Kannan Ramchandran, and Peter L. Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In *International Conference on Machine Learning*, 2018.

[17] Thien Duc Nguyen, Phillip Rieger, Huili Chen, Hossein Yalame, Helen Möllering, Hossein Fereidooni, Samuel Marchal, Markus Miettinen, Azalia Mirhoseini, Shaza Zeitouni, Farinaz Koushanfar, Ahmad-Reza Sadeghi, and Thomas Schneider. FLAME: Taming Backdoors in Federated Learning. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 1415–1432, 2022. ISBN 978-1-939133-31-1.

[18] Phillip Rieger, Thien Duc Nguyen, Markus Miettinen, and Ahmad-Reza Sadeghi. Deepsight: Mitigating backdoor attacks in federated learning through deep model inspection. In *29th Annual Network and Distributed System Security Symposium, NDSS 2022, San Diego, California, USA, April 24-28, 2022*. The Internet Society, 2022. URL https://www.ndss-symposium.org/ndss-paper/auto-draft-205/.

[19] Thuy Dung Nguyen, Anh Duy Nguyen, Thanh-Hung Nguyen, Kok-Seng Wong, Huy Hieu Pham, Truong Thao Nguyen, and Phi Le Nguyen. Fedgrad: Mitigating backdoor attacks in federated learning through local ultimate gradients inspection. In *2023 International Joint Conference on Neural Networks (IJCNN)*, pages 01–10, 2023. doi: 10.1109/IJCNN54540.2023.10191655.

[20] Yihang Lin, Pengyuan Zhou, Zhiqian Wu, and Yong Liao. Mitigating backdoors in federated learning with fld. *2024 5th International Seminar on Artificial Intelligence, Networking and Information Technology (AINIT)*, pages 530–535, 2023. URL https://api.semanticscholar.org/CorpusID:257255592.

[21] Wenke Huang, Mang Ye, Zekun Shi, Guancheng Wan, He Li, and Bo Du. Parameter disparities dissection for backdoor defense in heterogeneous federated learning. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 120951–120973. Curran Associates, Inc., 2024. doi: 10.52202/079017-3843. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/db0ee27cb50dd9087b133f6e7d28a90e-Paper-Conference.pdf.

[22] Ning Wang, Shanghao Shi, Yang Xiao, Yimin Chen, Y. Thomas Hou, and Wenjing Lou. Boba: Boosting backdoor detection through data distribution inference in federated learning. *CoRR*, abs/2407.09658, 2024. doi: 10.48550/ARXIV.2407.09658. URL https://doi.org/10.48550/arXiv.2407.09658.

[23] Binbin Ding, Penghui Yang, and Sheng-Jun Huang. Feddlad: A federated learning dual-layer anomaly detection framework for enhancing resilience against backdoor attacks. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2025, Montreal, Canada, August 16-22, 2025*, pages 5021–5029. ijcai.org, 2025. doi: 10.24963/IJCAI.2025/559. URL https://doi.org/10.24963/ijcai.2025/559.

[24] Jidong Yuan, Qihang Zhang, Naiyue Chen, Shengbo Chen, and Baomin Xu. A multi-granularity clustering approach for federated backdoor defense with the adam optimizer. In *International Joint Conference on Artificial Intelligence*, 2025.

[25] Jiahao Xu, Zikai Zhang, and Rui Hu. Detecting backdoor attacks in federated learning via direction alignment inspection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20654–20664, June 2025.

[26] Tiansheng Huang, Sihao Hu, Ka-Ho Chow, Fatih Ilhan, Selim Furkan Tekin, and Ling Liu. Lockdown: Backdoor defense for federated learning with isolated subspace training. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=V5cQH7JbGo.

[27] Sebastien Andreina, Giorgia Azzurra Marson, Helen Möllering, and Ghassan Karame. Baffle: Backdoor detection via feedback-based federated learning. In *2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS)*, pages 852–863, 2021. doi: 10.1109/ICDCS51616.2021.00086.

[28] Jinyuan Jia, Zhuowen Yuan, Dinuka Sahabandu, Luyao Niu, Arezoo Rajabi, Bhaskar Ramasubramanian, Bo Li, and Radha Poovendran. Fedgame: A game-theoretic defense against backdoor attacks in federated learning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 53090–53111. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/a6678e2be4ce7aef9d2192e03cd586b7-Paper-Conference.pdf.

[29] Songze Li and Yanbo Dai. BackdoorIndicator: Leveraging OOD Data for Proactive Backdoor Detection in Federated Learning. In *33rd USENIX Security Symposium (USENIX Security 24)*, pages 4193–4210, 2024. ISBN 978-1-939133-44-1.

[30] Phillip Rieger, Torsten Krauß, Markus Miettinen, Alexandra Dmitrienko, and Ahmad-Reza Sadeghi. Crowdguard: Federated backdoor detection in federated learning. In *31st Annual Network and Distributed System Security Symposium, NDSS 2024, San Diego, California, USA, February 26 - March 1, 2024*. The Internet Society, 2024. URL https://www.ndss-symposium.org/ndss-paper/crowdguard-federated-backdoor-detection-in-federated-learning/.

[31] Haibin Zheng, Wenjie Shen, and Jinyin Chen. HoneyFL: Using Honeypots to Catch Backdoors in Federated Learning. *IET Image Processing*, 19(1):e70201, 2025. ISSN 1751-9667. doi: 10.1049/ipr2.70201.

[32] Kichang Lee, Yujin Shin, Jonghyuk Yun, Songkuk Kim, Jun Han, and JeongGil Ko. Detrigger: A gradient-centric approach to backdoor attack mitigation in federated learning, 2025. URL https://arxiv.org/abs/2411.12220.

[33] El Mahdi El-Mhamdi, Sadegh Farhadkhani, Rachid Guerraoui, Arsany Guirguis, Lê-Nguyên Hoang, and Sébastien Rouault. Collaborative learning in the jungle (decentralized, byzantine, heterogeneous, asynchronous and nonconvex learning). *Advances in neural information processing systems*, 34:25044–25057, 2021.

[34] Edwige Cyffers, Mathieu Even, Aurélien Bellet, and Laurent Massoulié. Muffliato: Peer-to-peer privacy amplification for decentralized optimization and averaging. *Advances in Neural Information Processing Systems*, 35:15889–15902, 2022.

[35] Sayan Biswas, Mathieu Even, Anne-Marie Kermarrec, Laurent Massoulié, Rafael Pires, Rishi Sharma, and Martijn de Vos. Noiseless privacy-preserving decentralized learning. *Proceedings on Privacy Enhancing Technologies*, 2025.

[36] Sayan Biswas, Davide Frey, Romaric Gaudel, Anne-Marie Kermarrec, Dimitri Lerévérend, Rafael Pires, Rishi Sharma, and François Taïani. Low-cost privacy-preserving decentralized learning. *Proceedings on Privacy Enhancing Technologies*, 2025.

[37] Yuan Yuan, Anhao Zhou, Xiao Zhang, Yifei Zou, Yangguang Shi, and Dongxiao Yu. Badfl: Mitigating model poisoning in decentralized federated learning. *IEEE Transactions on Computers*, 74:3968–3979, 2025. URL https://api.semanticscholar.org/CorpusID:281078519.

[38] Jingwei Sun, Ang Li, Louis DiValentin, Amin Hassanzadeh, Yiran Chen, and Hai Li. FL-WBC: Enhancing robustness against model poisoning attacks in federated learning from a client perspective. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=96uH8HeGb9G.

[39] Danilo Menegatti, Alessandro Giuseppi, Sabato Manfredi, and Antonio Pietrabissa. A discrete-time multi-hop consensus protocol for decentralized federated learning. *IEEE Access*, 11: 80613–80623, 2023. doi: 10.1109/ACCESS.2023.3299443.

[40] Sayan Biswas, Anne-Marie Kermarrec, Rafael Pires, Rishi Sharma, and Milos Vujasinovic. Secure aggregation meets sparsification in decentralized learning. *arXiv preprint arXiv:2405.07708*, 2024.

[41] Kai Zhang, Xuanyu Cao, and Khaled B Letaief. Decentralized federated learning with energy harvesting devices. *IEEE Transactions on Wireless Communications*, 25:15392–15407, 2026.

[42] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. doi: 10.1109/TIP.2003.819861. URL https://ieeexplore.ieee.org/document/1284395.

[43] Erik G. Larsson and Nicolò Michelusi. Unified Analysis of Decentralized Gradient Descent: A Contraction Mapping Framework. *IEEE Open Journal of Signal Processing*, 6:507–529, 2025. ISSN 2644-1322. doi: 10.1109/OJSP.2025.3557332.

[44] Anastasia Koloskova, Nicolas Loizou, Sadra Boreiri, Martin Jaggi, and Sebastian Stich. A Unified Theory of Decentralized SGD with Changing Topology and Local Updates. In *Proceedings of the 37th International Conference on Machine Learning*, pages 5381–5393. PMLR, November 2020.

[45] Alex Krizhevsky. Learning multiple layers of features from tiny images. *University of Toronto*, 05 2012.

[46] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. doi: 10.1109/CVPR.2016.90.

[47] Sebastian Caldas, Sai Meher Karthik Duddu, Peter Wu, Tian Li, Jakub Konečný, H. Brendan McMahan, Virginia Smith, and Ameet Talwalkar. Leaf: A benchmark for federated settings, 2019. URL https://arxiv.org/abs/1812.01097.

[48] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

[49] Ya Le and Xuan S. Yang. Tiny imagenet visual recognition challenge, 2015. URL http://vision.stanford.edu/teaching/cs231n/reports/2015/pdfs/yle_project.pdf.

[50] Sai Aparna Aketi, Abolfazl Hashemi, and Kaushik Roy. Global update tracking: A decentralized learning algorithm for heterogeneous data. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 48939–48961. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/98f8c89ae042c512e6c87e0e0c2a0f98-Paper-Conference.pdf.

[51] Sai Aparna Aketi, Sakshi Choudhary, and Kaushik Roy. Averaging rate scheduler for decentralized learning on heterogeneous data. In *The Second Tiny Papers Track at ICLR 2024*, 2024. URL https://openreview.net/forum?id=w9ZzNmWmjA.

[52] Chen Wu, Xian Yang, Sencun Zhu, and Prasenjit Mitra. Toward cleansing backdoored neural networks in federated learning. In *2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS)*, pages 820–830, 2022. doi: 10.1109/ICDCS54860.2022.00084.

[53] Chen Wu, Sencun Zhu, Prasenjit Mitra, and Wei Wang. Unlearning backdoor attacks in federated learning. In *2024 IEEE Conference on Communications and Network Security (CNS)*, pages 1–9, 2024. doi: 10.1109/CNS62487.2024.10735680.

[54] Matthew Jagielski and Alina Oprea. Does differential privacy defeat data poisoning? In *ICLR 2021 Workshop on Distributed and Private Machine Learning (DPML)*, 2021. URL https://dp-ml.github.io/2021-workshop-ICLR/files/23.pdf. Workshop paper.

[55] Min Du, Ruoxi Jia, and Dawn Song. Robust anomaly detection and backdoor attack detection via differential privacy. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SJx0q1rtvS.

[56] Akash Dhasade, Anne-Marie Kermarrec, Rafael Pires, Rishi Sharma, and Milos Vujasinovic. Decentralized learning made easy with decentralizepy. In *Proceedings of the 3rd Workshop on Machine Learning and Systems*, EuroMLSys '23, page 34–41. ACM, May 2023. doi: 10.1145/3578356.3592587. URL http://dx.doi.org/10.1145/3578356.3592587.

[57] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Randomized gossip algorithms. *IEEE Transactions on Information Theory*, 52(6):2508–2530, June 2006. ISSN 1557-9654. doi: 10.1109/TIT.2006.874516.

A Symbol table

We provide a table summarizing all the symbols used throughout this work in Table 2.

Table 2: List of symbols used in this work.

Symbol	Description	Source
Learning setup and threat model		
$\mathcal{X}, \mathcal{Y}, K$	Input space, output space, and number of classes	Section 2
$\mathcal{V},\ n$	Node set and total number of nodes	Section 2
$\mathcal{D}_i,\ \mathcal{P}_i$	Local dataset and local data distribution at node i	Section 2
$\theta^t \in \mathbb{R}^{n\times d}$	Parameter matrix (local instances are $\theta_i^t \in \mathbb{R}^d$)	Section 2
$\mathcal{L},\ \mathcal{L}_i,\ \ell$	Global objective, local objective, and point-wise loss	Section 2
$g_i^t,\ \eta$	Stochastic gradient at node i and learning rate	Section 2
G	Number of local SGD steps	Algorithm 1
$\mathcal{G} = (\mathcal{V}, \mathcal{E}), N(i)$	Communication graph and neighborhood of node i	Section 2
$\mathcal{W}$	Base gossip/mixing matrix	Section 2
$\mathcal{M},\ \mathcal{H}$	Malicious and honest node sets	Section 3.1
$t^*,\ \tau^*,\ \theta^b$	Target label, true backdoor trigger, and backdoored model	Section 3.1
$x \oplus \tau^*,\ p$	Trigger application operator and trigger area ratio in $p \cdot H \cdot W$.	Section 3.1
Local detection and collaborative verification		
$\mathcal{D}_i^{val}$	Local validation subset used by node i	Section 3.2.1
$\tilde{\eta}$	Learning rate of the local trigger detection	Section 3.2.1
$\hat{\tau}_j^i,\ m_y,\ \gamma$	Trigger recovered by node i from sender j, binary support mask, and local detection threshold	Section 3.2.1
$\tilde{\tau},\ k$	Top-k clipped trigger map.	Section 3.2.2
$\mathrm{sim}(\cdot,\cdot),\ \xi,\ \kappa$	Trigger similarity metric, similarity threshold, and minimum confirmations	Section 3.2.2
$H,\ W,\ w$	Image height, image width, and window size in SSIM	Section 3.2.2
$\mu_x,\ \sigma_x^2,\ \sigma_{ab}$	Local SSIM moments/covariance (with stabilizing constants C_1, C_2)	Section 3.2.2
Trust state machine and ejection events		
$\mathcal{S}_j$	Trust state assigned by a node to neighbor j	Section 3.2.3
$\mathcal{A}_i^t$	Set of accepted neighbors at round t	Algorithm 1
$\delta_{i,j}^t$	Acceptance indicator for edge (i, j)	Assumption 2
$p_{\mathrm{fp}},\ p_{\mathrm{fn}}$	Honest false-positive rejection probability and attacker false-negative acceptance probability	Assumption 2
$k_1,\ k_2,\ k_3,\ \pi(p)$	Ejection rule thresholds and window success probability	Proposition 4
$\varrho_{\mathrm{fp}},\ \varrho_{\mathrm{fn}}$	Lag-1 correlation of rejections on honest and attacker edges	Appendix I.3
Convergence analysis		
S^t	Random mixing matrix after model rejection	Section 4.2
$\phi_{p_{\mathrm{fp}}},\ \psi_{p_{\mathrm{fp}}}$	Polynomial function linking p_{fp} to the gossip spectrum	Proposition 6
$T_{\mathrm{ej}},\ T_{\mathrm{cv}}$	Ejection round and number of further rounds of the end-to-end guarantee	Corollary 8
Threshold calibration		
$\tilde{E} = G_\sigma \star Z$	Smooth Gaussian-field null model for false-positive trigger energy maps	Equation (1)
σ	Correlation in the Gaussian-field model	Section 3.2.2
α	Dirichlet (data) heterogeneity parameter	Section 5.1

B Related work on backdoor attacks and defenses in FL

A large body of work addresses backdoor attacks in FL, where a central server aggregates client updates. They can be broadly grouped into four categories. *Robust aggregations* have been proposed

to replace the server's average rule with a Byzantine-tolerant aggregator, like Krum or coordinate-wise trimmed mean/median [15, 16]. FLAME [17] combines clustering with adaptive clipping and calibrated Gaussian noise to remove potential backdoors, while CRFL [13] adds norm clipping and noise to the aggregated model, yielding certified robustness for backdoors with limited magnitude. All of these methods rely on access to the full set of client updates collected at the server in each round, to compute pairwise distances [15, 17], adapt to parameter distributions [16] or to reach an aggregated global model [13]. In DL, each node could attempt to run them locally over its neighborhood, but the resulting sample size and bias introduced by local data heterogeneity severely damage their statistical robustness.

Another line of work focuses on *server-side model inspection and clustering* [18, 19, 20, 23, 24]. These approaches all require a single entity to collect every client's update and compute global statistics and/or clusterings. For example, AlignIns [25] flags updates based on their pairwise cosine similarity with the global aggregate and sign agreement, FDCR [21] clusters clients by Fisher-information-weighted gradient discrepancies from the global update, and BoBa [22] relies on a server with simultaneous visibility of client updates to infer data distributions, perform overlapping clustering, compute trust scores, and then carry out weighted aggregation.

Training-time hardening adapts the local training protocol so that backdoors are less likely to form or transfer. Lockdown [26] restricts each client to train in an isolated sparse subspace, which could be transferred to DL. Its per-round fusion step, however, assumes a coordinator can compare clients' subspace masks and purge parameters that lack quorum support, which is not transferable to a fully serverless setting.

Trigger-based and proactive detection methods attempt to detect backdoors by actively probing the received updates. Some approaches rely on applying dummy backdoor triggers and studying how they evolve through the training process [29, 31]. This would require a central authority or give attacker nodes knowledge of the dummy trigger, rendering it both impractical and ineffective. Others, like CrowdGuard [30] and BaFFLe [27], are conceptually closer to a collaborative setting, as they partly rely on client-side validation and voting. However, CrowdGuard still requires a central aggregation, and sends the aggregated model to a random subset of clients, not just neighbors, and BaFFLe's quorum protocol requires server coordination. Neither can be transferred to a purely decentralized setting. Finally, some approaches rely on direct trigger reverse-engineering. FedGame [28] models defense as a minimax game, using server-side trigger reverse-engineering to compute a *genuine score*. DETRIGGER [32] reverse-engineers a candidate trigger from each update via input-layer gradient analysis and verifies it on a server-held validation set. Among these, DETRIGGER is the most amenable to decentralization, as it operates on each single update independently. We draw inspiration from it in the local detection phase of ARGUS (Section 3.2.1) but show that, when used alone, it produces unacceptable drops in accuracy under NIID data (Section 5.3).

Finally, the interaction between differential privacy (DP) and backdoor robustness has also been studied. While the approach proposed in [54] provides mixed results on robustness and utility, Du et al. [55] have shown that it can improve outlier and novelty detection, and extended this idea to backdoor-poisoning detection, but their approach is formulated for centralized learning.

Unlike FL defenses, ARGUS operates without a central server or any view of updates beyond the local neighborhood. It directly identifies attacker nodes and the backdoor mechanism, rather than relying on model-level anomaly score which may be harmed by data heterogeneity. Most importantly, ARGUS *leverages* the peer-to-peer topology of DL as a *feature* rather than treating it as a limitation: cross-validation of recovered triggers enables separating true backdoors from false alarms induced by data heterogeneity. This collaborative verification enables ARGUS to achieve near-oracle accuracy while maintaining low ASR, without requiring any trusted entity, global aggregation or knowledge of the trigger.

C Additional design details

In this appendix, we provide additional design details of ARGUS that were omitted from Section 3 for brevity. Specifically, we cover the formal definition of the trigger similarity metric (Appendix C.1) and the trust state machine (Appendix C.2).

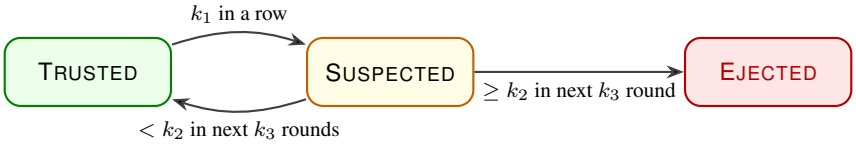

Figure 6: The per-neighbor trust state machine maintained by each (honest) node.

C.1 Formal definition of the trigger similarity metric

We give below the full definition of the trigger similarity metric introduced in Section 3.2.2. It consists of two steps:

Step 1: Top-k energy clipping. For a trigger $\hat{\tau} \in \mathbb{R}^{C\times H\times W}$, we define the per-pixel energy as the mean across channels of the trigger pixels: $E(\hat{\tau})_{hw} = \frac{1}{C}\sum_{c=1}^{C} \hat{\tau}_{chw}$. We retain at most k pixels with the highest energy, yielding the clipped trigger $\tilde{\tau}$. This ensures the similarity value depends only on the spatial arrangement of the k most influential pixels, not on trigger density which may vary across detecting nodes (this is required for the threshold calibration in Appendix D).

Step 2: SSIM comparison. The similarity metric between two triggers $\tilde{\tau}_a$ and $\tilde{\tau}_b$ is the mean *local* SSIM of the two clipped energy maps $E(\tilde{\tau}_a)$ and $E(\tilde{\tau}_b)$, as introduced by [42]:

$$\texttt{sim}(\tilde{\tau}_a, \tilde{\tau}_b) = \frac{1}{HW} \sum_{(h,w)\in H\times W} \frac{(2\mu_a\mu_b + C_1)(2\sigma_{ab} + C_2)}{(\mu_a^2 + \mu_b^2 + C_1)(\sigma_a^2 + \sigma_b^2 + C_2)}\Bigg|_{(h,w)}$$

where each term is taken over a window $w \times w$ using the uniform averaging kernel K where $K(x, y) = \frac{1}{w^2}$ for $(x, y) \in [-w/2, +w/2]^2$ and $K(x, y) = 0$ otherwise, and where:

$$\mu_x = K \star E(\tilde{\tau}_x), \quad \sigma_{ab} = K \star (E(\tilde{\tau}_a)E(\tilde{\tau}_b)) - \mu_a\mu_b, \quad \sigma_x^2 = K \star E(\tilde{\tau}_x)^2 - \mu_x^2$$

with $\star$ being the convolution operator and $C_1 = (0.01 \cdot L)^2$, $C_2 = (0.03 \cdot L)^2$ with $L = \max(\max(E(\tilde{\tau}_a)), \max(E(\tilde{\tau}_b)))$.

C.2 Specifications of the trust state machine

In ARGUS, each node maintains a per-neighbor trust state machine to reduce computations for nodes that are very likely to be malicious. The specification of this trust state machine is illustrated in Figure 6. We now explain these states and transitions in more detail.

States. Each node i maintains a *trust state* $\mathcal{S}_j \in \{\text{TRUSTED}, \text{SUSPECTED}, \text{EJECTED}\}$ for each of its neighbors $j \in N(i)$, governed by a simple state machine with 3 states:

(i) **TRUSTED:** This is the default case.

(ii) **SUSPECTED:** Model updates from a node in the SUSPECTED state are rejected regardless of the current round's decision outcome.

(iii) **EJECTED** (terminal state): Model updates originating from a node j in the EJECTED state are permanently rejected. Node i keeps j's latest collaboratively verified trigger mask in memory so that it can answer possible future COLLABORATIVEVERIFY requests. These trigger masks are very lightweight and efficient to store.

Transitions. Transitions may be governed by any rule. In experiments, we chose $(k_1, k_2, k_3) = (2, 1, 3)$, with trade-offs characterized in Proposition 4, and a numerical application in Remark 5.

Honest nodes that become backdoored. A consequence of our trust state machine is that an honest node that has been backdoored, *e.g.*, because it integrated a few malicious updates before ARGUS caught the attacker, may be forever ignored. This is by design: our experiments show that backdoors in DL are remarkably persistent (see Appendix G.1) and are present for many rounds even after attacker nodes are ejected from the network. An infected node would thus keep spreading the backdoor for many rounds.

D Calibrating the similarity threshold

We next provide the details behind the principled similarity threshold selection introduced in Section 3.3. Recall that we seek a threshold ξ such that two independent FP triggers under Insight 1 are unlikely to exceed it while TP triggers do. We aim to only use quantities known before training, *i.e.*, the image dimensions H and W, the clipping parameter k, the SSIM window size w and the model architecture.

D.1 Smooth null model

Motivation. According to Insight 1, FP triggers are caused by input-layer gradients being biased by each origin and detecting node's heterogeneous local data. These gradients are not pixel-wise independent: the structure of the first convolutional layer correlates nearby pixels. Intuitively, the gradient at each pixel aggregates contributions from all positions whose receptive field covers that pixel.

Gaussian random field model. We thus model each FP trigger energy map as a sample from an isotropic Gaussian random field on the $H \times W$ grid. Concretely:

$$\tilde{E} = G_\sigma \star Z \qquad \text{where} \qquad G_\sigma(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \tag{1}$$

with $Z \sim \mathcal{N}(0, I)$ of size $H \times W$ and $\star$ the convolution. Two FP triggers are modeled as two independent realizations of this process, after which the top-k clipping and SSIM computation from Section 3.2.2 are applied.

Choice of σ from the architecture. Based on the motivation above, σ should reflect the scale at which input-layer gradients vary, which for convolutional architectures depends on the first layer's receptive field. For our CIFAR-10 experiment, we use a ResNet-8 which begins with a 3×3 convolutional layer with stride 1, giving a 3×3 receptive field. A natural choice is thus $\sigma = 1.5$ (the receptive field half-width). However, we note that: *(i)* a larger σ produces smoother triggers, which raises ξ and reduces FPs at the cost of potentially more False Negative (FN) flaggings, but *(ii)* in ARGUS, having a few FNs is not catastrophic, since a single missed backdoor will be diluted by accepted honest updates. We thus propose to round σ up for all datasets. A more formal analysis of the relationship between architecture and the optimal σ is left for future work.

D.2 Monte Carlo calibration procedure

Given (H, W, k, w, σ), the threshold ξ is estimated from the Gaussian distribution as follows:

1. Sample M independent pairs of null triggers $(\tau_a^{(m)}, \tau_b^{(m)})$ for $m = 1, \ldots, M$ following Equation (1).
2. For each pair, apply top-k clipping and compute $\texttt{sim}(\tau_a^{(m)}, \tau_b^{(m)})$ with window size w (also see Section 3.2.2).
3. Set ξ at the desired quantile q of the resulting similarity distribution.

We use $M = 10\ 000$ and $q = 0.99$ for our experiments. This is a one-time, offline, pre-run computation that does *not* depend on any training data or the model weights, and runs in a few seconds on a single CPU.

D.3 Calibration across datasets

The clipping parameter k and window size w are set consistently across datasets:

- **Clipping parameter k.** We set $k \approx 0.05 \cdot H \cdot W$ (rounded to the nearest integer), an order of magnitude retaining the 5% of pixels with the highest energy. This is motivated by our threat model (see Section 3.1), which assumes spatially localized triggers that occupy a small fraction of the image.

Table 3: **Similarity threshold calibration across datasets.** SSIM parameters and resulting threshold ξ for each dataset, following the heuristic above. All use $\sigma = 2$ and $M = 10\ 000$ Monte Carlo samples.

Dataset	$H \times W$	**Architecture**	k	w	$\mathbb{E}[\texttt{sim}]$	std[sim]	$\xi\ (q_{99})$
CIFAR-10	32×32	ResNet-8	51	11	0.195	0.081	0.42
FEMNIST	28×28	CNN	39	9	0.279	0.089	0.51
TinyImageNet	64×64	ResNet-18	205	19	0.058	0.031	0.15

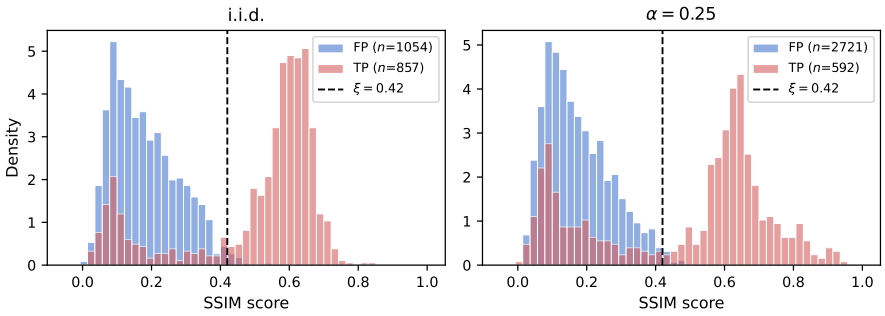

Figure 7: **Empirical FP vs. TP similarities (additional settings).** We consider IID (left) and $\alpha = 0.25$ (right) heterogeneity levels, using the CIFAR-10 dataset.

- **Window size** w**.** We set $w \approx H/3$, so that SSIM windows are large enough to capture meaningful spatial structure while remaining local.

Table 3 reports the resulting per-dataset parameters and threshold.

Several points are worth noting. First, the null SSIM distributions vary significantly across image resolutions, confirming the need for a per-dataset thresholding rather than a universal constant. Larger images (as in TinyImageNet) lead to much lower null similarities since two Gaussian random fields are less likely to align, σ staying the same. Second, the null distributions are relatively well concentrated, which validates the idea to discriminate based on similarities. Third, the thresholds ξ all sit well below 1 (perfect similarity), leaving a large margin for TP triggers.

Empirical validation. Figure 4 in Section 3.2.2 already shows the FP/TP trigger similarity distributions for $\alpha = 0.5$ on the CIFAR-10 dataset. We additionally show in Figure 7 the same distributions for additional heterogeneity levels: IID (left) and $\alpha = 0.25$ (right). Across all data distributions, $\xi = 0.42$ cleanly separates the two distributions. Lower TP SSIM values correspond to early iterations, where the backdoor is yet to be fully implanted, as well as occasional trigger reversal errors.[3] Appendix G.7 explores the effectiveness of ARGUS when the threshold varies.

E Additional experimental setup details

In this appendix, we provide additional details on the experimental setup given in Section 5.1.

Training hyperparameters. On the CIFAR-10 and FEMNIST datasets, nodes train for 160 and 80 rounds, respectively, with the SGD optimizer and learning rate $\eta = 0.01$. Each round with CIFAR-10 and FEMNIST consists of 5 and 15 local batches, respectively, of 64 images using the cross-entropy loss. On the TinyImageNet dataset, we train for 160 rounds with learning rate $\eta = 0.1$, momentum 0.9 and weight decay 10^{-4}. Each round with TinyImageNet consists of 60 batches of 64 images.

ARGUS hyperparameters. Because FPs will be filtered by collaborative verification, we set the local-detection threshold aggressively low to $\gamma = 0.5$ (also see Section 3.2.1). The number of

[3] Note that the distributions in the figure are normalized, which may exaggerate the perceived TP trigger counts.

refinement steps is 5 for CIFAR-10 and FEMNIST and 3 for TinyImageNet, with a step size of 0.2 in both cases. Each node's validation set is made of 1 sample per class, for classes available in the local dataset. The similarity threshold ξ and other collaborative verification parameters are calibrated per dataset as described in Appendix D.3 and the minimum number of confirmations is $\kappa = 1$, unless specified otherwise. The TRUSTED $\rightarrow$ SUSPECTED transition is triggered after $k_1 = 2$ consecutive rejections, and SUSPECTED $\rightarrow$ EJECTED after $k_2 = 1$ rejection in the following $k_3 = 3$ rounds.

Attack configuration. Attacker nodes poison a fixed fraction $p_{\text{poison}} = 0.3$ ($p_{\text{poison}} = 0.25$ for TinyImageNet) of their training set, by adding the appropriate backdoor trigger and changing the target label to $t^* = 7$. They do not incorporate incoming peer updates during averaging [11]. During collaborative verification, they send a NOT-SUSPICIOUS message to every query about another attacker node, and a SUSPICIOUS message with a random trigger of size k for queries about honest nodes.

Seeds and reproducibility. Each configuration in Table 1 and Figure 5 is run with random seeds $\{1, 2, 3\}$. We report mean and standard deviation. All our code and run scripts are released as a public repository (see Section 5.1).

Hardware and implementation. We run all experiments on our compute cluster. Nodes in this cluster are equipped with an NVIDIA A100 GPU with 40GB of VRAM. We implement ARGUS using the DECENTRALIZEPY framework [56].

F Baseline implementation details

We now describe the baselines introduced in Section 5.1 in more detail. All of them are applied locally at each node, since no central coordinator is available in our DL setting. When possible, we reuse the hyperparameters from the respective papers, otherwise we tune them.

NO DEFENSE. This baseline represents standard decentralized parallel stochastic gradient descent (D-PSGD) in which each node trains locally, broadcasts its update, and averages all incoming updates with uniform weights. There is no defense against any backdoor attacks.

ORACLE. This baseline represents the ground-truth defense, where honest nodes reject all updates from $\mathcal{M}$ and accept all updates from $\mathcal{H}$. This defense is not practical to deploy since it assumes the identities of attacker nodes are known in advance by all honest nodes, but it provides a lower bound for ASR and an upper bound for CA.[4]

MULTI-KRUM [15]. This is a Byzantine-robust aggregation rule and is commonly used as a baseline in related work on backdoor defenses [17, 29, 25]. We adapt this rule to DL by running it independently over the model updates received by each node from its neighbors. More precisely, given received updates $\{\theta_j\}_{j \in N(i)}$, each node i computes score $s_j^{(i)} = \sum_{k \in A_j} \|\theta_k - \theta_j\|^2$, where A_j is the set of $|N(i)| - r - 1$ closest updates and r is the number of updates to reject. The r updates with the largest score are rejected, while the rest is averaged with node i's own model. We set r to be the maximum number of attackers in the neighborhood of any node (*e.g.*, $r = 1$ in the experiments in Section 5).

BADFL [37]. This baseline corrects model weights after each local training round. Given post-aggregation snapshots X_{t-1} and X_t, and post-training weights $\tilde{X}_t$, it forms the diagonal Hessian estimate $\hat{H} = ((\tilde{X}_{t+1} - X_t) - (X_t - X_{t-1}))/\gamma$, clips $(I - \gamma\hat{H})$ element-wise to $[-q, q]$ to obtain G and corrects its local update: $X_t = \tilde{X}_t - \gamma\alpha G$. We adopt the hyperparameters from their work and set $\alpha = 0.08$ and $q = 0.1$.

P2PCD [10]. With this baseline, at round t, node i rescales each incoming neighbor update $U_j^t = \theta_j^t - \theta_j^{t-1}$ to satisfy $\|U_j^t\|_2 \leq C_{\text{neigh}}$ and applies the same rescaling to its own local update U_i^t with bound C_{local}, where $C_{\text{neigh}} < C_{\text{local}}$, before aggregating. Clipping is disabled during an initial

[4] Although, due to heterogeneity, this CA could possibly be exceeded while keeping a low ASR by accepting very few malicious model updates.

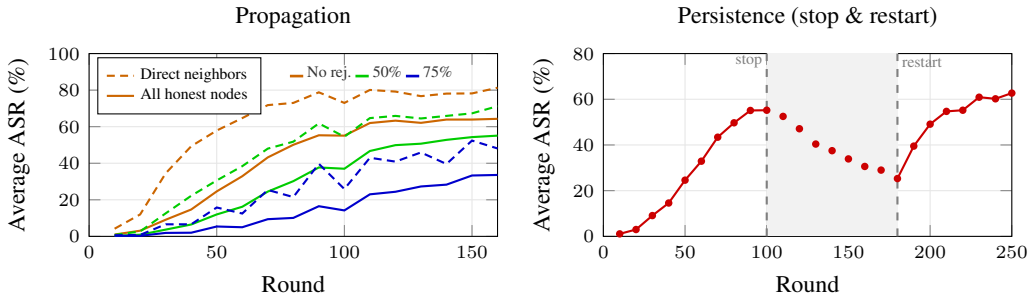

Figure 8: **Backdoor propagation and persistence in DL.** Experiments on CIFAR-10 with a 3-regular network of 16 nodes ($\alpha = 0.5$) and $m = 1$ attacker node. *Left*: Average ASR of honest nodes with varying malicious updates rejection rates, and when rejecting no malicious updates. The solid line indicates the average ASR across all honest nodes whereas the dashed line considers the ASR of nodes that are directly connected to the attacker node. *Right*: The average ASR of honest nodes where the attacker poisons during rounds 0–100, then goes silent (shaded), and resumes its malicious behavior at round 180.

agreement phase to improve accuracy. We adopt $C_{\text{neigh}} = 0.1$ and $C_{\text{local}} = 1.0$ as reported by the authors. We also adopt their choice of 50 agreement rounds for the CIFAR-10 and TinyImageNet datasets, but set it to 0 on FEMNIST since we found non-zero values allow the backdoor to propagate through the agreement phase and persist.

G Additional experimental results

We provide additional experiments that complement the main results presented in Section 5. These additional experiments address the following questions:

(Q1) How severe and persistent is the backdoor threat in DL, even from a single attacker node?

(Q2) Does ARGUS remain effective in the IID setting, where data heterogeneity is not a confounding factor?

(Q3) How effective is ARGUS when varying the backdoor trigger shape, size, and position?

(Q4) Can attacker nodes evade ARGUS by breaking the cross-node consistency of recovered triggers?

(Q5) How does ARGUS perform under different network topologies and attacker counts?

(Q6) Does ARGUS degrade gracefully when the connectivity assumption stated in Section 3.1 is violated?

(Q7) How sensitive is ARGUS to the choice of similarity threshold ξ?

(Q8) What is the computational and communication overhead introduced by ARGUS, and when is it acceptable?

G.1 Backdoor propagation and persistence in DL

To motivate the need for a defense against the backdoor attack in DL, we run an experiment to quantify the disruption caused by a single attacker node. We train an ML model using a 3-regular, 16-node network with heterogeneity $\alpha = 0.5$ on CIFAR-10, where a single node (node 0) injects a backdoor in their model updates. We consider a setting where honest nodes do not use any defense to detect or prevent backdoors, and aggregate all incoming model updates (*no rejection*). Additionally, to study the effectiveness of rejecting backdoored model updates, we also consider two settings where malicious model updates are rejected in a structured, periodic manner: with a 50% and 75% rejection rate, respectively.

Propagation. Figure 8 (left) shows the average ASR across honest nodes as training progresses when different proportions of malicious updates are rejected throughout, and without any rejection. With *no rejection*, the backdoor by the single attacker node propagates rapidly through the network,

Table 4: **CIFAR-10 IID results.** The Clean Test Accuracy (CA) (↑ is better) and Attack Success Rate (ASR) (↓ is better) with $m = 2$ attacker nodes out of $n = 16$ nodes, reporting mean and std over 3 seeds. We use the CIFAR-10 dataset in an IID setting ($\alpha = \infty$).

Defense	CA [%]	ASR [%]
ORACLE	66.6±0.4	0.0±0.0
NO DEFENSE	57.7±0.6	70.6±0.3
BADFL	53.0±0.8	52.7±1.3
P2PCD	58.8±0.8	32.9±0.3
MULTI-KRUM	65.0±0.2	0.1±0.0
ARGUS (Ours)	65.0±0.4	1.6±0.1

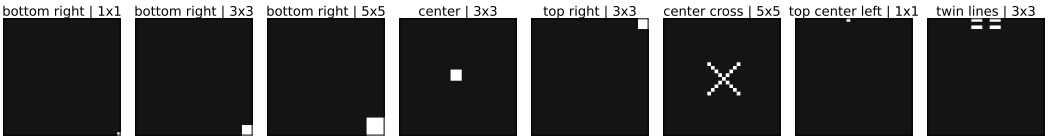

Figure 9: **Various trigger types.** We vary their shape, position and size. Actual triggers are colored in gray to be more inconspicuous.

reaching over 50% average ASR for all nodes within the first 80 rounds. Separating direct neighbors of the attacker node (dashed lines) from the whole network (solid) shows the backdoor propagation dynamics: direct neighbors are infected first and reach above 80% ASR. The other nodes in the network are infected with a delay but still reach an average ASR around 60%. Even when honest nodes can somehow identify and reject 50% or 75% of the malicious updates, the backdoor still propagates and is effective, with the average ASR for all nodes reaching 55% and 34% on average, respectively. Thus, Figure 8 (left) demonstrates that even when rejecting a majority of backdoored model updates, the backdoor is able to spread and manifest in the network.

Persistence. Figure 8 (right) shows the average ASR of honest nodes when the attacker stops injecting and spreading its backdoor at round 100 and later resumes at round 180. One would expect that the backdoor effectively gets overridden and erased through the model updates of honest nodes. However, even after the attacker goes silent, the ASR only decays slowly, decreasing by just 30 percentage points in 80 rounds and rebounds rapidly once poisoning resumes. This demonstrates that backdoors in DL are remarkably persistent: once injected, they remain in the network for many rounds, even when the attacker stops spreading malicious model updates.

G.2 The effectiveness of ARGUS and baselines on CIFAR-10 (IID)

Table 4 shows the CA and ASR on the CIFAR-10 dataset in the IID setting ($\alpha = \infty$). These results are obtained using the same setting as the experiment described in Section 5.2. Table 4 shows that both ARGUS and MULTI-KRUM achieve near-oracle performance in this setting. We attribute the effectiveness of MULTI-KRUM to the fact that attackers do not integrate model updates from their neighboring nodes to make their backdoor as effective as possible. This makes their model updates diverge geometrically from those of honest nodes, whose model updates remain close to one another under an IID data distribution. Consequently, this gives MULTI-KRUM a clean distance-based signal to identify and reject the model updates from attackers. As shown earlier in Table 1, the effectiveness of MULTI-KRUM in NIID settings degrades since the model updates from honest nodes are further apart, making it increasingly difficult to distinguish attacker from honest nodes as data heterogeneity increases. Nevertheless, Table 4 shows that ARGUS compared to MULTI-KRUM achieves similar CA and competitive ASR in an IID setting, demonstrating that our trigger-based detection does not meaningfully degrade CA even in this regime that structurally favors distance-based methods. Notably, BADFL and P2PCD fail to suppress the backdoor even under IID conditions, with average ASR of 52.7% and 32.9% respectively, and incur more pronounced degradations in CA compared to the ORACLE baseline.

Table 5: **Robustness to different trigger patterns**. The Clean Test Accuracy (CA) (↑ is better), Attack Success Rate (ASR) (↓ is better) and False Positive Rate (FPR) (↓ is better) with $m = 2$ attacker nodes out of $n = 16$ nodes. We use the CIFAR-10 ($\alpha = 0.5$) and FEMNIST datasets. Triggers vary in position, size and shape and are visualized in Figure 9.

	FEMNIST			CIFAR-10 ($\alpha = 0.5$)		
Trigger (position, size)	CA [%]	ASR [%]	FPR [%]	CA [%]	ASR [%]	FPR [%]
bottom right, 1×1	72.8	0.0	34.6	53.4	1.8	0.7
bottom right, 3×3	73.0	0.0	26.8	54.2	2.9	2.4
bottom right, 5×5	72.5	3.7	31.9	54.9	3.6	1.2
center, 3×3	72.9	0.5	29.4	55.4	10.3	1.0
top right, 3×3	72.9	0.1	31.7	52.8	6.0	1.0
center cross, 5×5	72.8	0.1	39.5	55.1	13.1	0.9
top center left, 1×1	72.9	0.0	29.0	54.6	6.8	0.7
twin lines, 3×3	72.9	0.0	29.1	55.1	12.1	0.7

Table 6: **Robustness to a moving trigger.** The Clean Test Accuracy (CA) (↑ is better), Attack Success Rate (ASR) (↓ is better), False Positive Rate (FPR) (↓ is better) and True Positive Rate (TPR) (↑ is better) when attacker nodes re-position their 3×3 trigger uniformly at random within a 16×16 region at every round. We use a 3-regular graph with $n = 16$ nodes, $m = 2$ attacker nodes and $\kappa = 1$, on the CIFAR-10 ($\alpha = 0.5$) dataset.

Defense	CA [%]	ASR [%]	FPR [%]	TPR [%]
ORACLE	55.7	0.5	0.0	100
NO DEFENSE	45.0	49.7	–	–
ARGUS	54.0	1.7	1.2	72.9

G.3 Robustness of ARGUS to different trigger patterns

In Section 5 we analyzed the effectiveness of ARGUS using a single, fixed trigger. To see if ARGUS remains effective with different trigger characteristics, we now evaluate ARGUS on eight triggers with varying size, position and shape, including a non-contiguous pattern (*twin lines*). This is consistent with the variety of triggers considered in [32]. We visualize these triggers in Figure 9.

Table 5 reports final CA, ASR and FPR for ARGUS on the FEMNIST and CIFAR-10 ($\alpha = 0.5$) datasets, with $m = 2$ attacker nodes in a 3-regular graph with $n = 16$ nodes. ARGUS effectively mitigates the backdoor attack in all configurations while keeping a CA close to the ORACLE baseline (see Table 1) regardless of the trigger. The ASR is consistently low for both datasets and across tested triggers. The FPR is clearly higher on FEMNIST than on CIFAR-10, where it is close to 0%, even if the accuracy remains high. This is consistent with the higher null SSIM predicted by our analysis in Appendix D. Triggers located in the center of the image (*e.g.*, *center* and *center cross*) lead to a higher ASR on CIFAR-10 compared to other trigger variations, but are still effectively detected, with a TPR of 85.0% and 99.3% respectively.

G.4 Robustness of ARGUS to adaptive attackers

All experiments in this paper already consider attacker nodes that are adaptive at the protocol level: they know each other, the communication graph and the ARGUS algorithm, and actively interfere with collaborative verification by shielding fellow attacker nodes and framing honest ones (Section 3.2.2 and Appendix E). We now discuss how attacker nodes could instead target the property on which collaborative verification relies, namely the consistency of the triggers recovered by different neighbors of the same node (Insight 1).

We first note that breaking this consistency is not trivial: the consistency ARGUS tests is not on something directly transmitted by the attacker, but a property of the received model, which each neighbor derives independently from its own data. Making two honest neighbors recover inconsistent

Table 7: **Robustness to different topologies and number of attacker nodes.** The Clean Test Accuracy (CA) (↑ is better), Attack Success Rate (ASR) (↓ is better), False Positive Rate (FPR) (↓ is better) and True Positive Rate (TPR) (↑ is better) with varying topologies, number of attackers m, total number of nodes n and confirmation thresholds κ. We use the CIFAR-10 ($\alpha = 0.5$) dataset. The backdoor trigger is a 3×3 square at the bottom-right corner.

Topology	CA [%]	ASR [%]	FPR [%]	TPR [%]
3-regular, $m = 2, n = 16, \kappa = 1$	54.2	2.9	2.4	82.8
Fully-connected, $m = 4, n = 8, \kappa = 3$	52.8	26.0	0.0	16.4
Erdős–Rényi-Gilbert, $m = 3, n = 32, \kappa = 1$	51.8	0.6	0.3	99.7

triggers from the *same* model requires a weak or ambiguous backdoor, which in turn reduces the effectiveness of the attack. This leaves three natural strategies:

(i) Using a distinct trigger per attacker node. This does not help, since collaborative verification is performed per sender: all the neighbors of a given attacker node inspect the same model, and thus the same trigger.

(ii) Sending different models to different neighbors, each carrying a different trigger. This can be detected at negligible cost by attaching a hash of the model received to each verification query and reply: mismatching hashes reveal that the sender did not send the same model to all its neighbors and thus either a faulty or malicious behavior.

(iii) Changing the trigger across rounds, to disperse the energy of the recovered trigger and harm reverse-engineering, but this weakens the backdoor.

We evaluate strategy *(iii)* with a *moving trigger*: at each iteration, attacker nodes re-position their 3×3 trigger uniformly at random within a 16×16 region. Table 6 reports the results on CIFAR-10 ($\alpha = 0.5$) with $m = 2$ attacker nodes in our default 3-regular graph of $n = 16$ nodes. We make two observations. First, the evasion strategy weakens the attack itself: without defense, the moving trigger reaches an ASR of 49.7%, compared to 79.0% for the static one (Table 11). Second, ARGUS still defends against it, with an ASR of 1.7% and a CA within 1.7 points of ORACLE. Reverse-engineering still recovers a localized trigger that neighbors confirm, with a TPR of 72.9% compared to 82.8% for the static trigger (Table 7).

G.5 Robustness of ARGUS to different topologies and numbers of attacker nodes

To further understand the robustness of ARGUS, we vary both the type of the communication topology and the number of attackers. We consider three topologies: our default 3-regular graph with $m = 2$, an 8-node fully connected graph with $m = 4$ (run for 250 rounds since only 4 honest nodes contribute), and an Erdős–Rényi-Gilbert random graph of $n = 32$ nodes including $m = 3$ malicious ones, where the probability of an edge being present is set to $p = 0.103$ so that the expected degree matches the 3-regular graph.[5] We also vary the number of required peer confirmations κ for the fully-connected graph, setting it to 3. Table 7 reports final CA, ASR, FPR and TPR on the CIFAR-10 dataset ($\alpha = 0.5$). In general, ARGUS remains effective across topologies and attacker counts. However, we observe an ASR of 26% with the fully-connected graph, which we explain by the following two factors. First, the attacker ratio is substantially higher (50% compared to 12.5% in the default setting), meaning honest nodes absorb proportionally more malicious updates before ejection. Second, the higher confirmation threshold $\kappa = 3$, which is necessary to maintain correctness guarantees under this attacker ratio, reduces the TPR to 16.4%. This difficulty in detecting actual triggers slows down the ejection of attacker nodes. However, training for additional rounds after ejection of the attacker nodes is likely to reduce the ASR.

Scaling the number of attacker nodes. In Table 7, the topology and the number of attacker nodes vary jointly. To isolate the effect of the latter, we fix our default 3-regular graph with $n = 16$ nodes and $\kappa = 1$, and increase the number of attacker nodes from $m = 1$ to $m = 5$. Table 8 reports the results on CIFAR-10 ($\alpha = 0.5$). Detection remains effective across all configurations, with an ASR

[5]The expected degree is $(n-1)p$, we set it to 3 to allow $\kappa = 1$.

Table 8: **Scaling the number of attacker nodes.** The Clean Test Accuracy (CA) (↑ is better) of ORACLE and ARGUS, and the Attack Success Rate (ASR) (↓ is better), False Positive Rate (FPR) (↓ is better) and True Positive Rate (TPR) (↑ is better) of ARGUS, with m attacker nodes (nodes 0 to $m-1$) in a 3-regular graph with $n = 16$ nodes and $\kappa = 1$. We use the CIFAR-10 ($\alpha = 0.5$) dataset. All runs use the same seed.

	ORACLE	ARGUS			
m	CA [%]	CA [%]	ASR [%]	FPR [%]	TPR [%]
1	56.4	55.3	0.7	3.1	83.8
2	55.7	54.2	2.9	2.4	82.8
3	55.9	52.4	6.5	4.6	83.3
4	53.9	49.5	6.4	3.1	83.6
5	52.8	51.1	6.1	0.5	80.8

Table 9: **Scaling the network size and sparsity.** The Clean Test Accuracy (CA) (↑ is better), Attack Success Rate (ASR) (↓ is better) and False Positive Rate (FPR) (↓ is better) of ARGUS with n nodes, m attacker nodes (keeping the attacker ratio constant) and $\kappa = 1$. We use the CIFAR-10 ($\alpha = 0.5$) dataset. On the ring, attacker nodes are placed apart so that each honest node keeps an honest witness.

Topology	n	m	CA [%]	ASR [%]	FPR [%]
3-regular	16	2	54.2	2.9	2.4
3-regular	32	4	53.4	1.6	0.1
Ring	16	2	49.4	2.1	0.8
Ring	32	4	44.2	3.5	0.0

of at most 6.5% (compared to 79.0% without defense for $m = 2$, see Table 11) and stable FPR and TPR. The mild decrease in CA as m grows is independent of ARGUS and also observed for ORACLE: each additional attacker node replaces an honest contributor, which lowers the accuracy achievable by any defense, especially with NIID data.

Larger and sparser networks. We also double the network size to $n = 32$ nodes for both the 3-regular graph and the ring topology (the sparsest connected topology, with degree 2), keeping the attacker ratio fixed ($m = 2$ for $n = 16$ and $m = 4$ for $n = 32$). On the ring, we place attacker nodes apart so that each honest node keeps an honest witness for its malicious neighbors (we study adjacent attacker nodes in Appendix G.6). Table 9 shows that the ASR stays low as the network grows and becomes sparser, at most 3.5%, with an FPR of at most 2.4%. The lower CA on the ring stems from its slower mixing, which slows down DL convergence independently of the defense. Such sparse topologies are particularly unfavorable to distance-based defenses like MULTI-KRUM, which must estimate outlier statistics from only two received updates per round.

G.6 The performance of ARGUS when violating the connectivity assumption

We assume in Section 3.1 that each node has at least $\kappa + 1$ honest neighbors. When this fails for a particular node, ARGUS may be unable to collect enough confirmations to confirm a true backdoor during the collaborative verification phase. We evaluate the impact on ARGUS' efficiency if this assumption is violated. Specifically, we consider a $n = 16$ node setting, connected with a 3-regular graph and $m = 4$ attackers. We additionally set $\kappa = 1$. Table 10 shows the per-node and averaged CA and ASR in a configuration where node 3 is the only honest neighbor of attacker node 1. As expected, node 3 is significantly more affected by the backdoor than its peers. The ASR of node 3 reaches 15.8%, which is the highest among all honest nodes and strongly above the average ASR of 5.4%. Interestingly, even though node 3 is significantly more affected by the backdoor than its peers, its honest neighbors are not in turn corrupted by it: the collaborative verification mechanism allows them to identify and reject node 3's compromised updates, effectively containing the spread of the backdoor and preventing it from propagating further into the network. Even in this pessimistic setting, ARGUS still effectively protects the rest of the network (the average ASR among honest nodes is much lower than the 72.4% with NO DEFENSE in Figure 5), since the remaining nodes still have

Table 10: **Connectivity assumption violations on CIFAR-10.** The Clean Test Accuracy (CA) (↑ is better) and Attack Success Rate (ASR) (↓ is better) on the CIFAR-10 dataset with $\alpha = 0.5$. We use a 3-regular graph with $n = 16$ nodes, $\kappa = 1$ and $m = 4$ attackers (nodes {0, 1, 6, 8}). Node 3 is the only honest node whose malicious neighbor has no other honest witnesses (Node 1's other neighbors are malicious too).

Honest node	2	3	4	5	7	9	10	11	12	13	14	15	**Avg.**
CA [%]	50.6	46.8	51.0	50.4	50.9	28.9	53.6	53.7	49.1	45.8	43.6	55.3	48.3 ± 6.7
ASR [%]	1.1	**15.8**	5.5	2.5	5.8	0.0	1.7	3.2	11.0	3.2	7.5	7.7	5.4 ± 4.4

Table 11: **Varying the similarity threshold ξ.** The Clean Test Accuracy (CA) (↑ is better), Attack Success Rate (ASR) (↓ is better) and rejection rate (↓ is better) using a 3-regular graph, $n = 16$ nodes, $m = 2$ attackers, and $\kappa = 1$. We use the CIFAR-10 ($\alpha = 0.5$) dataset. The calibrated value of ξ is $\xi^* = 0.42$ (see Appendix D.3).

Similarity threshold ξ	CA [%]	ASR [%]	Rej. Rate of all model updates [%]
0 (LOCAL DEF.)	45.3	1.1	75.7
0.27	44.1	1.5	37.7
0.42	54.2	2.9	13.3
0.57	55.3	5.7	11.4
0.72	50.9	47.7	8.4
1 (NO DEFENSE)	45.0	79.0	0.0

enough honest neighbors to collectively identify malicious updates. Even on node 3, the backdoor remains far weaker than without defense, since the malicious update is averaged with those of its two honest neighbors. Degradation under violation of the assumption is thus graceful and localized.

Worst case on a ring. We push this setting further with the sparsest connected topology: a ring of $n = 16$ nodes with $\kappa = 1$ on CIFAR-10 ($\alpha = 0.5$), where the $m = 2$ attacker nodes (nodes 0 and 1) are adjacent. Nodes 2 and 15 therefore have no honest witness for their malicious neighbor. Both these nodes reach an ASR of 48.5%, while the remaining honest nodes average at 5.7%: only the nodes that cannot verify are affected, and the rest of the network stays protected. The higher ASR of nodes 2 and 15 compared to node 3 above is expected, since on a ring the malicious update is averaged with a single honest update instead of two. Finally, we recall that κ is a design parameter that should be set from the graph degree and the expected fraction of attacker nodes (Section 3.3), and may even vary across honest nodes.

G.7 The effect of varying the similarity threshold

To quantify the effect of the similarity threshold ξ on the effectiveness of ARGUS, we vary ξ beyond the calibrated value $\xi^* = 0.42$ (see Appendix D.3). We consider a setting with a 3-regular graph, $n = 16$ nodes, $m = 2$ attackers, and $\kappa = 1$. We use the CIFAR-10 dataset with heterogeneity $\alpha = 0.5$. Table 11 shows the CA, ASR and rejection rate (Rej. Rate) of *all* model updates in this setting for different similarity thresholds ξ. We remark that $\xi = 0$ corresponds to the LOCAL DEF. baseline while $\xi = 1$ is equivalent to the NO DEFENSE baseline. Increasing the value of ξ beyond the calibrated value decreases CA and increases ASR as fewer malicious model updates will be rejected. On the other hand, a lower value of ξ also decreases CA since an increasing number of honest updates are rejected, but also reduces ASR. We also observed in our experiments that a small offset from the calibrated value of ξ does not significantly impact the CA nor the ASR, therefore providing some flexibility to ARGUS users to estimate ξ. Overall, these results suggest that the calibrated threshold ξ sits in a wide basin of near-optimal performance, making ARGUS robust to moderate miscalculations of ξ in practice.

G.8 Computation and communication overhead of ARGUS

Finally, we quantify the computation and communication overhead introduced by ARGUS. We adopt the same setting as used in the experiments described in Section 5.2 and consider $m = 2$ attacker

Table 12: **Computation and communication overhead of ARGUS.** Wall-clock training time (↓ is better) and maximum bytes sent per round (↓ is better) averaged over honest nodes and 3 seeds for NO DEFENSE and ARGUS. We report the relative slowdown with regard to NO DEFENSE in parentheses.

	Bytes per round [MB]			Time per round [s]		
Defense	CIFAR-10	FEMNIST	TinyImageNet	CIFAR-10	FEMNIST	TinyImageNet
NO DEFENSE	59.25	20.28	135.37	4.18 (1.0×)	21.1 (1.0×)	32.5 (1.0×)
ARGUS	59.32	20.28	135.38	22.5 (5.4×)	77.4 (3.7×)	291.6 (9.0×)

nodes in a network with $n = 16$ nodes, connected using a 3-regular graph. Table 12 reports the wall-clock time in seconds and the bytes exchanged per training round in MB for each dataset, with and without ARGUS. All times and communication volumes are averaged across iterations over honest nodes and over three seeds. For CIFAR-10, we also compute the mean values over all considered heterogeneity levels ($\alpha \in \{0.25, 0.5, \infty\}$).

Table 12 shows that communication overhead induced by ARGUS is negligible, since each neighbor confirmation requires the exchange of small trigger masks. The additional communication volume introduced by exchanging these masks is negligible compared to the size of exchanged models. The computational overhead is, however, more pronounced and is primarily caused by the local trigger reverse-engineering step, where each honest node tries to reconstruct a trigger for each target class. Since ARGUS has to optimize trigger masks for each possible target class, the computational cost grows with the number of classes. Table 12 shows that the increase in time per round is between 3.7× (for FEMNIST) and 9.0× (for TinyImageNet). We remark that the TinyImageNet dataset contains 200 classes, compared to 10 and 62 classes for the CIFAR-10 and FEMNIST datasets, respectively. Furthermore, larger sizes of input images also result in additional computation per optimization step.

Cost structure. Local trigger reverse-engineering accounts for more than 90% of the overhead on TinyImageNet. At each round, for each non-ejected neighbor and each of the K candidate target classes, an honest node computes an initial candidate and performs R refinement steps, each on its small validation set $\mathcal{D}_i^{val}$ (at most one sample per class, see Appendix E). The cost per round thus amounts to roughly $|N(i)| \cdot K \cdot R$ gradient steps over $\mathcal{D}_i^{val}$. It grows with the number of classes and the node degree, but neither with the network size n nor with the size of local datasets, two quantities that grow in real deployments. It also decreases over the course of training, since ejected neighbors are skipped entirely and attacker nodes are ejected with high probability (Proposition 4). Finally, our setup accentuates the relative overhead: nodes deliberately perform little local work per round (*e.g.*, 5 batches of 64 samples on CIFAR-10), whereas the detection cost does not depend on the local workload, so its share decreases as local workloads grow.

Reducing the overhead. We identify two promising avenues to reduce this computational cost. First, since trigger recovery is performed independently per target class, these computations can be trivially parallelized, with each class assigned to a separate thread or device. Second, we can significantly reduce the computational cost by exploiting the structure of realistic attack scenarios. In practice, an attacker is unlikely to switch target classes mid-training, as doing so would require re-injecting an entirely new backdoor into the network from scratch, which is a costly and slow process. ARGUS can exploit this temporal consistency: once a target class has been flagged as suspicious in a given round, subsequent rounds can concentrate their optimization budget on that class, while only spot-checking remaining classes with a reduced number of gradient steps or at a lower sampling rate. This yields a significant reduction in per-round compute cost in the presence of an attacker, with little risk of missing a class switch. On TinyImageNet ($K = 200$), replacing the full scan over all classes by roughly one class could reduce the dominant cost term by up to two orders of magnitude. In the benign case where no attacker is present, we can apply a complementary strategy: more compute budget can be spent at the beginning of training, when the model is still evolving rapidly and a backdoor could more easily take hold, and gradually reduced as training progresses and the model stabilizes. Together, these two strategies form an adaptive compute schedule that naturally allocates more resources where and when they are most needed. We leave a thorough investigation of these optimizations and other efficiency improvements to future work.

H Proof of Section 4.2

H.1 Notations

Recall that the averaging rule of Algorithm 1 can be written as

$$\theta_i^{t+1} = \frac{1}{|\mathcal{A}_i|+1}\left(\theta_i^{t+1/2} + \sum_{j\in N(i)} \delta_{i,j}^t \theta_j^{t+1/2}\right).$$

Equivalently, in matrix form, let

$$\theta^t := \begin{bmatrix} (\theta_1^t)^\top \\ \vdots \\ (\theta_n^t)^\top \end{bmatrix} \in \mathbb{R}^{n\times d}, \qquad G^t := \begin{bmatrix} (g_1^t)^\top \\ \vdots \\ (g_n^t)^\top \end{bmatrix} \in \mathbb{R}^{n\times d},$$

where g_i^t is the stochastic gradient at node i and round t.

$$\theta^{t+1/2} = \theta^t - \eta \nabla \mathcal{L}(\theta^t), \qquad \theta^{t+1} = S^t \theta^{t+1/2},$$

where S^t is the random row-stochastic mixing matrix induced by the accepted links at round t (and matches Stage 2 of Algorithm 1, see Section 4). We use

$$\Pi := I - \frac{1}{n} 1 1^\top, \qquad \widetilde{\theta}^t := \Pi \theta^t = \theta^t - 1\,\overline{\theta}^t, \qquad \overline{\theta}^t := \frac{1}{n} 1^\top \theta^t \in \mathbb{R}^d.$$

In words, $\widetilde{\theta}^t$ is the *disagreement (centered) iterate*: its i-th row is $\theta_i^t - \overline{\theta}^t$, hence $\|\widetilde{\theta}^t\|_F^2 = \sum_{i=1}^n \|\theta_i^t - \overline{\theta}^t\|^2$.

H.2 Auxiliary lemmas

We state and prove here useful lemmas that will be used throughout the proofs.

Lemma 9. *For all $x \in \mathbb{R}^n$, we have*

$$\frac{1}{n}\sum_{i=1}^n \|x_i - \bar{x}\|^2 = \frac{1}{2n^2}\sum_{i,j=1}^n \|x_i - x_j\|^2, \tag{2}$$

Proof. We have

$$\begin{aligned}
\frac{1}{2n^2}\sum_{i,j=1}^n \|x_i - x_j\|^2 =& \frac{1}{2n^2}\sum_{i,j=1}^n \|(x_i - \bar{x}) - (x_j - \bar{x})\|^2 \\
=& \frac{1}{2n^2}\sum_{i,j=1}^n \|x_i - \bar{x}\|^2 + \|x_j - \bar{x}\|^2 - 2\langle x_i - \bar{x}, x_j - \bar{x}\rangle \\
=& \frac{1}{n}\sum_{i=1}^n \|x_i - \bar{x}\|^2,
\end{aligned}$$

where the last equality uses

$$\sum_{i,j=1}^n \langle x_i - \bar{x}, x_j - \bar{x}\rangle = \sum_{j=1}^n \langle \sum_{i=1}^n x_i - \bar{x}, x_j - \bar{x}\rangle = \sum_{j=1}^n \langle 0, x_j - \bar{x}\rangle = 0$$

. □

Moreover, we provide a characterization of how the link failures affect the expected gossip matrix $\mathbb{E}[S^t]$:

Lemma 10 (Expected weights of the random mixing matrix)**.** *Let* $d_i = |\mathcal{N}(i)|$ *and* $M_i^t = \sum_{k\in\mathcal{N}(i)} \delta_{i,k}^t$*. Under Assumption 2, the entries of* S^t *are given by:*

$$S_{i,i}^t = \frac{1}{1+M_i^t}, \qquad S_{i,j}^t = \frac{\delta_{i,j}^t}{1+M_i^t} \text{ for } j \in \mathcal{N}(i),$$

and

$$q_i := \mathbb{E}\left[S_{i,i}^t\right] = \mathbb{E}\left[\frac{1}{1+M_i^t}\right] = \frac{1-{p_{\text{fp}}}^{d_i+1}}{(d_i+1)(1-p_{\text{fp}})}.$$

Moreover,

$$\bar{S}_{i,j} := \mathbb{E}[S_{i,j}^t] = \begin{cases} q_i & \text{if } j = i, \\ \frac{1-q_i}{d_i} & \text{if } j \in \mathcal{N}(i), \\ 0 & \text{otherwise}. \end{cases}$$

In particular, $\bar{S}$ *is row-stochastic. Moreover, if* $\mathcal{W}$ *is* d*-regular (*$\forall i,j \in \mathcal{V}, d_i = d_j = d$*), then* $\bar{S}$ *is symmetric and doubly-stochastic.*

Proof of Lemma 10. The identities for $S_{i,i}^t$ and $S_{i,j}^t$ follow directly from row normalization and $\delta_{i,i}^t = 1$.

We now focus on the expectation derivations. Denote $p = 1 - p_{\text{fp}}$. Since $M_i^t \sim \text{Binomial}(d_i, p)$,

$$\mathbb{E}\left[S_{i,i}^t\right] = \mathbb{E}\left[\frac{1}{1+M_i^t}\right] = \sum_{m=0}^{d_i} \frac{1}{m+1}\binom{d_i}{m} p^m (1-p)^{d_i-m} = \frac{1-(1-p)^{d_i+1}}{(d_i+1)p}.$$

All off-diagonal expectations on $\mathcal{N}(i)$ sum to $1-q_i$: $\sum_{j\in\mathcal{N}(i)} \mathbb{E}[S_{i,j}^t] = 1-q_i$. By symmetry across neighbors, all off-diagonal expectations on $\mathcal{N}(i)$ are equal, hence $\mathbb{E}[S_{i,j}^t] = \frac{1-q_i}{d_i}$ for $j \in \mathcal{N}(i)$. □

H.3 Proof of Proposition 6

We now restate and prove Proposition 6:

Proposition 6 (Spectral gap of $\mathbb{E}\big((S^t)^\top S^t\big)$)**.** *For an* undirected *and* d*-regular communication graph* $\mathcal{W}$*, if the entries of* S^t *satisfy Assumption 2, let* $\phi_{p_{\text{fp}}}(\nu) := (a - cd) + 2b\,\nu + c\,\nu^2$ *and* $\psi_{p_{\text{fp}}}(\mu) := \phi_{p_{\text{fp}}}((d+1)\mu - 1)$ *for some constants* a, b, c *(*cf.*, Equation* (4) *to* (6)*) depending on* p_{fp}*. Let* $1 = \mu_1 \geq \mu_2 \geq \cdots \geq \mu_n$ *be the eigenvalues of* $\mathcal{W}$*. Then, we obtain the following identities:*

$$\lambda_1\big(\mathbb{E}[(S^t)^\top S^t]\big) = 1 \quad \textit{and} \quad \rho := \lambda_2\big(\mathbb{E}[(S^t)^\top S^t]\big) = \max\big\{\psi_{p_{\text{fp}}}(\mu_2)\,,\ \psi_{p_{\text{fp}}}(\mu_n)\big\} \in [0,1).$$

Proof of Proposition 6. **Proof for the largest eigenvalue:** We have:

$$\mathbb{E}[(S^t)^\top S^t]1 = \mathbb{E}[(S^t)^\top S^t 1] = \mathbb{E}[(S^t)^\top 1] = \mathbb{E}[(S^t)^\top]1 = \mathbb{E}[(S^t)]^\top 1 = \mathbb{E}[(S^t)]1 = 1 \tag{3}$$

where we used that $\mathbb{E}[S^t]$ is symmetric when $\mathcal{W}$ is d-regular thanks to Lemma 10. This completes the proof for the largest eigenvalue.

Proof for the second largest eigenvalue For simplicity, denote $q = (1 - p_{\text{fp}})$, and let us write:

$$a := \mathbb{E}\left[\frac{1}{1+M}\right], \quad M \sim \text{Binomial}(d, 1-p_{\text{fp}}), \tag{4}$$

$$b := (1-p_{\text{fp}})\ \mathbb{E}\left[\frac{1}{(2+U)^2}\right], \quad U \sim \text{Binomial}(d-1, 1-p_{\text{fp}}), \tag{5}$$

$$c := (1-p_{\text{fp}})^2\ \mathbb{E}\left[\frac{1}{(3+V)^2}\right], \quad V \sim \text{Binomial}(d-2, 1-p_{\text{fp}}) \quad (c := 0 \text{ if } d = 1). \tag{6}$$

Let

$$Y := \mathbb{E}\left[(S^t)^\top S^t\right], \qquad Y_{j,k} = \sum_{i=1}^n \mathbb{E}\left[S_{i,j}^t S_{i,k}^t\right].$$

Write $K_i := 1 + M_i^t$ the number of effective neighbors for node i at step t. For each row i, nonzero entries of S^t are $S_{i,i}^t = 1/K_i$ and $S_{i,j}^t = \delta_{i,j}^t/K_i$ for $j \in \mathcal{N}(i)$.

For $j \neq k$, a term $S_{i,j}^t S_{i,k}^t$ is nonzero only if both columns j, k appear in row i. This gives three cases:

1. $i = j$ and $k \in \mathcal{N}(j)$: contribution $\mathbb{E}[\delta_{j,k}^t/K_j^2] = b$;
2. $i = k$ and $j \in \mathcal{N}(k)$: contribution $\mathbb{E}[\delta_{k,j}^t/K_k^2] = b$, and is symmetric to the previous case;
3. $i \notin \{j, k\}$ and $j, k \in \mathcal{N}(i)$: contribution $\mathbb{E}[\delta_{i,j}^t \delta_{i,k}^t/K_i^2] = c$.

Hence, for $j \neq k$,

$$Y_{j,k} = b\,1_{\{k\in\mathcal{N}(j)\}} + b\,1_{\{j\in\mathcal{N}(k)\}} + c\,\#\{i:\ j,k \in \mathcal{N}(i)\}. \tag{7}$$

Since $\mathcal{W}$ represents a d-regular undirected graph, we can write A to be its adjacency matrix with $A_{ii} = 0$ for all i, and we have:

$$\mathcal{W} = \frac{1}{d+1}(I + A). \tag{8}$$

For an undirected d-regular graph with adjacency matrix A following Equation (8), Equation (7) becomes:

$$Y_{j,k} = 2b\,A_{j,k} + c\,(A^2)_{j,k}, \qquad j \neq k.$$

For the diagonal terms, we have:

$$Y_{j,j} = \mathbb{E}\left[\frac{1}{K_j^2}\right] + \sum_{i\in\mathcal{N}(j)} \mathbb{E}\left[\frac{\delta_{i,j}^t}{K_i^2}\right].$$

By d-regularity and exchangeability across neighbors (symmetry), we have that for any fixed $\ell \in \mathcal{N}(j)$:

$$\sum_{i\in\mathcal{N}(j)} \mathbb{E}\left[\frac{\delta_{i,j}^t}{K_i^2}\right] = d\,\mathbb{E}\left[\frac{\delta_{j,\ell}^t}{K_j^2}\right] = \mathbb{E}\left[\frac{M_j^t}{K_j^2}\right].$$

Thus, we have:

$$Y_{j,j} = \mathbb{E}\left[\frac{1 + M_j^t}{K_j^2}\right] = \mathbb{E}\left[\frac{1}{1 + M_j^t}\right] = a.$$

Since $(A^2)_{j,j} = d$, the compact matrix identity is

$$Y = (a - cd)I + 2bA + cA^2$$

Now, following Equation (8), the matrices $\mathcal{W}$ and A share eigenvectors. Using the eigenvalues $\mu_1, \ldots, \mu_n$ defined above, define

$$\nu_k := (d+1)\mu_k - 1,$$

so that ν_k is the corresponding eigenvalue of A. Because $\mathcal{W}$ is symmetric (undirected gossip matrix), it has an orthonormal eigenbasis $\{v_k\}_{k=1}^n$, which is also an eigenbasis of A. For each k:

$$Y v_k = \big((a - cd) + 2b\nu_k + c\nu_k^2\big)v_k = \phi_{p_{\mathrm{fp}}}(\nu_k)\, v_k.$$

So the eigenvalues of Y are exactly $\{\phi_{p_{\mathrm{fp}}}(\nu_k)\}_{k=1}^n$. Using $\nu_k = (d+1)\mu_k - 1$, define

$$\psi_{p_{\mathrm{fp}}}(\mu) := \phi_{p_{\mathrm{fp}}}((d+1)\mu - 1)\,.$$

Then equivalently the eigenvalues of Y are $\{\psi_{p_{\mathrm{fp}}}(\mu_k)\}_{k=1}^n$.

Finally, for $k \geq 2$, we have $\mu_n \leq \mu_k \leq \mu_2$. Also, $\psi''_{p_{\mathrm{fp}}}(\mu) = 2c(d+1)^2 \geq 0$, so $\psi_{p_{\mathrm{fp}}}$ is convex on $[\mu_n, \mu_2]$. Therefore, for each $k \geq 2$,

$$\psi_{p_{\mathrm{fp}}}(\mu_k) \leq \max\big\{\psi_{p_{\mathrm{fp}}}(\mu_2)\,,\ \psi_{p_{\mathrm{fp}}}(\mu_n)\big\}\,.$$

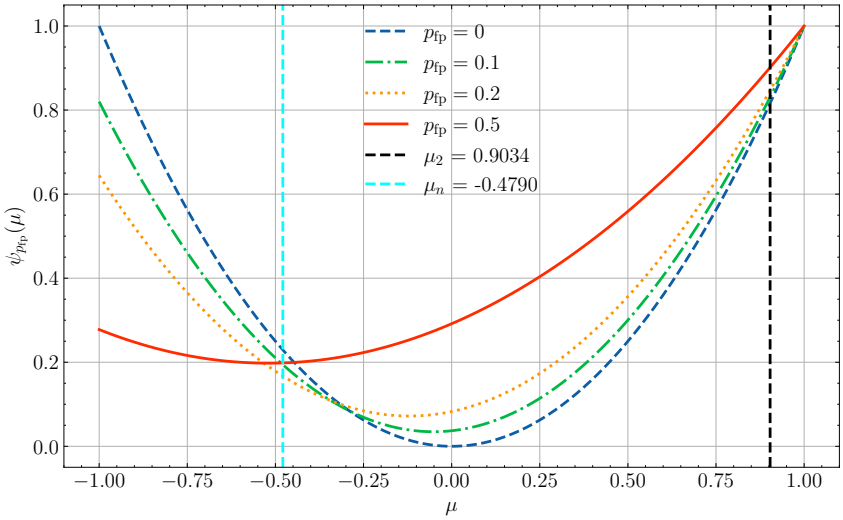

Figure 10: Evolution of the function $\psi_{p_{\text{fp}}}$ for different values of p_{fp}. We represent the behavior on the eigenvalues (μ_2, μ_n) for a given 3-regular graph with 16 nodes.

Thus, we get:

$$\max_{k\geq 2} \psi_{p_{\text{fp}}}(\mu_k) = \max\{\psi_{p_{\text{fp}}}(\mu_2)\,,\ \psi_{p_{\text{fp}}}(\mu_n)\}\,.$$

Moreover, since $\mu_1 = 1$, for every k:

$$\psi_{p_{\text{fp}}}(1) - \psi_{p_{\text{fp}}}(\mu_k) = \phi_{p_{\text{fp}}}(d) - \phi_{p_{\text{fp}}}(\nu_k) = (d-\nu_k)\big(2b + c(d+\nu_k)\big) \geq 0,$$

because $\nu_k \in [-d, d]$ for a d-regular undirected adjacency matrix and $b, c \geq 0$. Hence $\lambda_1(Y) = \psi_{p_{\text{fp}}}(1)$ and therefore

$$\lambda_2(Y) = \max_{k\geq 2} \psi_{p_{\text{fp}}}(\mu_k)\,.$$

Combining the two identities,

$$\lambda_2\big(\mathbb{E}[(S^t)^\top S^t]\big) = \lambda_2(Y) = \max\{\psi_{p_{\text{fp}}}(\mu_2)\,,\ \psi_{p_{\text{fp}}}(\mu_n)\}\,.$$

□

In addition to Proposition 6, we provide in Figure 10 an analysis of the function $\psi_{p_{\text{fp}}}$ as p_{fp} varies, which controls the gossip factor of the random mixing matrix. We observe that as p_{fp} increases, $\psi_{p_{\text{fp}}}$ tends to favor positive eigenvalues.

H.4 A mean-square contraction lemma

We now connect Proposition 6 to the standard *spectral gap* parameter used in DL analysis. Note that the relevant quantity is the second moment $\mathbb{E}\big[(S^t)^\top S^t\big]$ [57].

Lemma 11 (Mean-square consensus contraction)**.** *Assume the setting of Proposition 6 and define*

$$\rho := \lambda_2\big(\mathbb{E}\big[(S^t)^\top S^t\big]\big) \in [0, 1).$$

Then, for any matrix $X \in \mathbb{R}^{n\times d}$ satisfying $1^\top X = 0$, we have

$$\mathbb{E}\big\|S^t X\big\|_F^2 \leq \rho\, \|X\|_F^2.$$

In particular,

$$\mathbb{E}\big\|\widetilde{\theta}^{t+1}\big\|_F^2 = \mathbb{E}\big\|\Pi\theta^{t+1}\big\|_F^2 \leq \rho\, \mathbb{E}\big\|\widetilde{\theta}^{t+1/2}\big\|_F^2.$$

Proof. Let $M := \mathbb{E}\big[(S^t)^\top S^t\big]$. For any $X \in \mathbb{R}^{n\times d}$,

$$\mathbb{E}\|S^t X\|_F^2 = \mathbb{E}\,\mathrm{tr}\big(X^\top (S^t)^\top S^t X\big) = \mathrm{tr}\big(X^\top M X\big).$$

Since $1^\top X = 0$, every column of X lies in the subspace orthogonal to 1. By Proposition 6, M is symmetric with eigenvectors given by those of $\mathcal{W}$, and the largest eigenvalue of M on the subspace orthogonal to 1 is precisely ρ. Therefore, for each column x of X, we have $x^\top M x \le \rho \|x\|^2$, and summing over columns yields the claim.

Finally, using $\theta^{t+1} = S^t \theta^{t+1/2}$ and the identity $\widetilde{\theta}^{t+1/2} = \Pi\theta^{t+1/2}$ with $1^\top \widetilde{\theta}^{t+1/2} = 0$, the last inequality follows by applying the first bound to $X = \widetilde{\theta}^{t+1/2}$. □

H.5 Proof of Theorem 7

We adapt the convergence analysis of Epidemic Learning (EL) [3] under our scenario. The main differences are that we consider a fixed underlying d-regular topology and not a randomized d-regular topology at each round. We will thus perform the same assumption as EL, namely L-smoothness, bounded variance, and bounded heterogeneity.

The only difference in the proof lies in the mixing lemmas [3, Lemma 1], where the contraction factor of the mixing step is controlled by a value defined by the authors. We revisit this lemma with constants adapted to our setting:

Lemma 12 (Contraction Factor for ARGUS)**.** *Consider Algorithm 1. Let $n \ge 2, d \ge 1, T \ge 1$ and $t \in [|0, T-1|]$. Assume the underlying communication graph is d-regular and undirected, and the filtering masks are independent of the stochastic gradient noise at round t. Let $\rho := \lambda_2\left(\mathbb{E}\left[(S^t)^\top S^t\right]\right) \in [0,1)$. Then,*

(a) $\mathbb{E}\left[\overline{\theta}^{t+1}\right] = \mathbb{E}\left[\overline{\theta}^{t+1/2}\right]$

(b) $\frac{1}{n}\sum_{i\in\mathcal{V}} \mathbb{E}\left[\left\|\theta_i^{t+1} - \overline{\theta}^{t+1}\right\|^2\right] \le \frac{\rho}{n}\sum_{i\in\mathcal{V}} \mathbb{E}\left[\left\|\theta_i^{t+1/2} - \overline{\theta}^{t+1/2}\right\|^2\right]$

(c) $\mathbb{E}\left[\left\|\overline{\theta}^{t+1} - \overline{\theta}^{t+1/2}\right\|^2\right] \le \frac{\beta(d,p_{\mathrm{fp}})}{n}\frac{1}{n}\sum_{i\in\mathcal{V}} \mathbb{E}\left[\left\|\theta_i^{t+1/2} - \overline{\theta}^{t+1/2}\right\|^2\right]$

with

$$\beta(d, p_{\mathrm{fp}}) = \frac{1}{n}\mathbb{E}\left\|(S^t)^\top 1 - 1\right\|^2 \tag{9}$$

Proof of Lemma 12. Let $J := \frac{1}{n}11^\top$ and $\Pi := I - J$ denote respectively the averaging operator and the orthogonal projection onto the disagreement subspace. We denote $\overline{\theta}^t = J\theta^t$.

(a) Since link dropout is symmetric across nodes, $\mathbb{E}[S^t]$ is a symmetric row-stochastic matrix, and thus it is doubly stochastic. Therefore,

$$\mathbb{E}[\overline{\theta}^{t+1}] = \mathbb{E}\left[\frac{1}{n}1^\top S^t\theta^{t+1/2}\right] = \mathbb{E}\left[\frac{1}{n}1^\top\mathbb{E}[S^t]\theta^{t+1/2}\right] = \mathbb{E}[\overline{\theta}^{t+1/2}].$$

(b) $\Pi\theta$ represents the disagreement. Using $\Pi S^t J = 0$ and $\Pi J = 0$, we get:

$$\Pi\theta^{t+1} = \Pi S^t\theta^{t+1/2} = \Pi S^t \Pi\theta^{t+1/2}.$$

Since $1^\top\Pi\theta^{t+1/2} = 0$, we can apply Lemma 11 with $X = \Pi\theta^{t+1/2}$ to obtain

$$\mathbb{E}\left\|\Pi\theta^{t+1}\right\|_F^2 = \mathbb{E}\left\|\Pi S^t\Pi\theta^{t+1/2}\right\|_F^2 \le \mathbb{E}\left\|S^tX\right\|_F^2 \le \rho\,\mathbb{E}\|X\|_F^2 = \rho\,\mathbb{E}\left\|\Pi\theta^{t+1/2}\right\|_F^2.$$

Rewriting $\|\Pi\theta\|_F^2 = \sum_{i=1}^n \left\|\theta_i - \overline{\theta}\right\|^2$ yields the claim.

(c) We start from

$$\overline{\theta}^{t+1} - \overline{\theta}^{t+1/2} = \frac{1}{n}1^\top(S^t - I)\theta^{t+1/2} = \frac{1}{n}1^\top(S^t - I)\Pi\theta^{t+1/2},$$

since $(S^t - I)1 = 0$. Therefore,

$$\overline{\theta}^{t+1} - \overline{\theta}^{t+1/2} = \frac{1}{n}\left((S^t)^\top 1 - 1\right)^\top\Pi\theta^{t+1/2}.$$

By Cauchy–Schwarz,

$$\left\|\overline{\theta}^{t+1} - \overline{\theta}^{t+1/2}\right\|^2 \leq \frac{1}{n^2}\left\|(S^t)^\top 1 - 1\right\|^2 \cdot \left\|\Pi\theta^{t+1/2}\right\|_F^2.$$

Using $\left\|\Pi\theta^{t+1/2}\right\|_F^2 = \sum_{i\in\mathcal{V}}\left\|\theta_i^{t+1/2} - \overline{\theta}^{t+1/2}\right\|^2$ and taking expectation,

$$\mathbb{E}\left\|\overline{\theta}^{t+1} - \overline{\theta}^{t+1/2}\right\|^2 \leq \underbrace{\frac{1}{n}\mathbb{E}\left\|(S^t)^\top 1 - 1\right\|^2}_{=:\beta(d,p_{\mathrm{fp}})} \cdot \frac{1}{n}\mathbb{E}\sum_{i\in\mathcal{V}}\left\|\theta_i^{t+1/2} - \overline{\theta}^{t+1/2}\right\|^2.$$

where we used the fact that the filtering masks S^t are independent of the gradients computed in the same round.

□

Moreover, the uniform bound on the model drift and gradient drift remains the same as existing literature [3, Lemma 2], with the mixing factor β_s replaced by ρ in our setting, as the proof only relies on contraction properties of the mixing step and not on the average drift.

Lemma 13 (Uniform bound on model and gradient drift)**.** *Consider a stepsize $\eta \leq \frac{1}{20L}$ and identical initial models. For any $t \geq 0$, we have:*

$$\frac{1}{n}\sum_{i\in\mathcal{V}}\mathbb{E}\left[\left\|\theta_i^t - \overline{\theta}^t\right\|^2\right] \leq 40\frac{1+3\rho}{(1-\rho)^2}\rho\eta^2(\varsigma^2+\sigma^2), \tag{10}$$

and

$$\frac{1}{n^2}\sum_{i,j\in\mathcal{V}}\mathbb{E}\left[\left\|g_i^t - g_j^t\right\|^2\right] \leq 15(\varsigma^2+\sigma^2). \tag{11}$$

Proof. We unroll the proof of De Vos et al. [3, Lemma 2], with β_s replaced by ρ following Lemma 12. This is the only part of the proof of the lemma where the communication graph comes into play, ensuring that a similar bound holds in our setting. □

Finally, the rest of the proof of Theorem 7 follows by combining the descent lemma with the bounds on the average drift and gradient drift, in a similar fashion as De Vos et al. [3, Theorem 1].

Theorem 7 (Convergence rate of ARGUS)**.** *Consider Assumption 2, $G = 1$ and that all attackers have been ejected while honest neighbors remain* TRUSTED*. Moreover, assume that the setting of Proposition 6 holds, and that mixing matrices S^t are independent of the gradients computed in the same round. Assume each local objective $\mathcal{L}_i$ is L-smooth and that SGD steps satisfy $\forall x \in \mathbb{R}^d, t \in \mathbb{N} : \mathbb{E}\left\|g_i^t(x) - \nabla\mathcal{L}_i(x)\right\|^2 \leq \sigma^2$ and $\frac{1}{n}\sum_{i=1}^n\left\|\nabla\mathcal{L}_i(x) - \nabla\mathcal{L}(x)\right\|^2 \leq \varsigma^2$. Define $\Delta_0 := \mathcal{L}\left(\overline{\theta}^0\right) - \mathcal{L}^*$. Then, for a constant stepsize η (Equation (12)), and constants C_ρ and $\beta(d,p_{\mathrm{fp}})$, $\frac{1}{n}\sum_{i\in\mathcal{V}}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\left\|\nabla\mathcal{L}(\theta_i^t)\right\|^2\right]$ is bounded by:*

$$\mathcal{O}\left(\frac{L}{T}\Delta_0 + \sqrt{\frac{L\Delta_0 C_\rho\beta(d,p_{\mathrm{fp}})(\varsigma^2+\sigma^2)}{nT}} + \sqrt[3]{\frac{L^2\Delta_0^2 C_\rho(\varsigma^2+\sigma^2)}{T^2}}\right).$$

For clarity, we write here the value of the stepsize η before proceeding with the proof:

$$\eta := \min\left\{\frac{1}{20L}, \sqrt{\frac{4n\,\Delta_0}{T\,L\,(4\sigma^2 + (60+320C_\rho)\,\beta(d,p_{\mathrm{fp}})(\varsigma^2+\sigma^2))}}, \left(\frac{\Delta_0}{T\,160L^2C_\rho(\varsigma^2+\sigma^2)}\right)^{\frac{1}{3}}\right\}. \tag{12}$$

Proof of Theorem 7. We derive a similar proof to EL [3]. Using the usual convergence lemma, we derive [3, Equation 6]:

$$\frac{1}{n}\sum_{i\in\mathcal{V}}\mathbb{E}\left[\left\|\nabla\mathcal{L}(\theta_i^t)\right\|^2\right] \leq \frac{4}{\eta}\mathbb{E}\left[\mathcal{L}(\overline{\theta}^t) - \mathcal{L}(\overline{\theta}^{t+1})\right] + \frac{4L^2}{n}\sum_{i\in\mathcal{V}}\mathbb{E}\left[\left\|\theta_i^t - \overline{\theta}^t\right\|^2\right]$$

$$+ 4L\eta\frac{\sigma^2}{n} + \frac{4L}{\eta}\mathbb{E}\left[\left\|\overline{\theta}^{t+1} - \overline{\theta}^{t+1/2}\right\|^2\right]. \tag{13}$$

We can now bound terms individually: using Lemma 12, we have:

$$\begin{aligned}
\mathbb{E}\left[\left\|\overline{\theta}^{t+1} - \overline{\theta}^{t+1/2}\right\|^2\right] &\leq \frac{\beta(d,p_{\text{fp}})}{n}\frac{1}{n}\sum_{i\in\mathcal{V}}\mathbb{E}\left[\left\|\theta_i^{t+1/2} - \overline{\theta}^{t+1/2}\right\|^2\right] \\
&\overset{(2)}{=} \frac{\beta(d,p_{\text{fp}})}{2n}\frac{1}{n^2}\sum_{i,j\in\mathcal{V}}\mathbb{E}\left[\left\|\theta_i^{t+1/2} - \theta_j^{t+1/2}\right\|^2\right] \\
&= \frac{\beta(d,p_{\text{fp}})}{2n}\frac{1}{n^2}\sum_{i,j\in\mathcal{V}}\mathbb{E}\left[\left\|\theta_i^{t} - \theta_j^{t} - \eta(g_i^t - g_j^t)\right\|^2\right] \\
&\leq \frac{\beta(d,p_{\text{fp}})}{n^3}\sum_{i,j\in\mathcal{V}}\mathbb{E}\left[\left\|\theta_i^{t} - \theta_j^{t}\right\|^2\right] + \frac{\beta(d,p_{\text{fp}})\eta^2}{n^3}\sum_{i,j\in\mathcal{V}}\mathbb{E}\left[\left\|g_i^t - g_j^t\right\|^2\right] \\
&\overset{(2)}{=} \frac{2\beta(d,p_{\text{fp}})}{n^2}\sum_{i\in\mathcal{V}}\mathbb{E}\left[\left\|\theta_i^{t} - \overline{\theta}^{t}\right\|^2\right] + \frac{\beta(d,p_{\text{fp}})\eta^2}{n^3}\sum_{i,j\in\mathcal{V}}\mathbb{E}\left[\left\|g_i^t - g_j^t\right\|^2\right].
\end{aligned}$$

We can now plug this into Equation (13) to get:

$$\begin{aligned}
\frac{1}{n}\sum_{i\in\mathcal{V}}\mathbb{E}\left[\left\|\nabla\mathcal{L}(\theta_i^t)\right\|^2\right] \leq& \frac{4}{\eta}\mathbb{E}\left[\mathcal{L}(\overline{\theta}^t) - \mathcal{L}(\overline{\theta}^{t+1})\right] + \left(4L^2 + \frac{8L\beta(d,p_{\text{fp}})}{n\eta}\right)\frac{1}{n}\sum_{i\in\mathcal{V}}\mathbb{E}\left[\left\|\theta_i^t - \overline{\theta}^t\right\|^2\right] \\
&+ 4L\eta\frac{\sigma^2}{n} + \frac{4L\beta(d,p_{\text{fp}})\eta}{n}\frac{1}{n^2}\sum_{i,j\in\mathcal{V}}\mathbb{E}\left[\left\|g_i^t - g_j^t\right\|^2\right] \\
\overset{(a)}{\leq}& \frac{4}{\eta}\mathbb{E}\left[\mathcal{L}(\overline{\theta}^t) - \mathcal{L}(\overline{\theta}^{t+1})\right] + 4L\eta\frac{\sigma^2}{n} + \frac{4L\beta(d,p_{\text{fp}})\eta}{n}15(\varsigma^2+\sigma^2) \\
&+ \left(4L^2 + \frac{8L\beta(d,p_{\text{fp}})}{n\eta}\right)\left(40\frac{1+3\rho}{(1-\rho)^2}\rho\eta^2(\varsigma^2+\sigma^2)\right)
\end{aligned} \tag{14}$$

where (a) follows from Lemma 13. Denoting

$$\Delta_0 := \left[\mathcal{L}(\overline{\theta}^0) - \mathcal{L}^*\right] \tag{15}$$

and

$$C_\rho := \frac{1+3\rho}{(1-\rho)^2}\rho, \tag{16}$$

and summing over $t = 0$ to $T-1$, we get:

$$\begin{aligned}
\frac{1}{nT}\sum_{i\in\mathcal{V}}\sum_{t=0}^{T-1}\mathbb{E}\left[\left\|\nabla\mathcal{L}(\theta_i^t)\right\|^2\right] \leq& \frac{4}{T\eta}\Delta_0 + 4L\eta\frac{\sigma^2}{n} + \frac{4L\beta(d,p_{\text{fp}})\eta}{n}15(\varsigma^2+\sigma^2) \\
&+ \left(4L^2 + \frac{8L\beta(d,p_{\text{fp}})}{n\eta}\right)\left(40C_\rho\eta^2(\varsigma^2+\sigma^2)\right) \\
=& \frac{4}{T\eta}\Delta_0 + 4L\sigma^2\frac{\eta}{n} + 60L\beta(d,p_{\text{fp}})\frac{\eta}{n}(\varsigma^2+\sigma^2) \\
&+ \left(160L^2C_\rho\eta^2 + 320L\beta(d,p_{\text{fp}})C_\rho\frac{\eta}{n}\right)(\varsigma^2+\sigma^2) \\
=& \frac{4}{T\eta}\Delta_0 + \eta^2\left(160L^2C_\rho\right)(\varsigma^2+\sigma^2) \\
&+ \frac{\eta}{n}\left(4L\sigma^2 + (60+320C_\rho)L\beta(d,p_{\text{fp}})(\varsigma^2+\sigma^2)\right) \\
=& \frac{4}{T\eta}\Delta_0 + \eta^2\left(160L^2C_\rho\right)(\varsigma^2+\sigma^2)
\end{aligned}$$

$$+ \frac{\eta}{n} L \left(4\sigma^2 + (60 + 320 C_\rho)\, \beta(d, p_{\text{fp}})(\varsigma^2 + \sigma^2)\right) \tag{17}$$

We now select the stepsize η as in Equation (12). Plugging this into Equation (17), we absorb constants and use that $\frac{1}{\eta}$ is bounded by the sum of the inverse of each of the three terms in the minimum above, yielding that $\frac{1}{nT} \sum_{i \in \mathcal{V}} \sum_{t=0}^{T-1} \mathbb{E}\left[\|\nabla \mathcal{L}(\theta_i^t)\|^2\right]$ is bounded by:

$$\mathcal{O}\left(\frac{L}{T}\Delta_0 + \sqrt{\frac{L \Delta_0 C_\rho \beta(d, p_{\text{fp}})(\varsigma^2 + \sigma^2)}{nT}} + \sqrt[3]{\frac{L^2 \Delta_0^2 C_\rho (\varsigma^2 + \sigma^2)}{T^2}}\right). \tag{18}$$

□

I Ejection probability bounds

We prove Proposition 4 by first deriving a generic ejection bound in Theorem 14 and then instantiating it for malicious and honest nodes in Appendix I.2.

Theorem 14 (Ejection probability bounds)**.** *Fix $T \geq 1$ and integers $k_1 \geq 1$, $1 \leq k_2 \leq k_3$. Let $Z_1, \ldots, Z_T$ be i.i.d. Bernoulli(p) rejection indicators. Assume the state rule is: enter* SUSPECTED *after k_1 consecutive detections, then become* EJECTED *if at least k_2 detections occur in the next k_3 rounds; otherwise, the node returns to* TRUSTED*.*

Define

$$A_T := \mathbb{I}_{\{\textit{the node is } \text{EJECTED } \textit{by the end of round } T\}},$$

$$\pi(p) := p^{k_1} \sum_{r=k_2}^{k_3} \binom{k_3}{r} p^r (1-p)^{k_3 - r},$$

$$\hat{\pi}(p) := (1-p)\,\pi(p), \qquad M_T := \left\lfloor \frac{T}{1 + k_1 + k_3} \right\rfloor, \qquad N_T := \max\{T - k_1 - k_2 + 1, 0\}.$$

Then

$$1 - (1 - \hat{\pi}(p))^{M_T} \leq \mathbb{P}(A_T = 1) \leq N_T\, \pi(p),$$

where the lower bound is stated for $k_2 = 1$, the setting of our experiments (Appendix C.2); Proposition 16 gives a lower bound valid for all thresholds.

I.1 Proof of Theorem 14

Proof of Theorem 14. We prove both upper and lower bounds separately.

Upper bound: For the upper bound, if $N_T = 0$ there is no full window of length $k_1 + k_3$ and the claim is immediate. Assume $N_T > 0$. For each $i \in \{1, \ldots, N_T\}$, define

$$C_i := \left\{ Z_i = \cdots = Z_{i+k_1-1} = 1, \sum_{j=i+k_1}^{i+k_1+k_3-1} Z_j \geq k_2 \right\}.$$

Any ejection by round T must be triggered by such a window, so

$$\{A_T = 1\} \subseteq \bigcup_{i=1}^{N_T} C_i.$$

Hence, by the union bound,

$$\mathbb{P}(A_T = 1) \leq \sum_{i=1}^{N_T} \mathbb{P}(C_i) = N_T\, \pi(p),$$

where the last equality uses i.i.d. Bernoulli(p) indicators.

Lower bound: We partition time into disjoint blocks of length $k' = 1 + k_1 + k_3$. For each $\ell \in \{0, \dots, M_T - 1\}$, define

$$E_\ell := \left\{ Z_{\ell k'+1} = 0,\ Z_{\ell k'+2} = \cdots = Z_{\ell k'+1+k_1} = 1,\ \sum_{j=\ell k'+2+k_1}^{(\ell+1)k'} Z_j \geq k_2 \right\}.$$

Let $k_2 = 1$. We claim that E_ℓ implies ejection by the end of block ℓ, whatever the state of the node at the start of the block. If the node is SUSPECTED when the first of the k_1 consecutive detections occurs, this detection ejects it. Otherwise it is TRUSTED at that time, with no pending run of detections thanks to the initial zero (an evaluation window can only close on a round without detection). It then enters SUSPECTED after the k_1 detections, and the first detection among the next k_3 rounds ejects it. For $k_2 \geq 2$ this argument no longer applies, as detections of block ℓ may fall in an evaluation window opened in the previous block and be discarded when that window closes; this is why the lower bound is restricted to $k_2 = 1$. Hence,

$$\bigcup_{\ell=0}^{M_T-1} E_\ell \subseteq \{A_T = 1\}.$$

The events E_ℓ are independent (they depend on disjoint time blocks), and

$$\mathbb{P}(E_\ell) = \hat{\pi}(p) \qquad \text{for all } \ell.$$

Therefore,

$$\mathbb{P}(A_T = 1) \geq \mathbb{P}\left(\bigcup_{\ell=0}^{M_T-1} E_\ell \right) = 1 - (1 - \hat{\pi}(p))^{M_T}.$$

□

I.2 Proof of Proposition 4

Proposition 4. *In any round T, for thresholds $k_1, k_2 \leq k_3$, let $\pi(p) := p^{k_1} \sum_{r=k_2}^{k_3} \binom{k_3}{r} p^r (1-p)^{k_3-r}$, and define A_T^{hon} (resp. B_T^{mal}) as the event where an honest node reaches (resp. an attacker node does not reach) the state* EJECTED *by the end of round T. With Assumption 2 and $q_{\text{fn}} := 1 - p_{\text{fn}}$, we have*

$$\mathbb{P}(B_T^{\text{mal}} = 1) \leq (1 - p_{\text{fn}}\pi(q_{\text{fn}}))^{\lfloor T/(k_1+k_3+1) \rfloor}; \mathbb{P}(A_T^{\text{hon}} = 1) \leq \max\{T - k_1 - k_2 + 1, 0\}\, \pi(p_{\text{fp}}).$$

The first bound is stated for $k_2 = 1$, as in our experiments; Appendix I.3 covers all thresholds.

Proof of Proposition 4. Under Assumption 2, the per-round rejection indicators are IID Bernoulli, with

$$1 - \delta_{i,j}^t \sim \text{Bernoulli}(1 - p_{\text{fn}}) \ \ (j \in \mathcal{M}), \qquad 1 - \delta_{i,j}^t \sim \text{Bernoulli}(p_{\text{fp}}) \ \ (j \in \mathcal{H}).$$

Let

$$A_T^{\text{mal}} := \mathbb{I}_{\{\text{an attacker node is EJECTED by the end of round } T\}}, \qquad B_T^{\text{mal}} := 1 - A_T^{\text{mal}}.$$

Applying the lower bound in Theorem 14 with $p = 1 - p_{\text{fn}}$ gives

$$\mathbb{P}(A_T^{\text{mal}} = 1) \geq 1 - (1 - p_{\text{fn}}\pi(1 - p_{\text{fn}}))^{\lfloor T/(k_1+k_3+1) \rfloor},$$

hence

$$\mathbb{P}(B_T^{\text{mal}} = 1) \leq (1 - p_{\text{fn}}\pi(1 - p_{\text{fn}}))^{\lfloor T/(k_1+k_3+1) \rfloor}.$$

For the second bound, let:

$$A_T^{\text{hon}} := \mathbb{I}_{\{\text{an honest node is EJECTED by the end of round } T\}}.$$

Applying the upper bound in Theorem 14 with $p = p_{\text{fp}}$ gives

$$\mathbb{P}(A_T^{\text{hon}} = 1) \leq \max\{T - k_1 - k_2 + 1, 0\}\, \pi(p_{\text{fp}}).$$

□

I.3 Extension: correlated detections via a two-state Markov chain

Assumption 2 treats the per-round rejection indicators as independent across rounds whereas, as noted in Section 4, model rejections are temporally correlated in practice. We relax independence to a stationary first-order Markov chain, which adds a single correlation parameter per edge type, and extend Proposition 4 to this setting. For $i \in \mathcal{H}$ and $j \in N(i)$, write $Z_{i,j}^t$ for the detection indicator at round t, *i.e.*, the outcome of collaborative verification that drives the trust state machine. In the model of Assumption 2, $Z_{i,j}^t = 1 - \delta_{i,j}^t$.

Assumption 15 (Markov two-rate detection model)**.** *For each edge (i,j), $(Z_{i,j}^t)_{t\geq 0}$ is a stationary first-order Markov chain on $\{0,1\}$ with marginal $\mathbb{P}(Z_{i,j}^t = 1) = p \in (0,1)$ ($p = p_{\mathrm{fp}}$ for $j \in \mathcal{H}$, $p = 1 - p_{\mathrm{fn}}$ for $j \in \mathcal{M}$) and transition probabilities*

$$a := \mathbb{P}(Z_{i,j}^t = 1 \mid Z_{i,j}^{t-1} = 1) = p + (1-p)\varrho, \qquad b := \mathbb{P}(Z_{i,j}^t = 1 \mid Z_{i,j}^{t-1} = 0) = p(1-\varrho),$$

where

$$\varrho = a - b = \mathrm{Corr}(Z_{i,j}^t, Z_{i,j}^{t+1}) \in [\varrho_{\min}(p), 1), \qquad \varrho_{\min}(p) := \max\left(-\frac{1-p}{p}, -\frac{p}{1-p}\right),$$

is the (edge-type-specific) lag-1 correlation ϱ_{fp} (honest edges) or ϱ_{fn} (attacker edges). Chains on distinct edges are independent.

Taking $\varrho = 0$ gives $a = b = p$, *i.e.*, Assumption 2. To handle the correlation, we track the rejection indicator jointly with the trust state machine. Let $\mathcal{X}$ be the finite set of configurations of the state machine of Theorem 14: TRUSTED with a current run of fewer than k_1 consecutive detections, SUSPECTED with the number of rounds elapsed in the evaluation window (fewer than k_3) and the number of detections so far (fewer than k_2), and EJECTED. The state rule defines a deterministic update $x_t = \tau(x_{t-1}, Z_t)$, so that under Assumption 15 the pair $(x_t, Z_t)_{t\geq 0}$ is a time-homogeneous Markov chain on $\mathcal{X} \times \{0,1\}$, in which EJECTED is absorbing and (x_0, Z_0) is the state before the first round. We denote its transition matrix by $\mathbf{Q}$ and, for $L \geq 1$, define

$$q_{\min}(L) := \min_{\substack{x \in \mathcal{X}\setminus\{\text{EJECTED}\} \\ z \in \{0,1\}}} \sum_{z' \in \{0,1\}} \left(\mathbf{Q}^L\right)_{(x,z),(\text{EJECTED},z')},$$

the probability of reaching EJECTED within L rounds from the least favorable configuration. We also let

$$w_{k_2,k_3}(a,b) := \mathbb{P}\Big(\textstyle\sum_{s=1}^{k_3} Z_s \geq k_2 \,\Big|\, Z_0 = 1\Big) \qquad \text{and} \qquad \pi_\varrho(p) := p\, a^{k_1-1}\, w_{k_2,k_3}(a,b).$$

Proposition 16 (Ejection bounds under Markov-correlated detections)**.** *Fix $T \geq 1$ and integers $k_1 \geq 1$, $1 \leq k_2 \leq k_3$, and let A_T and N_T be as in Theorem 14. Under Assumption 15,*

$$\mathbb{P}(A_T = 1) \leq N_T\, \pi_\varrho(p),$$

and, for any integer $1 \leq L \leq T$, letting $K_L := \lfloor T/L \rfloor$,

$$\mathbb{P}(A_T = 1) \;\geq\; 1 - \big(1 - q_{\min}(L)\big)^{K_L}, \qquad \textit{hence} \qquad \mathbb{P}(A_T = 1) \;\geq\; \max_{1 \leq L \leq T} \Big[1 - \big(1 - q_{\min}(L)\big)^{K_L}\Big].$$

Proof. **Upper bound.** As in Appendix I.1, any ejection by round T is triggered by one of the windows C_i, $i \in \{1, \ldots, N_T\}$, so $\{A_T = 1\} \subseteq \bigcup_{i=1}^{N_T} C_i$. This inclusion only depends on the state rule, not on the law of the indicators. By stationarity and the Markov property, $\mathbb{P}(C_i) = p\, a^{k_1-1}\, w_{k_2,k_3}(a,b) = \pi_\varrho(p)$ for every i, and the union bound, which requires no independence, concludes.

Lower bound. Partition the first $K_L L$ rounds into K_L consecutive blocks of length L, and let $\mathcal{F}_{\ell L}$ be the history up to the start of block $\ell \in \{0, \ldots, K_L - 1\}$. By the Markov property of (x_t, Z_t), the conditional probability of reaching EJECTED during block ℓ given $\mathcal{F}_{\ell L}$ only depends on the configuration at the start of the block, and is at least $q_{\min}(L)$ whenever the node is not yet ejected. Hence $\mathbb{P}\big(x_{(\ell+1)L} \neq \text{EJECTED} \mid \mathcal{F}_{\ell L}\big) \leq \big(1 - q_{\min}(L)\big)\, \mathbb{I}_{\{x_{\ell L} \neq \text{EJECTED}\}}$, and applying the tower property over $\ell = K_L - 1, \ldots, 0$ yields $\mathbb{P}(A_T = 0) \leq \mathbb{P}(x_{K_L L} \neq \text{EJECTED}) \leq (1 - q_{\min}(L))^{K_L}$. □

Applying Proposition 16 with $(p, \varrho) = (p_{\mathrm{fp}}, \varrho_{\mathrm{fp}})$ and $(p, \varrho) = (1 - p_{\mathrm{fn}}, \varrho_{\mathrm{fn}})$ yields the counterpart of Proposition 4:

$$\mathbb{P}(A_T^{\mathrm{hon}} = 1) \le \max\{T - k_1 - k_2 + 1, 0\}\, \pi_{\varrho_{\mathrm{fp}}}(p_{\mathrm{fp}}), \quad \mathbb{P}(B_T^{\mathrm{mal}} = 1) \le \min_{1 \le L \le T} \big(1 - q_{\min}(L)\big)^{\lfloor T/L \rfloor}.$$

We make three observations on this result. (i) First, for $k_2 = 1$, the value used in our experiments, $w_{1,k_3}(a, b) = 1 - (1 - a)(1 - b)^{k_3 - 1}$, so that $\pi_\varrho(p)$ is explicit and $\pi_0(p) = \pi(p)$: the honest bound of Proposition 4 is recovered at $\varrho_{\mathrm{fp}} = 0$. For general k_2, $w_{k_2,k_3}(a, b)$ is computed by a recursion over the rounds of the window that tracks the last indicator and the number of detections so far. (ii) Second, the bound on $\mathbb{P}(B_T^{\mathrm{mal}} = 1)$ only requires powers of the matrix $\mathbf{Q}$, whose size is at most $2(k_1 + k_2 k_3 + 1)$, and holds for all thresholds since $q_{\min}(L)$ accounts for every configuration in which a block may start. (iii) Third, this bound is never looser than the one of Proposition 4 when $k_2 = 1$ and $\varrho_{\mathrm{fn}} = 0$: the proof of Theorem 14 shows that ejection occurs within T rounds with probability at least $1 - (1 - \hat{\pi}(p))^{M_T}$ from any configuration, hence $1 - q_{\min}(T) \le (1 - \hat{\pi}(p))^{M_T}$.

Remark 17 (Measured correlations)**.** *We evaluate Assumption 15 on our CIFAR-10 experiments ($\alpha = 0.5$, three seeds), taking $Z_{i,j}^t$ to be the outcome of collaborative verification on the rounds before an edge reaches* EJECTED*, and obtain $\varrho_{\mathrm{fp}} \approx 0.09$ on honest edges (17,112 transitions) and $\varrho_{\mathrm{fn}} \approx 0.31$ on attacker edges (643 transitions, standard error 0.07). Note that p_{fp} and p_{fn} are aggregate rates over the whole run, which include the rounds in which a neighbor is already* SUSPECTED *or* EJECTED*, whereas the correlations are fitted on the rounds before ejection; ϱ_{fn} in particular rests on few transitions, since attackers are ejected early, and should be read as indicative.*

Remark 18 (Numerical instance)**.** *With $p_{\mathrm{fp}} = 1.1\%$, $p_{\mathrm{fn}} = 20.0\%$, $(k_1, k_2, k_3) = (2, 1, 3)$ and $T = 50$ as in Remark 5, the two bounds above evaluate to:*

Correlation	$\mathbb{P}(A_T^{\mathrm{hon}} = 1) \le$	$\mathbb{P}(B_T^{\mathrm{mal}} = 1) \le$
$\varrho_{\mathrm{fp}} = \varrho_{\mathrm{fn}} = 0$ *(Assumption 2)*	0.019%	$< 0.01\%$
$\varrho_{\mathrm{fp}} = 0.09$, $\varrho_{\mathrm{fn}} = 0.31$ *(measured)*	0.62%	$< 0.01\%$
$\varrho_{\mathrm{fp}} = \varrho_{\mathrm{fn}} = 0.3$	5.2%	$< 0.01\%$
$\varrho_{\mathrm{fp}} = \varrho_{\mathrm{fn}} = 0.6$	19.4%	$< 0.01\%$

At the measured correlations, the honest-ejection bound moves from 0.019% to 0.62%, still well under 1%, and the attacker-survival bound is unchanged. Correlation mostly affects honest edges, as bursts of false positives make the k_1 consecutive rejections that open an evaluation window more likely.

J End-to-end convergence under attack

This section states and proves the formal version of Corollary 8, which combines the ejection bound of Proposition 4 with the convergence rate of Theorem 7.

Corollary 8 (End-to-end convergence under attack, informal)**.** *Fix $\delta \in (0, 1)$ and $\epsilon > 0$. Under the assumptions of Proposition 4 and Theorem 7, and if the loss and the disagreement of honest models remain bounded while attackers are present (Assumption 19), there exist $T_{\mathrm{ej}} = \mathcal{O}(\log(m/\delta))$ and $T_{\mathrm{cv}} = \mathcal{O}(1/(\delta^2 \epsilon^2))$ such that, running* ARGUS *for $T = T_{\mathrm{ej}} + T_{\mathrm{cv}}$ rounds,*

$$\mathbb{P}\left(\frac{1}{|\mathcal{H}|\, T_{\mathrm{cv}}} \sum_{i \in \mathcal{H}} \sum_{t = T_{\mathrm{ej}}}^{T-1} \left\| \nabla \mathcal{L}(\theta_i^t) \right\|^2 \le \epsilon \right) \;\ge\; 1 - \delta.$$

J.1 Setting and additional assumption

Following the assumptions of Theorem 7 that all attackers have been ejected, all quantities of that theorem are understood here when only honest nodes remain: $n := |\mathcal{H}|$, average model over $\mathcal{H}$, $\mathcal{L}$ is the average of the honest local losses, and d, ρ, $\beta(d, p_{\mathrm{fp}})$ and C_ρ are those of the graph induced by the honest nodes. We write $\mathcal{E}_{\mathcal{M}} := \{(i, j) : i \in \mathcal{H},\ j \in N(i) \cap \mathcal{M}\}$ for the set of honest–attacker edges and $\mathcal{E}_{\mathcal{H}} := \{(i, j) : i \in \mathcal{H},\ j \in N(i) \cap \mathcal{H}\}$ for the set of ordered pairs of honest neighbors; each such pair carries its own trust state machine. As in the proofs of Proposition 6 and Theorem 7, the indicators of Assumption 2 are taken independent across edges and rounds, and independent of

the stochastic gradients, which are unbiased, and of the attackers' behavior. Rounds are indexed from 0, $\mathcal{F}_t$ denotes the history of the first t rounds, and we use:

$$\bar{\sigma}^2 := \sigma^2 + \varsigma^2, \qquad \Xi_t := \frac{1}{n}\sum_{i\in\mathcal{H}} \left\|\theta_i^t - \overline{\theta}^t\right\|^2, \qquad \Delta_t := \mathcal{L}\big(\overline{\theta}^t\big) - \mathcal{L}^*.$$

Theorem 7 describes the system once attackers are gone. Before that, attackers may have arbitrary models accepted by their honest neighbors, so that the state reached by the honest nodes when the last attacker is ejected is not controlled by the assumptions made so far. We therefore assume that this transient phase leaves the honest nodes in a bounded state.

Assumption 19 (Bounded transient)**.** *There exist $\Delta_{\max}, \Xi_{\max} < \infty$ such that, at round $t_0 = T_{\mathrm{ej}}(\delta)$ defined in Corollary 21, the honest models satisfy $\Delta_{t_0} \le \Delta_{\max}$ and $\Xi_{t_0} \le \Xi_{\max}$ almost surely.*

The first condition holds as soon as the training loss is bounded. Both conditions hold when the models produced during the transient phase remain in a bounded set, in which case $\Xi_{\max}$ is at most the squared diameter of this set.

J.2 Convergence from an arbitrary starting point

The proof of Theorem 7 relies on a uniform bound on the model drift (Lemma 13) that starts from identical models. The following lemma removes this requirement, at the price of an additional term that depends on the initial disagreement Ξ_{t_0}.

Lemma 20 (Convergence from an arbitrary start)**.** *Assume that the setting of Theorem 7 holds from some round t_0 on, the models at round t_0 being arbitrary and the masks and gradient noise of rounds $t \ge t_0$ being independent of $\mathcal{F}_{t_0}$. Let $T' \ge 1$ and consider a constant stepsize $\eta \le \frac{1-\rho}{5L}$ over rounds $t_0, \dots, t_0 + T' - 1$. Then*

$$\frac{1}{n}\sum_{i\in\mathcal{H}} \frac{1}{T'} \sum_{t=t_0}^{t_0+T'-1} \mathbb{E}\left[\left\|\nabla\mathcal{L}(\theta_i^t)\right\|^2 \mid \mathcal{F}_{t_0}\right] \le \frac{4\widetilde{\Delta}_{t_0}}{\eta T'} + \frac{\eta}{n}\Gamma_1 + \eta^2\Gamma_2 + \frac{16L^2\,\Xi_{t_0}}{(1-\rho)T'},$$

where $\widetilde{\Delta}_{t_0} := \Delta_{t_0} + \frac{32L\beta(d,p_{\mathrm{fp}})}{n(1-\rho)}\Xi_{t_0}$ and

$$\Gamma_1 := L\Big(4\sigma^2 + \big(24 + 384C_\rho\big)\beta(d,p_{\mathrm{fp}})\,\bar{\sigma}^2\Big), \qquad \Gamma_2 := 48L^2C_\rho\,\bar{\sigma}^2.$$

Proof. All expectations are conditional on $\mathcal{F}_{t_0}$; for $t > t_0$ we still write Ξ_t and Δ_t for the conditional expectations of these quantities. Since the masks of rounds $t \ge t_0$ are independent of $\mathcal{F}_{t_0}$ and of the gradient noise, Lemma 12 applies conditionally.

Gradient drift. Let $\overline{g}^t := \frac{1}{n}\sum_{i\in\mathcal{H}} g_i^t$. Since the average minimizes the sum of squared distances, and decomposing $g_i^t - \nabla\mathcal{L}(\overline{\theta}^t)$ into $g_i^t - \nabla\mathcal{L}_i(\theta_i^t)$, $\nabla\mathcal{L}_i(\theta_i^t) - \nabla\mathcal{L}_i(\overline{\theta}^t)$ and $\nabla\mathcal{L}_i(\overline{\theta}^t) - \nabla\mathcal{L}(\overline{\theta}^t)$,

$$\frac{1}{n}\sum_{i\in\mathcal{H}} \mathbb{E}\left\|g_i^t - \overline{g}^t\right\|^2 \le \frac{1}{n}\sum_{i\in\mathcal{H}} \mathbb{E}\left\|g_i^t - \nabla\mathcal{L}(\overline{\theta}^t)\right\|^2 \le 3\bar{\sigma}^2 + 3L^2\,\Xi_t, \tag{19}$$

using the bounded variance, L-smoothness and bounded heterogeneity.

Model drift. Since $\theta_i^{t+1/2} - \overline{\theta}^{t+1/2} = (\theta_i^t - \overline{\theta}^t) - \eta(g_i^t - \overline{g}^t)$, Lemma 12(b), Young's inequality with a parameter $u > 0$ and Equation (19) give

$$\Xi_{t+1} \le \rho(1+u)\,\Xi_t + \rho\Big(1 + \frac{1}{u}\Big)\eta^2\big(3\bar{\sigma}^2 + 3L^2\,\Xi_t\big).$$

If $\rho = 0$ then $\Xi_{t+1} = 0$. Otherwise, take $u = \frac{1-\rho}{2\rho}$, so that $\rho(1+u) = \frac{1+\rho}{2}$ and $\rho(1+\frac{1}{u}) = \frac{\rho(1+\rho)}{1-\rho}$. As $\eta L \le \frac{1-\rho}{5}$ and $\rho(1+\rho) \le 2$, we have $\frac{3\rho(1+\rho)}{1-\rho}\eta^2L^2 \le \frac{1-\rho}{4}$, hence

$$\Xi_{t+1} \le \frac{3+\rho}{4}\,\Xi_t + \frac{3\rho(1+\rho)}{1-\rho}\eta^2\bar{\sigma}^2.$$

Unrolling this recursion from t_0 and using $\rho(1+\rho) \le (1-\rho)^2 C_\rho$ (Equation (16)), we get, for all $s \ge 0$,

$$\Xi_{t_0+s} \le \Big(\frac{3+\rho}{4}\Big)^s \Xi_{t_0} + 12 C_\rho \eta^2 \bar{\sigma}^2, \qquad \text{thus} \qquad \frac{1}{T'} \sum_{t=t_0}^{t_0+T'-1} \Xi_t \le \frac{4\,\Xi_{t_0}}{(1-\rho)T'} + 12 C_\rho \eta^2 \bar{\sigma}^2. \tag{20}$$

Descent. The one-step inequality (13) holds at every round, irrespective of the initial models, for $\eta \le \frac{1}{2L}$ [3, Lemma 3]. Its last term is bounded with Lemma 12(c) and Equation (19):

$$\mathbb{E}\Big\|\overline{\theta}^{t+1} - \overline{\theta}^{t+1/2}\Big\|^2 \le \frac{\beta(d, p_{\mathrm{fp}})}{n}\Big(2\,\Xi_t + 2\eta^2\big(3\bar{\sigma}^2 + 3L^2\,\Xi_t\big)\Big).$$

Plugging this into Equation (13), and using $\eta L \le 1$ to bound $24L^3\beta(d,p_{\mathrm{fp}})\eta/n$ by $24L\beta(d,p_{\mathrm{fp}})/(n\eta)$, we obtain

$$\frac{1}{n}\sum_{i\in\mathcal{H}} \mathbb{E}\big\|\nabla\mathcal{L}(\theta_i^t)\big\|^2 \le \frac{4}{\eta}\big(\Delta_t - \Delta_{t+1}\big) + \Big(4L^2 + \frac{32L\beta(d,p_{\mathrm{fp}})}{n\eta}\Big)\Xi_t + \frac{\eta}{n}L\big(4\sigma^2 + 24\beta(d,p_{\mathrm{fp}})\,\bar{\sigma}^2\big).$$

Averaging over $t \in \{t_0, \dots, t_0+T'-1\}$, the first term telescopes to at most $\frac{4\Delta_{t_0}}{\eta T'}$ since $\Delta_{t_0+T'} \ge 0$, and Equation (20) bounds the second one by

$$\frac{16L^2\,\Xi_{t_0}}{(1-\rho)T'} + \frac{4}{\eta T'}\cdot\frac{32L\beta(d,p_{\mathrm{fp}})}{n(1-\rho)}\Xi_{t_0} + 48L^2 C_\rho\eta^2\bar{\sigma}^2 + \frac{\eta}{n}\,384LC_\rho\beta(d,p_{\mathrm{fp}})\,\bar{\sigma}^2.$$

Collecting terms gives the claim. □

With identical initial models ($\Xi_{t_0} = 0$), Lemma 20 has the same form as Equation (17): the transient only enters through the effective initial gap $\widetilde{\Delta}_{t_0}$ and a term of order $1/T'$.

J.3 Formal statement and proof

Corollary 21 (Formal version of Corollary 8)**.** *Fix $\delta \in (0,1)$ and $\epsilon > 0$. Let Assumption 2 hold with $0 < p_{\mathrm{fn}} < 1$, let $k_2 = 1$ and $\hat{p} := \hat{\pi}(1-p_{\mathrm{fn}}) = p_{\mathrm{fn}}\,\pi(1-p_{\mathrm{fn}})$ with π as in Proposition 4, and define*

$$T_{\mathrm{ej}}(\delta) := (k_1 + k_3 + 1)\left\lceil \frac{\ln\big(2\,|\mathcal{E}_{\mathcal{M}}|\,/\delta\big)}{\ln\big(1/(1-\hat{p})\big)} \right\rceil.$$

Assume that the other hypotheses of Theorem 7 hold for the honest nodes, that honest neighbors always remain TRUSTED *(see Remark 22), and that Assumption 19 holds with $\widetilde{\Delta} > 0$ defined below. Let Γ_1 and Γ_2 be as in Lemma 20, $\widetilde{\Delta} := \Delta_{\max} + \frac{32L\beta(d,p_{\mathrm{fp}})}{n(1-\rho)}\Xi_{\max}$, and*

$$T_{\mathrm{cv}}(\delta,\epsilon) := \left\lceil \max\left\{ \frac{24\big(5L\widetilde{\Delta} + 4L^2\,\Xi_{\max}\big)}{(1-\rho)\,\delta\epsilon},\ \frac{576\,\widetilde{\Delta}\,\Gamma_1}{n\,\delta^2\epsilon^2},\ \frac{96\sqrt{3}\,\widetilde{\Delta}\sqrt{\Gamma_2}}{(\delta\epsilon)^{3/2}} \right\} \right\rceil.$$

Run Algorithm 1 for $T := T_{\mathrm{ej}}(\delta) + T_{\mathrm{cv}}(\delta,\epsilon)$ rounds, using from round $T_{\mathrm{ej}}(\delta)$ on the constant stepsize

$$\eta := \min\left\{ \frac{1-\rho}{5L},\ \sqrt{\frac{4n\widetilde{\Delta}}{\Gamma_1\,T_{\mathrm{cv}}(\delta,\epsilon)}},\ \left(\frac{4\widetilde{\Delta}}{\Gamma_2\,T_{\mathrm{cv}}(\delta,\epsilon)}\right)^{\frac{1}{3}} \right\}.$$

where a term with a null denominator is read as $+\infty$. Then

$$\mathbb{P}\left(\frac{1}{n}\sum_{i\in\mathcal{H}} \frac{1}{T_{\mathrm{cv}}(\delta,\epsilon)} \sum_{t=T_{\mathrm{ej}}(\delta)}^{T-1} \big\|\nabla\mathcal{L}(\theta_i^t)\big\|^2 \le \epsilon \right) \ \ge\ 1-\delta.$$

In particular, $T_{\mathrm{ej}}(\delta) = \mathcal{O}\big(\log(|\mathcal{E}_{\mathcal{M}}|\,/\delta)\big)$ and $T_{\mathrm{cv}}(\delta,\epsilon) = \mathcal{O}\big(\widetilde{\Delta}\,\Gamma_1/(n\delta^2\epsilon^2)\big)$ as $\delta\epsilon \to 0$.

Proof. Write $t_0 := T_{\mathrm{ej}}(\delta), T' := T_{\mathrm{cv}}(\delta, \epsilon)$, and let X be the average squared gradient norm appearing in the statement.

Step 1: ejection. Proposition 4 applies to the trust state machine of each edge $(i, j) \in \mathcal{E}_{\mathcal{M}}$: the probability that i has not ejected j within the first t_0 rounds is at most $(1 - \hat{p})^{\lfloor t_0/(k_1+k_3+1) \rfloor} \leq \delta/(2\,|\mathcal{E}_{\mathcal{M}}|)$ by the choice of t_0. Let Ω_{ej} be the event that every attacker has been ejected by all its honest neighbors within the first t_0 rounds. By the union bound over $\mathcal{E}_{\mathcal{M}}$, $\mathbb{P}(\Omega_{\mathrm{ej}}^c) \leq \delta/2$.

Step 2: convergence after ejection. The event Ω_{ej} belongs to $\mathcal{F}_{t_0}$. On Ω_{ej}, honest nodes no longer aggregate any attacker model from round t_0 on, and the masks and gradient noise of these rounds are independent of $\mathcal{F}_{t_0}$. Lemma 20 thus applies on Ω_{ej}. Its right-hand side is non-decreasing in Δ_{t_0} and Ξ_{t_0}, which are bounded by $\Delta_{\max}$ and $\Xi_{\max}$ (Assumption 19), so that, on Ω_{ej},

$$\mathbb{E}\left[X \mid \mathcal{F}_{t_0}\right] \leq R := \frac{4\widetilde{\Delta}}{\eta T'} + \frac{\eta}{n}\Gamma_1 + \eta^2\Gamma_2 + \frac{16L^2\,\Xi_{\max}}{(1-\rho)T'}.$$

Step 3: stepsize. The inverse of the stepsize is at most the sum of the inverses of the three terms of the minimum defining it, hence

$$R \leq \frac{20L\widetilde{\Delta} + 16L^2\,\Xi_{\max}}{(1-\rho)T'} + 4\sqrt{\frac{\widetilde{\Delta}\,\Gamma_1}{nT'}} + 2\left(\frac{16\,\widetilde{\Delta}^2\,\Gamma_2}{T'^2}\right)^{\frac{1}{3}}.$$

Each of the three terms in the definition of T' makes the corresponding term above at most $\delta\epsilon/6$, so $R \leq \delta\epsilon/2$.

Step 4: tail bound. As $X \geq 0$, Markov's inequality applied conditionally on $\mathcal{F}_{t_0}$ gives

$$\mathbb{P}\big(\{X > \epsilon\} \cap \Omega_{\mathrm{ej}}\big) = \mathbb{E}\big[\mathbb{I}_{\Omega_{\mathrm{ej}}}\,\mathbb{P}(X > \epsilon \mid \mathcal{F}_{t_0})\big] \leq \frac{1}{\epsilon}\,\mathbb{E}\big[\mathbb{I}_{\Omega_{\mathrm{ej}}}\,\mathbb{E}\left[X \mid \mathcal{F}_{t_0}\right]\big] \leq \frac{R}{\epsilon} \leq \frac{\delta}{2}.$$

Therefore $\mathbb{P}(X > \epsilon) \leq \mathbb{P}(\Omega_{\mathrm{ej}}^c) + \mathbb{P}(\{X > \epsilon\} \cap \Omega_{\mathrm{ej}}) \leq \delta$. □

Remark 22 (Scope and extensions)**.** (i) Honest neighbors. *Theorem 7 and Corollary 21 assume that honest neighbors always remain* TRUSTED*: the masks S^t model neither the systematic rejection of a* SUSPECTED *neighbor nor its ejection.* ARGUS *behaves exactly like this until an honest neighbor becomes* SUSPECTED*, which takes k_1 consecutive false rejections. By a union bound, this happens within T rounds with probability at most $|\mathcal{E}_{\mathcal{H}}|\;T\,p_{\mathrm{fp}}^{k_1}$, so the guarantee of Corollary 21 holds for* ARGUS *with probability at least $1 - \delta - |\mathcal{E}_{\mathcal{H}}|\;T\,p_{\mathrm{fp}}^{k_1}$. This extra term grows with T but decreases geometrically with k_1.*

(ii) Other ejection bounds. *The proof only needs a bound on the probability that an attacker edge survives t_0 rounds. $T_{\mathrm{ej}}(\delta)$ can thus be replaced by $\ell\,\lceil \ln(2\,|\mathcal{E}_{\mathcal{M}}|\,/\delta)/\ln(1/(1-q_{\min}(\ell)))\rceil$ for any block length ℓ with $0 < q_{\min}(\ell) < 1$ (Proposition 16), which holds for all thresholds and when attacker detections are correlated (Assumption 15).*

(iii) Topology. *Theorem 7 requires a connected regular graph. Here, this applies to the graph induced by the honest nodes, which is restrictive: removing attackers from a regular graph generally breaks regularity.*

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: References to the sections that back the claims are made in the introduction. Specifically, Section 3 introduces ARGUS, Section 4 provides the convergence analysis and Section 5 justifies the claims on experiments.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss limitations and future directions in a dedicated paragraph in Section 6, as well as in Appendix G.8.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: We provide all proofs of claims from Section 4 in the appendix, specifically in Appendix H, Appendix I and Appendix J. Moreover, our only novel assumption linked to our setting is clearly stated in Assumption 2, and other assumptions are part of the theorems themselves.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We disclose all information needed for reproduction in Section 5.1 and Appendix E.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We have included a link to the code in Section 5.1, with scripts and documentation to reproduce the results.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide main experimental setting details in Section 5.1, in addition to Appendix E as well as Appendix F for baselines.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report mean and standard deviation over 3 seeds in Table 1 and Figure 5.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide this information in the dedicated paragraph in Appendix E.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have read and agree with the code of ethics, and have ensured we adhere to it.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the need for a DL backdoor defense scheme and the corresponding positive impacts in Section 1.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks. We do not release models and only rely on public datasets (CIFAR-10, FEMNIST, TinyImageNet).

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The datasets (CIFAR-10, FEMNIST and TinyImageNet), the network architectures and the DECENTRALIZEPY framework are properly cited in Section 5.1 and Appendix E. All are used under their respective licenses, for research purposes.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The implementation of ARGUS is included in a public repository, with the appropriate documentation (including README, run scripts and configuration files).

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects, so no IRB review is applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.