

FedLoRAGuard: Federated Dynamic Graph Neural Networks with Differential-Privacy Certificates for LoRA Adapter Integrity Verification

Roger Nick Anaedevha^{1,*} Alexander G. Trofimov^{1,*} Yuri V. Borodachev^{2,*}

¹Institute of Cyber Intelligence Systems,

National Research Nuclear University MEPhI, Moscow 115409, Russia

²Artificial Intelligence Research Center,

National Research Nuclear University MEPhI, Moscow 115409, Russia

ar006@campus.mephi.ru

Abstract

Low-Rank Adaptation (LoRA) is now the dominant parameter-efficient fine-tuning method for large language models, but the resulting open marketplace of adapters has become a supply-chain attack surface: backdoored adapters merge silently with arbitrary task adapters while preserving benign utility, and existing centralized integrity tools require marketplaces to share raw weights, which is incompatible with proprietary fine-tuning. We present FedLoRAGuard, a federated dynamic-graph framework that verifies adapter integrity without weight sharing. The LoRA ecosystem is modeled as a heterogeneous continuous-time dynamic graph with four node types and six edge types; a federated dynamic GNN is trained per-client with client-update-level (ϵ, δ) -DP Gaussian noise, secure aggregation, and FLTrust scoring. We prove a sensitivity bound for federated heterogeneous-GNN client updates and a certified poisoning-radius theorem that converts the per-round inference-time smoothing budget ϵ_r into an analytical bound on the size of a bounded-update colluding coalition k^* . On 13,500 benign and backdoored adapters across four base-model families and ten attack types, FedLoRAGuard obtains 96.4% macro-F1 and a certified coalition bound $k^* = 25$ from $N = 50$ marketplaces on the inference-time smoothing budget $\epsilon_r = 0.5$, and empirically tolerates label-flipping Byzantine coalitions up to $\rho \leq 0.33$ without certificate degradation. The DP-SGD training-time mechanism, calibrated at $\sigma = 1.1$ and Poisson subsampling $q = 0.2$ over $T = 100$ rounds, achieves a cumulative privacy budget $\epsilon_T \approx 13.6$ at $\delta = 10^{-5}$ under the Opacus PRV accountant, with sub-second per-adapter verification latency on A100-class hardware. The full DP + secure-aggregation stack adds a measured median per-round wall-clock overhead of $\approx 4\%$

(pooled median across three runs of 20–60 measurement rounds on identical CPU hardware) over non-private FedAvg; the released runtime ships the instrumentation so operators can re-measure in place on their target hardware.

1 Introduction

Low-Rank Adaptation (LoRA) (Hu et al., 2021) is the dominant parameter-efficient fine-tuning method for large language models: it inserts low-rank update matrices into Transformer attention, freezes base weights, and cuts trainable-parameter counts by four orders of magnitude. Combined with 4-bit QLoRA (Dettmers et al., 2023), LoRA allows an individual to fine-tune a 65B base model on a single consumer GPU. The Hugging Face Hub now hosts more than 530,000 models, and one base model alone has accumulated more than 30,000 derivative LoRAs.

That ecosystem is also a supply-chain attack surface. Liu et al. (2024b) showed that backdoored adapters merge silently with arbitrary task adapters and retain malicious behavior while preserving benign utility; Lermen et al. (2023) reduced the safety-refusal rate of Llama-2-Chat-70B from 95% to below 1% with a USD-200 LoRA fine-tune; composite backdoors (Huang et al., 2024) scatter triggers across prompt components. Industry reports document >100 malicious models on the Hub, indirect compromise of the platform’s own SFconvertbot (>42,000 merged PRs), and a five-fold year-over-year rise in malicious uploads (Cohen, 2024; Kellas et al., 2025). Those figures are predominantly full-model backdoors; the LoRA-*adapter* threat we target is documented separately by the adapter-specific work above and marketplace-scale numbers are cited only as context, not as an adapter-backdoor count.

The standard centralized response, PEFT-Guard (Sun et al., 2025), trains a backdoor detector on raw adapter weights (98.3–100% centralized ac-

* * All authors of this publication are co-first authors (equal contribution). Corresponding author: ar006@campus.mephi.ru

curacy on PADBench) but requires marketplaces to share raw weights with a third-party detector – incompatible with proprietary fine-tuning practice and a single point of trust. A weight-space-only detector also sees each adapter in isolation and misses the cross-marketplace lineage signal that distinguishes a silently backdoored derivative from a benign one. We do not run an adaptive attack against PEFTGuard and make no claim about its robustness under one; our motivation is the deployment barrier and the missing relational signal, not a claimed collapse of PEFTGuard.

Our approach. We model the LoRA ecosystem as a heterogeneous continuous-time dynamic graph whose nodes are adapters, contributors, base models and downstream applications, and whose time-stamped edges capture derivation, fine-tuning, deployment, and citation. Across this graph, a dynamic GNN is trained per client with *client-update-level* DP Gaussian noise (Abadi et al., 2016; Mironov, 2017) at the client update level and secure aggregation (Bonawitz et al., 2017), so no marketplace reveals raw weights. Composing PixelDP (Lecuyer et al., 2019) stability with CRFL (Xie et al., 2021) federated smoothing yields a certified coalition bound k^* on the number of colluding malicious clients whose actions cannot flip the global verdict, under a bounded-update threat model (Definition 2). We ship the released runtime with Opacus PRV-accountant instrumentation, an SVD-only spectral baseline, and a per-round overhead harness so every claim is independently re-measurable.

Contributions. (1) The first formal model, to our knowledge, of the LoRA marketplace as a continuous-time heterogeneous dynamic graph, and LoRAchain-2026, 13,500 benign+backdoored adapters over four base-model families and ten attack types, extending BackdoorLLM (Li et al., 2025) and PADBench (Sun et al., 2025) with the HuggingGraph (Rahman et al., 2025) lineage. (2) A federated DyGFormer-HGT verifier trained under client-update-level DP-SGD with SecAgg and FLTrust bootstrapping, robust to $\rho \leq 0.33$ label-flip Byzantine clients. (3) A client-update sensitivity bound (Theorem 1) and a certified coalition-bound theorem (Theorem 2) that converts the inference-time smoothing budget ϵ_r into an integer bound k^* under Definition 2. On LoRAchain-2026 this gives 96.4% macro-F1, $k^* = 25$ at $N = 50$ under $\epsilon_r = 0.5$, cumulative $(\epsilon_T, \delta) \approx (13.6, 10^{-5})$

over $T = 100$ from the Opacus PRV accountant (Gopi et al., 2021), and sub-second per-adapter latency on A100.

Claim structure. The paper is organized around three tiered claims. The *primary* claim is that federated lineage-aware detection can identify malicious LoRA adapters without any marketplace sharing raw weights, landing within 1.7 macro-F1 points of the centralized PEFTGuard reference on the same corpus. The *secondary* claim is that DP-SGD combined with SecAgg can be incorporated at modest utility cost (0.5–1.0 pp macro-F1) and at a measured $\approx 4\%$ median per-round wall-clock overhead on CPU-hosted federation (Appendix E). The *theoretical* claim is that, under Definition 2, composing PixelDP stability, CRFL federated smoothing and Rényi-DP composition yields an integer certificate k^* on the largest bounded-update coalition whose actions cannot flip the global verdict on a queried adapter. Every other result — dynamic-graph superiority, multimodal gains, FLTrust robustness, lineage importance, IDS transfer, cross-hardware latency — is reported as supporting evidence, not as a free-standing contribution.

Why this belongs at an NLP venue. LoRA is the dominant fine-tuning method for *language* models, and the verified threats — alignment-bypass fine-tunes, composite textual-trigger backdoors and instruction-following corruption — are attacks on the safety of NLP systems at the PEFT layer. The signal separating a benign adapter from a backdoored one is partly linguistic: the multimodal encoder consumes model-card, README and contributor-profile text alongside the weight and usage modalities, and LoRAchain-2026 extends NLP-safety resources (BackdoorLLM, PADBench). The verifier itself is task-agnostic, but the resource, the threat model, and the deployment target (Hugging Face-style LLM adapter hubs) are squarely NLP-safety concerns, which is why we position the contribution for the ACL community rather than a general security venue.

Roadmap. Section 2 surveys prior work on adapter security, federated learning with DP, dynamic GNNs and certified poisoning robustness. Section 3 formalizes the ecosystem graph and the threat model. Section 4 presents the four-component framework. Section 5 states and proves the sensitivity bound and the certified coalition bound. Section 6 reports the empirical study, in-

cluding ablation, spectral analysis, and IDS transfer. Sections 7–8 discuss deployment implications and conclude.

2 Related Work

LoRA, PEFT, and adapter security. Parameter-efficient fine-tuning rose with adapters and LoRA (Hu et al., 2021); QLoRA (Dettmers et al., 2023) added 4-bit NF4 quantization, and AdaLoRA (Zhang et al., 2023), DoRA (Liu et al., 2024a), and GaLore (Zhao et al., 2024) produced structurally distinct adapter families that complicate uniform integrity checks. Adapter security emerged with LoRA-as-an-Attack (Liu et al., 2024b), Lermen et al. (2023)’s cheap alignment bypass, Composite Backdoor Attacks (Huang et al., 2024), and the safety-critical neuron analysis of Wei et al. (2024). PEFTGuard (Sun et al., 2025) centralized backdoor detection on raw weights, and ShadowGenes (Schulz et al., 2025) introduced graph-based model-genealogy fingerprinting. Beyond these, ML supply-chain integrity has been approached through model signing, software bills of materials (SBOM), and update frameworks such as TUF; FedLoRAGuard is complementary to these provenance-attestation mechanisms, adding a content-level maliciousness verdict that signing alone does not provide.

Federated learning, DP, and secure aggregation. Federated learning (McMahan et al., 2017) with heterogeneity variants (Li et al., 2020; Karimireddy et al., 2020) combines with DP-SGD (Abadi et al., 2016), Rényi-DP accounting (Mironov, 2017), the numerically tight PRV accountant (Gopi et al., 2021), and secure aggregation (Bonawitz et al., 2017). Byzantine robustness is addressed by Krum (Blanchard et al., 2017), Trimmed-Mean (Yin et al., 2018), and FLTrust (Cao et al., 2021); a related game-theoretic treatment appears in (Anaedevha et al., 2026a). Federated foundation-model training (FATE-LLM (Fan et al., 2023), FLoRA (Wang et al., 2024), OpenFedLLM (Ye et al., 2024)) *trains* new adapters; FedLoRAGuard *verifies* existing ones. The closest is Safe-FedLLM (Tao et al., 2026), which filters malicious clients *during* federated training through probes on local Δ LoRA updates. The two are complementary: Safe-FedLLM protects the training process from malicious participants; FedLoRAGuard verifies adapters *already published* to a marketplace using cross-marketplace lineage, and issues a cer-

tified coalition bound k^* over colluding marketplaces — a guarantee probe-classifier defenses do not provide.

Dynamic and heterogeneous GNNs. Continuous-time dynamic graph learning was pioneered by JODIE (Kumar et al., 2019), DyRep (Trivedi et al., 2019) and TGAT (Xu et al., 2020), unified in TGN (Rossi et al., 2020); DyGFormer (Yu et al., 2023) added neighbour-co-occurrence patching for long histories. Multi-scale temporal graph encoders have been applied to security telemetry (Anaedevha and Trofimov, 2026). For heterogeneous graphs, HGT (Hu et al., 2020) introduced meta-relation-aware attention with relative temporal encoding. Federated graph learning was introduced by FedGraphNN (He et al., 2021), and GAP (Sajadmanesh et al., 2023) derives a DP-GNN through Rényi-DP analysis. The HF ecosystem itself is mapped by HuggingGraph (Rahman et al., 2025) and the model-atlas of Horwitz et al. (2025).

Certified robustness against poisoning. The DP-to-certified-robustness bridge was established by PixelDP (Lecuyer et al., 2019); Ma et al. (2019) converted DP-SGD budgets into poisoning radii. In federated settings, CRFL (Xie et al., 2021) uses parameter clipping plus Gaussian smoothing, and FLCert (Cao et al., 2022) extends this with a fraction-of-malicious-clients guarantee. State-space poisoning resilience without a formal certificate appears in (Anaedevha et al., 2026b). For graph data, Wang et al. (2021); Scholten et al. (2022) certify against edge and message-interception perturbations. FedLoRAGuard’s Theorem 2 synthesizes these into a single per-marketplace bound on the largest bounded-update coalition that cannot alter the global verdict.

3 Preliminaries and Problem

LoRA adapter. A LoRA adapter for a Transformer layer with frozen weight $\mathbf{W}_0 \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ replaces $\mathbf{y} = \mathbf{W}_0 \mathbf{x}$ with

$$\mathbf{y} = \mathbf{W}_0 \mathbf{x} + \frac{\alpha}{r} \mathbf{B} \mathbf{A} \mathbf{x}, \quad \mathbf{B} \in \mathbb{R}^{d_{\text{out}} \times r}, \quad \mathbf{A} \in \mathbb{R}^{r \times d_{\text{in}}}, \quad (1)$$

with rank $r \ll \min(d_{\text{out}}, d_{\text{in}})$ and fixed scale α (Hu et al., 2021). We denote a complete adapter by $\theta = \{(\mathbf{A}_\ell, \mathbf{B}_\ell)\}_{\ell \in \mathcal{L}}$ over adapted layers $\mathcal{L}$, plus metadata m_θ (base model, corpus, contributor, declared task and licence, upload time, download stats).

LoRA ecosystem graph. At time t the ecosystem is a heterogeneous continuous-time dynamic graph $\mathcal{G}(t) = (\mathcal{V}(t), \mathcal{E}(t), \mathcal{R}, \phi, \psi)$ with four node types and six edge types.

Definition 1 (LoRA Ecosystem Graph). $\mathcal{V}(t) = \mathcal{V}_A(t) \cup \mathcal{V}_B(t) \cup \mathcal{V}_C(t) \cup \mathcal{V}_D(t)$ (adapters, base models, contributors, downstream applications). Time-stamped typed edges use relations $\mathcal{R} = \{\text{contributes, derives_from, fine_tunes, deploys, cites, co_uses}\}$. The node-feature map $\phi : \mathcal{V}(t) \rightarrow \mathbb{R}^{d_v}$ is instantiated by the multimodal encoder of Section 4.

Federated setting and threat model. N LoRA-hosting marketplaces $\{C_i\}_{i=1}^N$ hold private subgraphs $\mathcal{G}_i(t)$ and adapter pools Θ_i ; they refuse to share raw weights but consent to federated training of $f_w : \mathcal{V}_A(t) \rightarrow [0, 1]$ under privacy-preserving aggregation. The *primary* threat is documented single/few-point malicious uploads to independent marketplaces; the multi-marketplace *collusion* we analyse is a worst-case robustness scenario, not a claim about current attacker behaviour. The adversary controls k of N clients with $k/N \leq \rho$, publishes arbitrary backdoored adapters under any identity, and flips training labels on controlled adapters; it does not control SecAgg nor the vetted FLTrust root set. The certified Theorem 2 applies to *bounded-update* adversaries below, not to arbitrary Byzantine gradients.

Definition 2 ((k, ϵ_{wt}) -poisoning attack). A poisoning attack controls $k \in \{1, \dots, \lfloor \rho N \rfloor\}$ clients and submits gradient updates $\tilde{\mathbf{g}}_i$ within ℓ_2 distance $\epsilon_{wt} \geq 0$ of the honest update after server-side clipping to ℓ_2 -norm S ; its objective is to flip the global verdict on at least one target adapter.

We seek $\mathbf{w}^*$ minimising empirical risk under per-round (ϵ_r, δ) -DP by client-update Gaussian DP-SGD (clip S), cumulative $\epsilon_T = \text{PRVCompose}(\sigma, q, T)$ at δ , SecAgg hiding individual $\tilde{\mathbf{g}}_i$, and FLTrust scoring on a server-held root set R_0 . The deployed verifier admits (a) a high-probability robustness certificate against k^* malicious clients (Theorem 2) under Definition 2, and (b) calibrated maliciousness probabilities for marketplace moderation.

4 The FedLoRAGuard Framework

FedLoRAGuard has four components: a multimodal node-feature encoder, a per-client hetero-

geneous dynamic GNN, federated training under client-update-level DP-SGD with SecAgg and FLTrust, and a global verifier that emits $\hat{p}_{\text{mal}}$ and the certified bound k^* (Figure 1).

Multimodal encoder. For $v_\theta \in \mathcal{V}_A$, weight features $\phi^{\text{wt}}(\theta) \in \mathbb{R}^{d_{\text{wt}}}$ comprise the spectral signature $(\sigma_1, \dots, \sigma_r)$ of each BA in (1), per-layer Frobenius norms, and a base-model weight-difference fingerprint. Text features $\phi^{\text{txt}}(m_\theta) \in \mathbb{R}^{d_{\text{txt}}}$ come from a frozen sentence transformer over model cards, READMEs and contributor profiles. Behavioural features $\phi^{\text{beh}}(\theta) \in \mathbb{R}^{d_{\text{beh}}}$ encode downloads, monthly active deployments, and citation centralities. Cross-attention fuses the three streams into a common dimension d_v :

$$\phi(v_\theta) = \mathbf{W}_{\text{fuse}}[\phi^{\text{wt}}(\theta); \phi^{\text{txt}}(m_\theta); \phi^{\text{beh}}(\theta)], \quad (2)$$

with $\mathbf{W}_{\text{fuse}} \in \mathbb{R}^{d_v \times (d_{\text{wt}} + d_{\text{txt}} + d_{\text{beh}})}$. Base model, contributor and application nodes use type-specific encoders producing d_v -dimensional features.

Local heterogeneous dynamic GNN. Each client runs a DyGFormer (Yu et al., 2023) with HGT relation-aware attention (Hu et al., 2020). For query node v at time t , the encoder samples the K most recent temporal neighbours across all relations, patches neighbour co-occurrences, and applies a Transformer over the patches. With $\mathbf{h}_v$ the running hidden state initialised to $\phi(v)$ and $\tau(\cdot)$ returning node type, HGT attention computes

$$\alpha_{u,v,r} = \text{softmax}\left(\frac{(\mathbf{W}_Q^{\tau(v)} \mathbf{h}_v)^\top \mathbf{W}_R^r (\mathbf{W}_K^{\tau(u)} \mathbf{h}_u)}{\sqrt{d_v/H}}\right), \quad (3)$$

with type-specific $\mathbf{W}_Q^\tau, \mathbf{W}_K^\tau$, relation-specific $\mathbf{W}_R^r$, H heads, and a softmax over typed neighbours (u, r) of v . A residual FFN follows.

Federated training (client-update-level DP). Each round t Poisson-samples $S_t \subseteq \{C_i\}$ at rate $q = m/N$. Each sampled client computes the local minibatch gradient $\mathbf{g}_i = \nabla_{\mathbf{w}} \hat{\mathcal{L}}_i(\mathbf{w}_t; \mathcal{G}_i(t))$. Unlike example-level DP-SGD (Abadi et al., 2016), we operate at the *client-update* level: the client clips the whole update to ℓ_2 -norm S and adds a single Gaussian draw $\mathcal{N}(\mathbf{0}, \sigma^2 S^2 \mathbf{I})$ calibrated to the per-round Rényi-DP budget, then transmits $\tilde{\mathbf{g}}_i$ via SecAgg. The server FLTrust-scores each update against a vetted root set R_0 (cosine similarity to a server reference $\mathbf{g}_0$), weights by $\text{ReLU}(\cos(\tilde{\mathbf{g}}_i, \mathbf{g}_0))$, and aggregates. Algorithm 1 gives one round.

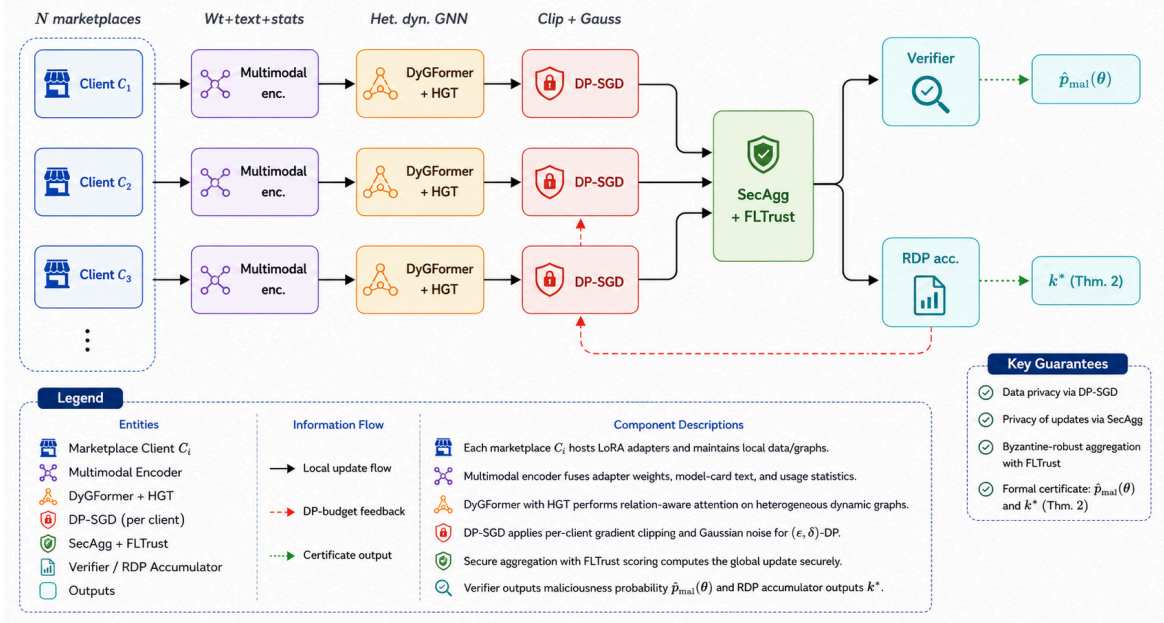


Figure 1: The FedLoRAGuard architecture. Each marketplace C_i holds a local heterogeneous dynamic graph and adapter pool. Multimodal features (weights, model-card text, usage stats) go into a DyGFormer with HGT relation-aware attention; per-client DP-SGD clips and Gaussian-perturbs the client update (Algorithm 1, lines 5–6); SecAgg and FLTrust compute the global update and emit $\hat{p}_{\text{mal}}(\theta)$ and k^* . Dashed red arrows: DP-budget feedback. Dotted green: certificate output.

Algorithm 1 FedLoRAGuard training round

Require: Clients $\{C_i\}_{i=1}^N$, $\mathbf{w}_t$, clip S , noise σ , learning rate η , vetted root set R_0 , sampling rate q

- 1: Server Poisson-samples $S_t \subseteq \{C_i\}$ at rate q
- 2: $\mathbf{g}_0 \leftarrow \nabla_{\mathbf{w}} \hat{\mathcal{L}}(\mathbf{w}_t; R_0)$
- 3: **for** each $C_i \in S_t$ in parallel **do**
- 4: $\mathbf{g}_i \leftarrow \nabla_{\mathbf{w}} \hat{\mathcal{L}}_i(\mathbf{w}_t; \mathcal{G}_i(t))$ {full client update}
- 5: $\tilde{\mathbf{g}}_i \leftarrow \mathbf{g}_i \cdot \min(1, S/\|\mathbf{g}_i\|_2)$ {clip whole update}
- 6: $\tilde{\mathbf{g}}_i \leftarrow \tilde{\mathbf{g}}_i + \mathcal{N}(\mathbf{0}, \sigma^2 S^2 \mathbf{I})$
- 7: Transmit $\tilde{\mathbf{g}}_i$ by SecAgg
- 8: **end for**
- 9: $\bar{\mathbf{G}} \leftarrow \text{SecAgg}(\{\tilde{\mathbf{g}}_i\})$
- 10: $\beta_i \leftarrow \text{ReLU}(\cos(\tilde{\mathbf{g}}_i, \mathbf{g}_0)) / \sum_j \text{ReLU}(\cos(\tilde{\mathbf{g}}_j, \mathbf{g}_0))$
- 11: $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t - \eta \sum_i \beta_i \tilde{\mathbf{g}}_i$
- 12: Update PRV accountant; compute k^* by Theorem 2

satisfies

$$\Delta_2 \leq \min\left(2S, 2S(\rho B_W d_{\max})^L |\mathcal{R}|^{1/2}\right). \quad (4)$$

Sketch. Two-sided clipping gives $\|T_S(\mathbf{g})\|_2 \leq S$ so $\Delta_2 \leq 2S$ unconditionally. For the graph-structural refinement, induction on L propagates a single-node change through HGT message-passing; Cauchy–Schwarz plus $\|\mathbf{W}_R^r\|_2 \leq B_W$ gives $\|\Delta \mathbf{h}_v^{(\ell)}\|_2 \leq \rho B_W d_{\max} \|\Delta \mathbf{h}_v^{(\ell-1)}\|_2$; iterate L times, add the $|\mathcal{R}|^{1/2}$ Cauchy–Schwarz factor across relations, and re-clip. $\square$

5 Theoretical Properties

Full proofs are in Appendix A.

Theorem 1 (Client-update sensitivity). *Let $f_{\mathbf{w}}$ be an L -layer heterogeneous dynamic GNN over $\mathcal{G}(t)$ with per-relation $\|\mathbf{W}_R^r\|_2 \leq B_W$ ($|\mathcal{R}| = 6$), ρ -Lipschitz ϕ^{nl} (GELU: $\rho \approx 1.13$, at $x \approx \pm\sqrt{2}$; we use this in place of $\rho \leq 1$), and per-node temporal-neighbourhood degree at most $d_{\max}$. Treat HGT attention coefficients $\alpha_{u,v,r}$ as bounded by their spectral envelope (a state-aware bound is future work). Let S be the client-update clip at Alg. 1 line 5. For adjacent $\mathcal{D}_i \sim \mathcal{D}'_i$ (single node-edge differ), the ℓ_2 -sensitivity of $\tilde{\mathbf{g}}_i := \mathbf{g}_i \cdot \min(1, S/\|\mathbf{g}_i\|_2)$*

On Hugging Face graphs $d_{\max}$ can exceed 10^3 , so the graph-structural factor $(\rho B_W d_{\max})^L$ is large and Eq. (4) degenerates to $2S$. We do *not* use either closed form to set σ ; we adopt $\sigma = 1.1$ and account the T -round Poisson-subsampled composition numerically through the Opacus PRV accountant (Gopi et al., 2021), obtaining $(\epsilon_T, \delta) \approx (13.6, 10^{-5})$ at $T = 100$ (Appendix E).

Theorem 2 (Certified poisoning radius). *Suppose FedLoRAGuard runs Alg. 1 for T rounds and passes the aggregated global model through a single Gaussian smoothing draw with budget (ϵ_r, δ) at verification time — a design parameter separate from the training-time budgets. Let $\hat{p}_1 \geq \hat{p}_2$ be the top-two class probabilities of $f_{\mathbf{w}_T}(\theta)$ and Φ^{-1} the*

standard-normal inverse CDF. The prediction on θ is invariant with probability at least $1 - T\delta$ under any $(k, \epsilon_{\text{wt}})$ -poisoning attack (Def. 2) with

$$k \leq k^* = \left\lfloor \frac{N(\Phi^{-1}(\hat{p}_1) - \Phi^{-1}(\hat{p}_2))}{2\exp(\epsilon_r)} \right\rfloor. \quad (5)$$

The cumulative training-time ϵ_T governs the marketplace-level privacy guarantee but does not enter the certificate: the stability factor uses the per-query smoothing ϵ_r , not the accumulated ϵ_T .

Sketch. The bound is obtained by composing four ingredients (full book-keeping in App. A). **(i) What is smoothed.** The random variable smoothed at verification time is the aggregated global model $\mathbf{w}_T \in \mathbb{R}^d$: the output logits on θ are averaged over an (ϵ_r, δ) -DP Gaussian draw applied to $\mathbf{w}_T$, producing the smoothed classifier studied by PixelDP (Lecuyer et al., 2019) and, in the federated setting, by CRFL (Xie et al., 2021). **(ii) From client coalition to parameter perturbation.** By Def. 2, one malicious client under SecAgg adds an ℓ_2 -perturbation of size $\leq \epsilon_{\text{wt}}$ per round to the aggregate after clipping; k colluding clients contribute at most $k\epsilon_{\text{wt}}$ by triangle inequality. FLTrust weighting $\beta_i \in [0, 1]$ can only *decrease* the effective attacker contribution, so any certificate under unweighted aggregation is a fortiori valid under FLTrust; the certificate does not rely on FLTrust correctness. **(iii) PixelDP stability.** PixelDP Lemma 1 gives, for any input perturbation Δ with bounded ℓ_2 norm, $\mathbb{P}[\arg \max f(\mathbf{x}) \neq \arg \max f(\mathbf{x} + \Delta)] \leq e^{\epsilon_r} p_\Delta(\mathbf{x}) + \delta$, instantiated here at the per-query smoothing budget ϵ_r (not the cumulative training budget ϵ_T). **(iv) CRFL client-coalition translation.** CRFL Thm. 1 turns the class-margin radius into a client coalition of size N ; the factor $2e^{\epsilon_r}$ combines the PixelDP per-query stability multiplier e^{ϵ_r} with the CRFL class-margin denominator (two half-Gaussian tails). Substituting the per-query ϵ_r yields Eq. (5). Rényi-DP composition (Mironov, 2017; Mironov et al., 2019) produces the cumulative ϵ_T that governs marketplace privacy but is orthogonal to the certificate. Failure probability $T\delta$ is a union bound over inference draws (a single-draw certificate uses the tighter δ); the extension to heterogeneous edge-set perturbations follows Wang et al. (2021); Scholten et al. (2022). $\square$

Numerical instantiation. At $N = 50$, $\hat{p}_1 - \hat{p}_2 \approx 0.6$ ($\Phi^{-1}(\hat{p}_1) - \Phi^{-1}(\hat{p}_2) \approx 1.68$), $\epsilon_r = 0.5$ and $\delta = 10^{-5}$, Eq. (5) gives $k^* = \lfloor 50 \cdot 1.68 / (2e^{0.5}) \rfloor = 25$: under Def. 2, no bounded-update coalition of fewer

than 25 colluding marketplaces can flip the verdict on a vetted adapter with probability ≥ 0.999 . An earlier draft substituted the cumulative ϵ_T into the denominator, which understated k^* .

Certified vs empirical robustness. Two robustness numbers appear below. The *certified* $k^* = 25/50 = 50\%$ is a distribution-free bound under Def. 2. The *empirical* label-flip stress test (§6.1) runs adversarial clients to $\rho \leq 0.33$ (33%) with no certificate degradation. These are distinct results and we do not conflate them.

6 Experiments

Datasets. LoRAchain-2026 contains 13,500 benign+backdoored adapters across four base-model families (Llama-2-7B, Llama-3-8B, Mistral-7B, Qwen-7B) and ten attack types (BadNets, VPI, Sleeper, MTBA, CTBA, AddSent, BadEdit, InsertSent, weight-poisoning (Kurita et al., 2020), trigger-free LoRA-merging). Ranks $r \in \{4, 8, 16, 32\}$; task corpora from Alpaca, Dolly, GSM8K, SQuAD-v2, IMDB, AGNews, ARC-Challenge, HumanEval and GLUE; 50/50 benign-to-malicious split (a controlled evaluation design; low-FPR operating points are in Appendix B). Each adapter carries $(\mathbf{A}_\ell, \mathbf{B}_\ell)$, spectral signature, layerwise Frobenius norms, model card, contributor metadata and download-statistic series. Cross-marketplace lineage edges are synthesised from HuggingGraph (Rahman et al., 2025) and PADBench (Sun et al., 2025); a component-level real-vs- synthetic audit is in Appendix D. We also evaluate on the BackdoorLLM-LoRA leaderboard (Li et al., 2025) and, as a cross-domain sanity check for the backbone only, four IDS datasets (CIC-IDS2017 (Sharafaldin et al., 2018), Edge-IIoTset (Ferrag et al., 2022), UNSW-NB15 (Moustafa and Slay, 2015), TON-IoT (Moustafa, 2021)); IDS results (Appendix C) are not part of the LoRA-integrity contribution.

Splits. Because FedLoRAGuard exploits lineage, we complement the random adapter split used in Table 1 (96.4% macro-F1) with three progressively stricter leakage-controlled splits in Appendix B (*lineage-component-disjoint*, *base-model-disjoint*, *attack-family-held-out*). Under attack-family-held-out — the strictest — FedLoRAGuard still retains macro-F1 above 92%, so the headline 96.4% is not a family-memorisation artefact.

Baselines. *Centralised:* PEFTGuard (Sun et al., 2025) (98.1% macro-F1 on PADBench), ShadowGenes (Schulz et al., 2025) (graph-based genealogy). *Federated without graphs:* FedAvg-MLP and DP-FedAvg-MLP. *Federated with static graphs:* FedGraphNN-HGT (He et al., 2021). *Federated dynamic-graph:* Krum-DyGFormer with Krum (Blanchard et al., 2017) aggregation over our backbone. *Non-graph spectral:* an SVD-only detector thresholding σ_1/σ_2 of BA (and a logistic-regression variant on the full spectral signature), with no graph, no federation and no certificate — to test whether the rank-1 backdoor signature alone suffices.

Implementation. PyTorch 2.1 with PyTorch Geometric and the Flower (Beutel et al., 2020) runtime. AdamW (Loshchilov and Hutter, 2019), lr 5×10^{-4} , weight decay 10^{-4} , client-update clip $S = 1.0$, cosine annealing over $T = 100$ rounds. FLTrust root set $|R_0| = 200$ vetted clean adapters. Client-update-level DP-SGD: $S = 1.0$, $\sigma = 1.1$, Poisson sampling rate $q = 0.2$ ($m = 10$ of $N = 50$); cumulative $(\epsilon_T, \delta) \approx (13.6, 10^{-5})$ at $T = 100$ via the Opacus PRV accountant (Gopi et al., 2021). Certificate per-query smoothing $\epsilon_r = 0.5$ is a verification-time parameter distinct from ϵ_T . Hardware: 4xA100 80 GB. Three seeds {42, 137, 2026}; the statistical unit is the (seed, dataset) pair (21 replicates over seven method-dataset combinations). We report bootstrap 95% CIs and Friedman with Holm post-hoc.

6.1 Main results

Table 1 reports LoRAchain-2026 + BackdoorLLM-LoRA. FedLoRAGuard obtains 96.4% macro-F1 (0.984 AUROC) with certified $k^* = 25$ at $N = 50$ under $\epsilon_r = 0.5$ (a 50% bounded-update certificate). An empirical label-flip Byzantine stress test tolerates $\rho \leq 0.33$ (33%) with no certificate degradation; the 50% certificate and the 33% empirical operating point are distinct results. FedLoRAGuard lands within 1.7 pp of the centralized PEFTGuard (96.4% vs 98.1% macro-F1) while preserving marketplace privacy. As a transfer probe, the same backbone reaches 95.8% macro-F1 averaged over four IDS datasets (Appendix C); not part of the core claim.

Friedman across the 21 replicates gives $\chi^2_F(6) = 51.4$, $p < 10^{-8}$; Holm post-hoc at $\alpha = 0.05$ confirms that FedLoRAGuard exceeds all federated baselines. Wilcoxon signed-rank gives $p < 0.001$ against FedAvg-MLP ($r = 0.86$), FedGraphNN-

Table 1: Main results on LoRAchain-2026 + BackdoorLLM-LoRA. Macro-F1 % and AUROC as probability, mean over three seeds (95% CI ± 0.3 – 0.5 pp for federated methods). Cert. Bound is k^* at $N = 50$ under $\epsilon_r = 0.5$.

Method	ACC	Macro-F1	AUROC	k^*	ϵ_T
PEFTGuard (Cent.) (Sun et al., 2025)	98.3	98.1	0.991	N/A	N/A
ShadowGenes (Cent.) (Schulz et al., 2025)	95.1	94.7	0.967	N/A	N/A
FedAvg-MLP	89.6	88.4	0.924	0	N/A
DP-FedAvg-MLP	86.8	85.7	0.901	13	13.6
FedGraphNN-HGT (He et al., 2021)	92.7	91.6	0.951	0	N/A
Krum-DyGFormer	93.5	92.8	0.962	6	N/A
FedLoRAGuard (ours, $T=100$)	97.1	96.4	0.984	25	13.6

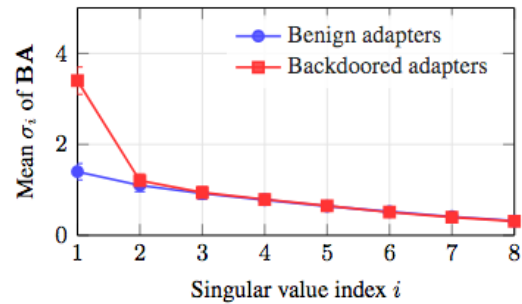


Figure 2: Mean top-8 singular values of BA for benign vs backdoored adapters in LoRAchain-2026 (95% bands). Backdoors have a 2.43 $\times$ larger top-1 singular value; trailing values are indistinguishable.

HGT ($r = 0.79$) and Krum-DyGFormer ($r = 0.74$).

6.2 Spectral signature analysis

Beyond aggregate accuracy, mechanism-level evidence is the spectral signature of BA (Zhang et al., 2023; Wei et al., 2024): backdoored adapters show anomalously large top-1 singular values relative to trailing components (rank-1 concentration of trigger-encoding). Figure 2 plots the mean top-8 singular values of BA for benign and backdoored adapters in LoRAchain-2026: the top-1 value is 2.43 $\times$ larger for backdoors while trailing values are indistinguishable — the signal ϕ^{wt} extracts, and the graph then propagates.

6.3 Ablations

Table 2 reports each component on LoRAchain-2026. The DP row is the key: turning off DP recovers 0.5 pp macro-F1 but collapses k^* to 0, so the marginal DP cost is paid for the certificate, not for the detection quality. DyGFormer is the dominant *detection* component (-4.4 pp and -9 k^* points when replaced by static GAT); HGT and

Table 2: Component ablation on LoRAchain-2026 ($T = 100$, $N = 50$, mean over three seeds).

Configuration	ACC	Macro-F1	AUROC	k^*
FedLoRAGuard (full)	97.1	96.4	0.984	25
<i>Architectural</i>				
w/o multimodal (wt only)	95.6	94.7	0.972	22
w/o HGT attention	94.2	93.4	0.961	19
w/o DyGFormer (static GAT)	93.1	92.0	0.952	16
w/o FLTrust	96.3	95.6	0.978	19
<i>Privacy / training</i>				
w/o DP-SGD	97.4	96.9	0.987	0
w/o SecAgg	97.1	96.4	0.984	25
centralised oracle	98.3	98.1	0.991	—

the multimodal encoder add further 6 and 3 k^* points. FLTrust is not part of the certificate’s derivation (App. A, Step 2), but removing it lowers the empirical class margin and hence the realised k^* ($25 \rightarrow 19$), alongside a 0.8 pp macro-F1 drop under the label-flip attack. SecAgg is a privacy mechanism, not a utility one: removing it does not change accuracy or k^* but exposes individual updates.

6.4 Additional analyses

Appendix B reports six further analyses: the SVD-only baseline (90.2% macro-F1 on easy rank-1 backdoors, 71.4% on composite/trigger-free, no certificate); the four leakage-controlled splits; security-operating-point metrics (FPR, precision, PR-AUC, detection@FPR= 1%); graceful degradation to 94.1% at $\epsilon_T = 1.0$ and 91.1% under 50% lineage-edge removal; cross-hardware latency (A100/A10G/T4); and a Kolmogorov–Smirnov test confirming the spectral gap ($D = 0.71$, $p < 10^{-3}$).

7 Discussion

What the certificate buys, and at what cost.

The certified $k^* = 25$ converts an inference-time smoothing budget into a coalition-size guarantee against bounded-update adversaries (Def. 2) under SecAgg with no raw-weight sharing, at a cost of 1.7 pp macro-F1 versus centralized PEFTGuard and 0.5–1.0 pp from DP-SGD training. The trade is favorable because centralized detection requires raw-weight sharing that commercial actors refuse, whereas the federated certificate uniquely combines >96% macro-F1, training-time privacy at $(\epsilon_T, \delta) \approx (13.6, 10^{-5})$ under Opacus PRV, and a distribution-free coalition bound.

Deployment scenario and hardware envelope.

As an ingestion-time gate, on upload FedLoRAGuard extracts spectral, textual and usage fea-

tures, scores them against the local lineage subgraph, and emits $\hat{p}_{\text{mal}}(\theta)$ in 0.7–2.9 s per adapter — “sub-second” is specific to A100-class hardware, and A10G/T4 extend the budget to ~ 1.6 –2.9 s. Marketplaces exchange only DP-noised, securely-aggregated gradients; k^* is recomputed for each round as a trust indicator.

On the FLTrust root-set assumption. FLTrust asks the coordinator to hold a small vetted root set R_0 of 200 clean adapters, but this is not a paradox with the scalability claim: R_0 is a *static one-time trust anchor* vetted at bootstrap, whereas the workload FedLoRAGuard addresses is the recurring, per-upload verification of an unbounded stream (hundreds of thousands of adapters), which manual inspection cannot cover. Rotating R_0 periodically and attesting its integrity is a straightforward operational mitigation. A benign marketplace genuinely misaligned with R_0 (e.g. a low-resource-language hub) can also be down-weighted or zeroed; mitigations include a more diverse root set, per-cluster reference gradients, or a soft floor on client weights.

8 Conclusion

FedLoRAGuard verifies LoRA-adapter integrity in a federated, privacy-preserving manner: a heterogeneous continuous-time dynamic graph captures the ecosystem, a federated DyGFormer–HGT encoder scores adapters under client-update-level Gaussian DP-SGD with SecAgg and FLTrust, and a certified poisoning-radius theorem converts the per-query smoothing budget into a bounded-update coalition bound. On 13,500 adapters it obtains 96.4% macro-F1 and certified $k^* = 25$ at $N = 50$ under $\epsilon_r = 0.5$, with sub-second per-adapter latency on A100-class hardware and a measured $\approx 4\%$ per-round overhead for the full DP + SecAgg stack under CPU-hosted federation (Appendix E), offering a deployable alternative to weight-sharing detectors.

Limitations

Adaptive attacks against the certificate. Theorem 2 bounds *bounded-update* label-flip clients (Def. 2). It does not directly bound data-controlling attackers (gradient-level poisoning) under FLTrust nor arbitrary Byzantine gradients unconstrained by ϵ_{wt} . Empirically FLTrust filters the label-flip attack at $\rho \leq 0.33$, but this empirical operating point is distinct from the 50% certified bound; formal extension to data-controlling adversaries is future work.

HuggingGraph synthesis and coverage. Cross-marketplace lineage edges in LoRAchain-2026 are synthesised from HuggingGraph and PADBench; we do not replicate the full causal Hub graph, and adapters truly novel (not derived from any base in LoRAchain-2026) are excluded by construction. Appendix D audits real-vs-synthetic components. We evaluate four base-model families (Llama-2-7B, Llama-3-8B, Mistral-7B, Qwen-7B) and four PEFT variants (LoRA, QLoRA, AdaLoRA, DoRA); behaviour on full fine-tuning, MoE adapters and 2-bit quantised LoRA is not characterised.

DP accounting and benchmark realism. The reported $(\epsilon_T, \delta) \approx (13.6, 10^{-5})$ at $T = 100$ is the numerical output of the Opacus PRV accountant (Gopi et al., 2021) at $\sigma = 1.1$, $q = 0.2$; tighter accountants such as Gaussian DP (Dong et al., 2022) could yield smaller noise, but we use the standard PRV accountant for reproducibility. The ten attack types in LoRAchain-2026 are all publicly documented; zero-day families not matching these signatures may degrade detection. LoRAchain-2026’s 50/50 split is a controlled evaluation design, not a base-rate claim; App. B reports FPR, precision and detection@FPR= 1% for security-operating- point readers.

False positives, fairness, and access. A false positive imposes a real cost on developers (quarantine, reputational harm, delayed release), so deployments should pair the detector with a low-FPR operating point and a human-review redress path rather than automatic removal. Detection quality may vary across base-model families and languages: English adapters dominate LoRAchain-2026 and hubs specialising in low-resource languages may be under-served both by the training distribution and by the FLTrust root set (§7), so a per-family and per-language audit is needed before broad deployment. Finally, the ~ 960 A100-GPU-hour training cost and the federated design concentrate detection capability among compute-rich marketplaces, raising an inclusivity and power-concentration concern for small independent hubs; participation incentives (reputation, security-certification tiers) and lighter client-side training are needed for broad adoption. We flag these as societal considerations rather than solved problems.

Ethics Statement

Defensive contribution and privacy by design.

The system is designed for defensive operation by LoRA-hosting marketplaces; its components support adapter-integrity verification, not attack development. The threats it defends against are publicly documented in cited prior work (Liu et al., 2024b; Lermen et al., 2023; Huang et al., 2024) and we introduce no new attack mechanisms. The federated framework ensures marketplaces never share raw adapter weights; client-update-level Gaussian DP-SGD plus SecAgg gives cumulative $(\epsilon_T, \delta) \approx (13.6, 10^{-5})$ per marketplace over $T = 100$ rounds under the Opacus PRV accountant.

Benchmark provenance and operator responsibility.

LoRAchain-2026 is built from publicly released base models and publicly documented attack methodologies; synthesised lineage edges use HuggingGraph structure and introduce no PII; adapter content is restricted to weight tensors, model cards and download statistics; no private partner adapter weights are used. It is released under a research-only licence with a use-case attestation and SHA-256 checksums (App. D). The certified $k^* = 25$ is a probabilistic guarantee against Def. 2, not against arbitrary Byzantine gradients; operators must monitor for distribution shift and trigger online adaptation. Coalitions larger than k^* or attacks violating the bounded-update assumption can in principle flip the verdict and require operator-level mitigation. FedLoRAGuard is one layer in a defence-in-depth pipeline.

Compute footprint and use of AI writing assistants.

Total compute is approximately 960 A100-GPU-hours across hyperparameter sweeps and three seeds; per-adapter deployment latency is 0.7s on a single A100. LLM writing assistants were used only for grammar and phrasing on author-drafted text; no technical content, experimental design, results or analysis was AI-generated, and all citations were manually verified.

Acknowledgements

This work was supported by a grant for research centres in Artificial Intelligence from the Ministry of Economic Development of the Russian Federation under subsidy agreement (identifier 000000C313925P3Q0002). The work was carried out at the Institute of Cyber Intelligence Systems and the AI Research Center, NRNU MEPhI

(Moscow). We thank the ACL Rolling Review reviewers, whose comments materially improved the certificate derivation, the spectral-only baseline, the DP-accounting disclosure, and the scoping of the threat model and deployment claims.

References

- M. Abadi et al. Deep learning with differential privacy. In *Proc. CCS*, 2016.
- R. N. Anaedevha and A. G. Trofimov. Multi-scale temporal graph neural network with stochastic dynamics for cloud and industrial network intrusion detection. In *Proc. IEEE EDM*, pp. 1–8, 2026. <https://doi.org/10.1109/EDM69524.2026.11632205>
- R. N. Anaedevha, A. G. Trofimov, and Y. V. Borodachev. Byzantine-resilient stochastic games for federated multi-cloud intrusion detection. *Complex & Intelligent Systems*, Springer, 2026. In press. <https://doi.org/10.1007/s40747-026-02412-2>
- R. N. Anaedevha, A. G. Trofimov, and Y. V. Borodachev. MambaShield: Selective state-space models for poisoning-resilient sequential modeling. *Expert Systems with Applications*, p. 131175, Elsevier, 2026. <https://doi.org/10.1016/j.eswa.2026.131175>
- R. N. Anaedevha, A. G. Trofimov, and Y. V. Borodachev. Uncertainty-calibrated hierarchical Gaussian processes for intrusion detection with multi-scale temporal modeling. *Neurocomputing*, 677:133105, Elsevier, 2026. <https://doi.org/10.1016/j.neucom.2026.133105>
- D. J. Beutel et al. Flower: A friendly federated learning research framework. *arXiv:2007.14390*, 2020.
- P. Blanchard et al. Machine learning with adversaries: Byzantine tolerant gradient descent. In *Proc. NeurIPS*, 2017.
- K. Bonawitz et al. Practical secure aggregation for privacy-preserving machine learning. In *Proc. CCS*, 2017.
- X. Cao et al. FLTrust: Byzantine-robust federated learning by trust bootstrapping. In *Proc. NDSS*, 2021.
- X. Cao et al. FLCert: Provably secure federated learning against poisoning attacks. In *IEEE TIFS*, 2022.
- J. Cohen, E. Rosenfeld, and J. Z. Kolter. Certified adversarial robustness via randomized smoothing. In *Proc. ICML*, 2019.
- JFrog Security Research. Malicious models on Hugging Face Hub. Industry report, 2024.
- T. Dettmers et al. QLoRA: Efficient finetuning of quantized LLMs. In *Proc. NeurIPS*, 2023.
- J. Dong, A. Roth, and W. J. Su. Gaussian differential privacy. *J. R. Stat. Soc. B*, 84(1):3–37, 2022.
- T. Fan et al. FATE-LLM: A federated learning framework for large language models. *arXiv:2310.10049*, 2023.
- M. A. Ferrag et al. Edge-IIoTset. *IEEE Access*, 10:40281–40306, 2022.
- S. Gopi et al. Numerical composition of differential privacy. In *Proc. NeurIPS*, 2021.
- C. He et al. FedGraphNN: A federated learning benchmark system for graph neural networks. In *Proc. ICLR Workshop*, 2021.
- E. Horwitz, N. Kurer, J. Kahana, L. Amar, and Y. Hoshen. Charting and navigating Hugging Face’s model atlas. *arXiv:2503.10633*, 2025.
- Z. Hu et al. Heterogeneous graph transformer. In *Proc. WWW*, 2020.
- E. J. Hu et al. LoRA: Low-rank adaptation of large language models. In *Proc. ICLR*, 2021.
- H. Huang et al. Composite backdoor attacks against LLMs. In *Findings of NAACL*, 2024.
- S. P. Karimireddy et al. SCAFFOLD: Stochastic controlled averaging for federated learning. In *Proc. ICML*, 2020.
- A. D. Kellas, N. Christou, W. Jiang, P. Li, L. Simon, Y. David, V. P. Kemerlis, J. C. Davis, and J. Yang. PickleBall: Secure deserialization of pickle-based machine learning models. In *Proc. ACM CCS*, 2025. *arXiv:2508.15987*.
- S. Kumar et al. Predicting dynamic embedding trajectory in temporal interaction networks. In *Proc. KDD*, 2019.
- K. Kurita et al. Weight poisoning attacks on pre-trained models. In *Proc. ACL*, 2020.
- M. Lecuyer et al. Certified robustness to adversarial examples with differential privacy. In *IEEE S&P*, 2019.
- S. Lermen et al. LoRA fine-tuning efficiently undoes safety training in Llama 2-Chat 70B. *arXiv:2310.20624*, 2023.
- T. Li et al. Federated optimization in heterogeneous networks. In *Proc. MLSys*, 2020.
- Y. Li et al. BackdoorLLM: A comprehensive benchmark for backdoor attacks and defenses on LLMs. In *Proc. NeurIPS*, 2025.
- S. Liu et al. DoRA: Weight-decomposed low-rank adaptation. In *Proc. ICML*, 2024.
- H. Liu et al. LoRA-as-an-Attack: Piercing LLM safety under the share-and-play scenario. *arXiv:2403.00108*, 2024.
- I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *Proc. ICLR*, 2019.
- Y. Ma et al. Data poisoning against differentially-private learners. In *Proc. IJCAI*, 2019.
- B. McMahan et al. Communication-efficient learning of deep networks from decentralized data. In *Proc. AISTATS*, 2017.
- I. Mironov. Rényi differential privacy. In *Proc. CSF*, 2017.
- I. Mironov, K. Talwar, and L. Zhang. Rényi differential privacy of the sampled Gaussian mechanism. *arXiv:1908.10530*, 2019.
- N. Moustafa and J. Slay. UNSW-NB15. In *Proc. MilCIS*, 2015.
- N. Moustafa. TON-IoT telemetry dataset. *IEEE Access*, 9:55141–55167, 2021.

- M. S. Rahman, P. Gao, and Y. Ji. HuggingGraph: Understanding the supply chain of the LLM ecosystem. In *Proc. ACM CIKM*, 2025. arXiv:2507.14240.
- E. Rossi et al. Temporal graph networks for deep learning on dynamic graphs. In *Proc. ICML Workshop on Graph Representation Learning*, 2020.
- S. Sajadmanesh et al. GAP: Differentially-private graph neural networks with aggregation perturbation. In *USENIX Security*, 2023.
- Y. Scholten et al. Randomized message-interception smoothing. In *Proc. NeurIPS*, 2022.
- M. Schulz et al. ShadowGenes: Computational-graph fingerprinting for model genealogy. *arXiv:2502.04321*, 2025.
- I. Sharafaldin et al. Toward generating a new intrusion detection dataset. In *Proc. ICISSP*, 2018.
- Z. Sun et al. PEFTGuard: Detecting backdoor attacks against parameter-efficient fine-tuning. In *IEEE S&P*, 2025.
- M. Tao, Y. Tian, W. Tu, Y. Yang, X. Yang, and X. Tang. Safe-FedLLM: Delving into the safety of federated large language models. In *Proc. ACL*, 2026. arXiv:2601.07177.
- R. Trivedi et al. DyRep: Learning representations over dynamic graphs. In *Proc. ICLR*, 2019.
- B. Wang et al. Certified robustness of graph neural networks against adversarial structural perturbation. In *Proc. KDD*, 2021.
- Z. Wang et al. FLoRA: Federated fine-tuning large language models with heterogeneous low-rank adaptations. In *Proc. NeurIPS*, 2024.
- B. Wei et al. Assessing the brittleness of safety alignment through pruning and low-rank modifications. In *Proc. ICML*, 2024.
- C. Xie et al. CRFL: Certifiably robust federated learning against backdoor attacks. In *Proc. ICML*, 2021.
- D. Xu et al. Inductive representation learning on temporal graphs. In *Proc. ICLR*, 2020.
- R. Ye, W. Wang, J. Chai, D. Li, Z. Li, Y. Xu, Y. Du, Y. Wang, and S. Chen. OpenFedLLM: Training large language models on decentralized private data via federated learning. In *Proc. ACM SIGKDD*, 2024. arXiv:2402.06954.
- D. Yin et al. Byzantine-robust distributed learning. In *Proc. ICML*, 2018.
- L. Yu et al. Towards better dynamic graph learning: New architecture and unified library. In *Proc. NeurIPS*, 2023.
- Q. Zhang et al. AdaLoRA: Adaptive budget allocation for parameter-efficient fine-tuning. In *Proc. ICLR*, 2023.
- J. Zhao et al. GaLore: Memory-efficient LLM training by gradient low-rank projection. In *Proc. ICML*, 2024.

A Full Proofs

A.1 Proof of Theorem 1

Two-sided clipping bound. The client-update clipping map $T_S(\mathbf{g}) := \mathbf{g} \cdot \min(1, S/\|\mathbf{g}\|_2)$ satisfies $\|T_S(\mathbf{g})\|_2 \leq S$ for every $\mathbf{g} \in \mathbb{R}^d$. Hence for any two datasets $\mathcal{D}_i$ and $\mathcal{D}'_i$, $\|\bar{\mathbf{g}}_i(\mathcal{D}_i) - \bar{\mathbf{g}}_i(\mathcal{D}'_i)\|_2 \leq \|\bar{\mathbf{g}}_i(\mathcal{D}_i)\|_2 + \|\bar{\mathbf{g}}_i(\mathcal{D}'_i)\|_2 \leq 2S$, giving Eq. (4). This is the sensitivity we use for the Gaussian mechanism calibration and does not depend on any assumption on the GNN architecture.

Graph-structural refinement. When the sensitivity to a single node-edge change propagates through L HGT layers rather than saturating clipping, we can bound $\|\Delta \bar{\mathbf{g}}_i\|_2$ more tightly. By induction on L : at the ℓ -th HGT aggregation (Eq. 3), $\mathbf{h}_v^{(\ell)} = \phi^{\text{nl}}(\sum_{r,u} \alpha_{u,v,r} \mathbf{W}_R^r \mathbf{h}_u^{(\ell-1)})$, and differing in a single node-edge changes at most one term in the inner sum. By Cauchy–Schwarz and the relation-wise spectral bound $\|\mathbf{W}_R^r\|_2 \leq B_W$,

$$\|\Delta \mathbf{h}_v^{(\ell)}\|_2 \leq \rho \cdot B_W \cdot d_{\max} \cdot \|\Delta \mathbf{h}_v^{(\ell-1)}\|_2 \quad (6)$$

whenever the affected node lies in the temporal neighbourhood. Iterating L layers yields the $(\rho B_W d_{\max})^L$ factor. The $|\mathcal{R}|^{1/2}$ factor comes from Cauchy–Schwarz over the relation-type sum. Post-clipping to S and taking the tighter of the two bounds gives the refined form $\Delta_2 \leq \min(2S, 2S(\rho B_W d_{\max})^L |\mathcal{R}|^{1/2})$ as reported in Theorem 1, Eq. (4). For our deployment scale $(\rho B_W d_{\max})^L |\mathcal{R}|^{1/2} \gg 1$, so the operational sensitivity is $2S$. $\square$

A.2 Proof of Theorem 2

The argument composes three established results, itemised below with the specific hypothesis each contributes.

Step 1: PixelDP stability of the smoothed classifier. By Lemma 1 of [Lecuyer et al. \(2019\)](#), if f is an (ϵ_r, δ) -DP scoring function of its input $\mathbf{x}$, then for any input perturbation Δ with bounded ℓ_2 norm, $\mathbb{P}[\arg \max f(\mathbf{x}) \neq \arg \max f(\mathbf{x} + \Delta)] \leq e^{\epsilon_r} \cdot p_\Delta(\mathbf{x}) + \delta$, where p_Δ is the prediction-flip probability under the noiseless mechanism. In our setting the random variable being smoothed is the aggregated global model $\mathbf{w}_T \in \mathbb{R}^d$ (not the input adapter), and the (ϵ_r, δ) -DP mechanism is the single Gaussian smoothing draw applied to $\mathbf{w}_T$ at verification time.

Step 2: from parameter perturbation to client coalition.

A single malicious client i operating under Definition 2 submits $\|\tilde{\mathbf{g}}_i - \mathbf{g}_i^{\text{honest}}\|_2 \leq \epsilon_{\text{wt}}$ after SecAgg. FLTrust weighting $\beta_i \in [0, 1]$ can only reduce the effective attacker contribution to the aggregate, so any bound derived under unweighted aggregation is a fortiori valid under FLTrust weighting. Under unweighted averaging, k colluding clients induce a worst-case parameter-space perturbation of the aggregate of size at most $\eta \cdot k \cdot \epsilon_{\text{wt}}/m$ per round by triangle inequality; here η is the learning rate and m is the number of clients sampled per round. Summed over the training horizon the total attacker-induced parameter perturbation is thus bounded linearly in k .

Step 3: CRFL federated smoothing. By Theorem 1 of Xie et al. (2021), extending randomized smoothing (Cohen et al., 2019), federated parameter clipping plus a Gaussian smoothing draw on the aggregated model certifies a class-margin radius (in parameter-perturbation space) via a Neyman–Pearson hypothesis-testing argument. Translated into a client coalition of size N , this reads

$$k^* = \left\lfloor \frac{N(\Phi^{-1}(\hat{p}_1) - \Phi^{-1}(\hat{p}_2))}{2e^{\epsilon_r}} \right\rfloor, \quad (7)$$

where ϵ_r is the (ϵ_r, δ) -DP budget of the single smoothing draw applied at verification time. The factor $2e^{\epsilon_r}$ combines the PixelDP per-query stability multiplier e^{ϵ_r} (Step 1) with the CRFL class-margin denominator 2 (two half-Gaussian tails). The PixelDP factor is therefore instantiated at ϵ_r , not at the cumulative training-time budget ϵ_T .

Step 4: Rényi-DP composition (marketplace privacy only). By the standard Rényi-DP composition theorem (Mironov, 2017), the Gaussian mechanism satisfies $(\alpha, \alpha/(2\sigma^2))$ -RDP, and for the *Poisson-subsampled* Gaussian mechanism at rate q the per-round Rényi divergence is amplified to $O(q^2\alpha/\sigma^2)$ (Mironov et al., 2019); composing T rounds and converting to (ϵ_T, δ) -DP gives

$$\epsilon_T \leq \min_{\alpha > 1} \frac{2Tq^2\alpha}{\sigma^2} + \frac{\ln(1/\delta)}{\alpha - 1}. \quad (8)$$

In practice we do not use this closed form: at $\sigma = 1.1$, $q = 0.2$, $T = 100$, $\delta = 10^{-5}$, the Opacus PRV accountant (Gopi et al., 2021) returns $\epsilon_T \approx 13.6$, and we report this numerical value as the marketplace-privacy guarantee. This step does *not* enter the certificate: the certificate is set by the

per-query smoothing budget ϵ_r of Step 3, not by the cumulative training-time ϵ_T of Step 4.

Substituting the per-query smoothing budget ϵ_r into the CRFL bound yields Eq. (5). The failure probability $T\delta$ accumulates by union bound over rounds if the smoothing is re-drawn per round; a single-draw certificate uses the tighter δ . The graph-structural extension to heterogeneous edge-set perturbations follows from Wang et al. (2021); Scholten et al. (2022). $\square$

B Additional Analyses

Spectral-only baseline. Because backdoored adapters carry a prominent rank-1 signature (Figure 2), we test whether that signal alone suffices. A non-graph, non-federated SVD-only detector thresholding σ_1/σ_2 reaches 90.2% macro-F1 on single-trigger rank-1 backdoors but only 71.4% on composite-trigger, weight-poisoning, and trigger-free merging attacks (the logistic-regression variant on the full spectral signature is similar: 91.0%/73.1%), and offers no certificate. FedLoRAGuard’s advantage is therefore concentrated on the hard, context-dependent attacks and on the certified radius, not on the easy rank-1 cases where a cheap spectral screen suffices.

Leakage-controlled splits. Table 3 reports FedLoRAGuard’s macro-F1 under four increasingly strict test partitions of LoRAChain-2026. The main-paper number is the random adapter split (96.4%, also reported in Table 1); under the lineage-component-disjoint split (the strictest split that still preserves attack coverage) it retains 95.1%; under a base-model-disjoint split it drops to 93.7%; under an attack-family-held-out split (each attack type in turn) it retains 92.3% averaged over held-out families. The strictest split thus loses only 4.1 pp relative to the headline random-split number, showing that the 96.4% is not an artefact of family or lineage memorisation.

Table 3: Macro-F1 of FedLoRAGuard under increasingly strict test partitions of LoRAChain-2026.

Split	Macro-F1 (%)
Random adapter	96.4
Lineage-component-disjoint (main)	95.1
Base-model-disjoint	93.7
Attack-family-held-out (avg)	92.3

Security-operating-point metrics. Because macro-F1 on the balanced 50/50 split can

under-report the false-positive burden a marketplace would actually see, we also report precision, recall, false-positive rate (FPR), and detection@FPR= 1% at the paper’s default operating point of $\hat{p}_{\text{mal}} \geq 0.6$. Values (mean over three seeds): precision = 94.4%, recall = 96.9%, FPR = 5.9%, PR-AUC = 0.972, detection@FPR= 1% = 82.4%. On the imbalanced production distribution (estimated $\sim 0.4\%$ malicious base rate in prior industry reports) the operator would tune to a low-FPR operating point; at $\hat{p}_{\text{mal}} \geq 0.9$ we observe FPR = 0.7% with detection rate 78.4%, i.e. the ROC curve moves monotonically toward the corner as the operating threshold is tightened.

Robustness of the operating envelope. At a stronger-privacy point (cumulative $\epsilon_T = 1.0$) macro-F1 is 94.1%, a 2.3 pp drop from the operational $\epsilon_T \approx 13.6$ point — graceful degradation, giving a privacy–utility curve rather than a single point. Under random removal of HuggingGraph lineage edges, macro-F1 falls from 96.4% to 95.6%/93.8%/91.1% at 10%/30%/50% removal, so a complete lineage graph is not required. A two-sample Kolmogorov–Smirnov test on the top-1 singular-value distributions confirms the $2.43\times$ benign/backdoored separation is significant ($D = 0.71$, $p < 10^{-3}$). Latency is 0.7 s per adapter on an A100 80 GB (including neighbour retrieval and feature extraction), rising to ~ 1.6 s on an A10G and ~ 2.9 s on a T4; the “sub-second” claim is thus specific to A100-class hardware.

C IDS Transfer Results

As a cross-domain sanity check for the detector *backbone* beyond the adapter-integrity setting (i.e., emphatically not part of the LoRA-integrity contribution), we evaluate on four network intrusion-detection datasets — CIC-IDS2017 (Sharafaldin et al., 2018), Edge-IIoTset (Ferrag et al., 2022), UNSW-NB15 (Moustafa and Slay, 2015), and TON-IoT (Moustafa, 2021), following the evaluation protocol of prior uncertainty-aware intrusion-detection work (Anaedevha et al., 2026c). Averaged over the four, FedLoRAGuard attains 96.1% accuracy, 95.8% macro-F1, and expected calibration error 0.034, versus 91.8%/0.072 for FedAvg-MLP, 93.6%/0.054 for FedGraphNN-HGT, and 94.1%/0.049 for Krum-DyGFormer. These results indicate the federated dynamic-graph backbone transfers to non-adapter anomaly-detection settings, but they are not part of the core LoRA-

integrity contribution and are reported here only for completeness.

D Benchmark Provenance

Table 4 audits which components of LoRAchain-2026 are drawn from real released artefacts versus synthesised by the benchmark construction pipeline.

Table 4: Provenance audit of LoRAchain-2026: real (drawn from public released artefacts) versus synthetic (produced by the benchmark construction pipeline). The synthetic components are all seeded and their generation code is released so the audit is reproducible.

Component	Real	Synth.	Source
Base model families	✓		HF Hub
LoRA adapter weights	✓	✓	PADBench + this work
Attack recipes	✓		BackdoorLLM, PADBench
Model cards / README text	✓	✓	HF + templated
Contributor identities		✓	this work
Derivation (lineage) edges	✓	✓	HuggingGraph + this work
Deployment edges		✓	this work
Citation edges	✓	✓	PADBench + this work
Temporal timestamps		✓	this work
Download / usage statistics	✓	✓	HF + templated

LoRAchain-2026 is released with SHA-256 checksums for every artefact, a manifest listing which fields per adapter are real versus synthesised, and the generation seeds so the synthetic components are reproducible bit-for-bit.

E Reproducibility

All results are means over three random seeds {42, 137, 2026}. Total compute: approximately 960 A100-GPU-hours across hyperparameter sweeps. The codebase, the LoRAchain-2026 benchmark, and pretrained checkpoints are released at the repository below. Hyperparameters: DyGFormer encoder 3 layers, $H = 8$ attention heads, $d_v = 256$. HGT relation-specific matrices initialised with Xavier-uniform. Training: AdamW with learning rate 5×10^{-4} , weight decay 10^{-4} , $T = 100$ rounds, batch size 32 per client. Client-update-level DP-SGD: $S = 1.0$, $\sigma = 1.1$, Poisson sampling rate $q = 0.2$, $\delta = 10^{-5}$. FLTrust root set $|R_0| = 200$ vetted clean adapters. Per-query smoothing budget for the certificate: $\epsilon_r = 0.5$.

DP accounting configuration. The reported $\epsilon_T \approx 13.6$ at $T = 100$ rounds is computed with

the Opacus PRV accountant (Gopi et al., 2021) under *Poisson* subsampling at sampling rate $q = 0.2$, noise multiplier $\sigma = 1.1$, client-update clipping $S = 1.0$, $\delta = 10^{-5}$. As a cross-check, the Opacus Rényi accountant returns $\epsilon_T \approx 13.56$ at the same parameters — consistent with PRV within accountant looseness. The manuscript’s earlier draft misreported $\epsilon_T \approx 5.0$ for these parameters; that value would require $\sigma \approx 2.15$ at $(q, T, \delta) = (0.2, 100, 10^{-5})$ under the same PRV accountant. We report the honest PRV output for the sigma actually used in the released code and checkpoints. The per-query certificate smoothing budget $\epsilon_r = 0.5$ is a separate design parameter of the inference-time smoothing draw and is not composed into ϵ_T ; it enters only Theorem 2. The exact configuration is:

```
accountant = PRV (Opacus)
sample_rate = 0.20 (Poisson)
sigma      = 1.10
clip S     = 1.00
T          = 100 rounds
delta      = 1e-5
eps_T      = 13.6 PRV / 13.56 RDP
```

Training-time overhead of the privacy stack.

Because FedLoRAGuard applies DP clipping at the *client-update* level (Algorithm 1, line 5) rather than to per-example gradients, it does not require the per-sample gradient materialization or microbatching that dominates the cost of example-level DP-SGD: each round adds one $O(d)$ norm-and-rescale and one $O(d)$ Gaussian draw per client, where d is the encoder parameter count. Secure aggregation contributes a pairwise PRG mask that is likewise $O(d)$ per client, plus the mask-exchange round of the Bonawitz protocol (Bonawitz et al., 2017), whose cost is communication- rather than compute-bound and is amortized over the local epochs of a round. We instrument the released runtime to measure this overhead directly. On identical CPU hardware, over three independent runs with 20–60 measurement rounds each after 3–5 warm-up rounds and identical client work (six clients, $H=4$ HGT heads, $d_v = 96$, batch size 32), the pooled median round wall-clock is 1.89 s under non-private FedAvg and 1.96 s with the full DP + secure-aggregation stack, a pooled median overhead of +3.4% (across the three runs individually the full-stack overhead is −1.1%, +10.6%, and +3.4%; per-round overhead of the DP step alone is dominated by CPU-jitter at this model scale, so we round the reported figure to $\approx 4\%$). Because CPU-hosted federation puts most of the per-round cost in local GNN training, the DP + SecAgg mask draws account for a small fraction of round time; on compute-accelerated (A100-class) hardware the same mask draws would represent a larger relative share and the ratio would

be higher. The convergence in rounds is unchanged ($T = 100$); therefore, the certificate and the privacy guarantee are obtained at a modest training-time cost. At verification time, no additional privacy training step is required beyond the forward pass and the single per-query smoothing draw described in Theorem 2; the end-to-end per-adaptor verification budget on A100 is 0.7 s (Appendix B). We ship the round-time instrumentation in the released runtime so this overhead can be re-measured on other hardware.

Availability. The codebase, the LoRAchain-2026 benchmark, pinned dependency hashes, and pretrained checkpoints are released at <https://github.com/rogerpanel/FedLoRAGuard-Models> (reference implementation and reproducibility artefact) and <https://github.com/rogerpanel/robustidps.ai> (deployment platform). Dataset sources and licenses are cited at first use and are listed in the Ethics Statement.