The Missing Sampler: Distributed Privacy Accounting for Asynchronous Federated Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

Differentially private federated learning (DP-FL) typically reports a system-level privacy guarantee under the assumption that clients participate uniformly. Buffered asynchronous FL violates this assumption: the server aggregates the fastest responders, so low-latency clients participate disproportionately often. As a result, fast clients can accumulate up to 97× the reported privacy loss, while slow clients consume little of their allocated budget. This creates a protocol-driven privacy disparity that is invisible to standard accounting. We introduce a distributed privacy-accounting framework for buffered asynchronous DP-FL. The framework repurposes the persistent committee nodes of modern one-shot secure aggregation protocols as privatisers. The committee maintains per-client Gaussian-DP ledgers, enforces retirement through a safety cap, injects distributed noise, and threshold-decrypts only authorised aggregates. The same committee also draws a uniformly random k'-of-K subset from each latency-biased buffer using public randomness, accomplishing for the first time privacy amplification by subsampling in asynchronous DP-FL. Experiments on CIFAR-10, AG News, and DBpedia show that committee subsampling recovers up to +38 percentage points of accuracy at matched privacy budget, while per-client tracking is necessary for safety: even under the strongest amplification setting, standard accounting still violates the target privacy budget by at least $1.4\times$. A parameter-space canary audit confirms that the framework removes the disparate vulnerability induced by latency-biased participation.

1 Introduction

Federated learning (FL) trains a shared model across many devices without centralising user data [McMahan et al., 2018b].[1] To bound what the trained model can reveal about any single client, production FL systems add calibrated noise via differential privacy (DP) [Abadi et al., 2016, Kairouz et al., 2021a], while secure aggregation (SecAgg) ensures that the server observes only aggregate updates rather than individual client contributions [Bonawitz et al., 2017]. The resulting privacy guarantee is usually reported by a system-wide accountant as an (ε, δ)-DP bound, under the assumption that every client participates with the same probability in each aggregation. This assumption is natural in synchronous FL, but it breaks in buffered asynchronous FL: the server closes an aggregation once the first K responses arrive [Bonawitz et al., 2019, Nguyen et al., 2022], so low-latency clients enter the buffer disproportionately often. A noise scale calibrated to the average participation rate is therefore wrong at the client level. Fast clients accumulate privacy loss beyond $\varepsilon_{\text{target}}$ (under-privatisation), while slow clients consume little of their allocated budget and contribute less signal to the aggregate (over-privatisation).

[1] Code: https://anonymous.4open.science/r/AsyncDDP-66CB/.

Submitted to 40th Conference on Neural Information Processing Systems (NeurIPS 2026). Do not distribute.

The disparity is not merely an accounting artifact. Device capability, availability windows, and user behaviour are correlated [Bonawitz et al., 2019, Eichner et al., 2019, Cho et al., 2022], so latency-biased participation can align with population subgroups. Uniform accounting then hides a form of disparate vulnerability [Kulynych et al., 2022]: the clients who participate most often are also the clients least protected by the reported guarantee.

A natural repair is to track privacy loss per client and retire clients approaching their budget. Enforcement cannot be left to the server in our threat model (§5.3): a malicious server could under-report fast clients' counts, keep them past their budget, or pick the aggregated subset adversarially. Clients cannot self-report either. Both ledger enforcement and buffer-level randomisation must therefore be performed by a party separate from the server.

We provide this separation using the persistent committee already present in modern one-shot SecAgg protocols [Bell-Clark et al., 2025, Karthikeyan and Polychroniadou, 2025, Taiello et al., 2025]. We repurpose these committee nodes as *privatisers*: they observe only aggregate metadata, maintain per-client Gaussian-DP ledgers [Dong et al., 2022], enforce retirement through a safety cap, inject distributed noise, and authorise decryption only for budget-compliant aggregates. The same committee also supplies the missing randomisation step. Using a public randomness beacon [Ma et al., 2023, Cloudflare, 2024], it draws a uniform k'-subset $\mathcal{S}_t$ from the latency-biased buffer $\mathcal{B}_t$ and authorises decryption only of that partial aggregate (Figure 1). No single party observes individual updates: clients submit encrypted updates, the server receives only the noised aggregate over $\mathcal{S}_t$, and privatisers never see plaintext client contributions.

Together, these mechanisms address the two failures introduced by buffered asynchrony: the ledger enforces per-client privacy under heterogeneous participation, while committee subsampling restores a known inclusion probability at the buffer-to-aggregate step. Our contributions are as follows. First, we formalise the *under- and over-privatisation* problem in buffered asynchronous DP-FL, showing that latency-biased participation makes the true per-client privacy loss diverge from the system-level value reported by uniform accounting. We validate the effect with scale simulations and a parameter-space canary audit (§§4, 7.2, 7.2). Second, we introduce a privatiser-committee architecture for distributed per-client privacy accounting inside one-shot SecAgg. The committee maintains the privacy ledger, applies a hard safety cap, retires near-exhausted clients, and threshold-decrypts only authorised noised aggregates, we prove that any safety-capped spending policy enforces $\varepsilon_i \leq \varepsilon_{\text{target}}$ for every client (§§5.1, 6). Third, we prove that committee-drawn k'-of-K subsampling restores user-level subsampling amplification at the buffer-to-aggregate step (§5.2), making the buffer size K a deployment parameter for the wall-clock-versus-noise trade-off. Finally, we evaluate the framework on CIFAR-10, AG News, and DBpedia, showing that committee subsampling recovers most of the utility lost to DP noise while per-client tracking remains necessary for safety: naïve accounting still violates $\varepsilon_{\text{target}}$ by at least $1.4\times$ even under the strongest amplification setting (§7.3).

2 Related work

Distributed differential privacy in FL. In FL, the natural privacy unit is the *user*: each client's entire local dataset should be protected [McMahan et al., 2018b]. User-level DP can be enforced in several ways, which differ primarily in where noise is added and how much the server must be trusted. In *local* DP [Kasiviswanathan et al., 2011], each client fully perturbs its update before transmission, requiring no server trust but yielding aggregate noise that grows as $\sqrt{n}/\varepsilon$ rather than $1/\varepsilon$, and hence poor utility. In *central* DP [McMahan et al., 2018b], a trusted server collects raw updates, adds noise once to the aggregate, and achieves tight guarantees, but the server sees every individual gradient, enabling reconstruction attacks [Zhu et al., 2019]. *Distributed* DP [Kairouz et al., 2021a] represents the middle ground: each client adds a *share* of discrete Gaussian noise to its clipped update, and SecAgg [Bonawitz et al., 2017] ensures the server sees only the noised sum, preventing gradient inversion while achieving the same noise level as central DP §3.2. DP-FTRL [Kairouz et al., 2021b] relaxes this at the cost of central trust (Appendix E).

Individualized privacy and disparity. A parallel line of work pursues *individualized* privacy: Feldman and Zrnic [2021] introduce Rényi-DP filters that track per-data-point consumption and retire exhausted points, and Boenisch et al. [2023] propose IDP-SGD, assigning heterogeneous ε budgets per sample. Both operate at the *sample* level in centralised training with controlled Poisson subsampling. We transfer the same principle to the *user* level in distributed FL, where participation

heterogeneity is an inherent consequence of device speed, not a design choice. More broadly, DP can create or amplify disparities: Farrand et al. [2020] show that data imbalance interacts with DP noise to produce disparate accuracy, and Kulynych et al. [2022] formalise *disparate vulnerability*, proving that subgroup membership inference attack success equals the distributional generalisation gap. These works identify data-driven sources of disparity, whereas we identify a *protocol-driven* one: asynchronous participation heterogeneity creates privacy disparity even when data is IID, invisible under standard accounting.

Async FL and the subsampling-amplification gap. Buffered asynchronous aggregation [Nguyen et al., 2022, Fraboni et al., 2023] improved throughput over synchronous FL but introduced latency-dependent participation heterogeneity, with participation rates varying by orders of magnitude in production [Kairouz et al., 2021c, Mohammadi et al., 2025]. This heterogeneity also breaks the workhorse of DP-SGD and synchronous DP-FedAvg [Abadi et al., 2016, McMahan et al., 2018a]: privacy amplification by subsampling [Kasiviswanathan et al., 2011, Balle et al., 2018, Mironov, 2017, Wang et al., 2019] requires each unit to be sampled with a known uniform probability q, but the buffer fills with whichever clients finish first, breaking both the known-probability and round-to-round-independence prerequisites. Prior work works around this rather than solving it: downgrade to local DP (DP-FedBuff [lrd, 2025], FLAME [Liu et al., 2024], async-DP edge [Lu et al., 2020]), replace Poisson with windowed self-sampling (random check-ins [Balle et al., 2020, Girgis et al., 2021], one participation per window), or drop amplification and substitute correlated noise via tree aggregation or matrix factorisation [Kairouz et al., 2021b, Choquette-Choo et al., 2023b,a, McMahan and Xu, 2023], hence requiring server trust. To our knowledge, no prior work delivers q^2-Gaussian amplification at the buffer-to-aggregate step in the streaming buffered setting (where fast clients contribute repeatedly across rounds), none of these works tracks per-client privacy cost either. We close both gaps with a committee-driven k'-of-K subsample of the latency-biased buffer (§5.2, Appendix I), recovering the standard subsampled-Gaussian rate at a tunable wall-clock cost.

3 Preliminaries

3.1 Federated learning and buffered aggregation

We consider a federation of n clients coordinated by a central server. Client i holds a local dataset $\mathcal{D}_i$ defining a local objective $F_i(\theta)$, and the federation jointly minimises $F(\theta) := (1/n)\sum_{i=1}^{n} F_i(\theta)$. Training proceeds in T *aggregations* indexed by $t \in \{0, \dots, T\}$ (what the synchronous literature calls *rounds*, we use "aggregation" throughout since in asynchronous FL no global synchronisation barrier exists). Here t counts aggregation events, not wall-clock time, so a slow client's local computation may span several values of t.

Following FedBuff [Nguyen et al., 2022], at aggregation t the server forms a buffer $\mathcal{B}_t \subseteq [n]$ of the first $K := |\mathcal{B}_t|$ clients to respond. We let $\tau_t \in \mathbb{R}_{>0}$ denote the wall-clock duration of aggregation t (the latency of the K-th responder), so $T_{\text{wall}} := \sum_{t=0}^{T-1} \tau_t$ is the total federation time. Each client $i \in \mathcal{B}_t$ runs local SGD starting from a copy of the global model, clips the result to ℓ_2-sensitivity C [Abadi et al., 2016], and stochastically rounds it to an integer vector $\Delta_i \in \mathbb{Z}^d$ (required for compatibility with modular arithmetic in SecAgg over $\mathbb{Z}_{\mathfrak{q}}$ [Canonne et al., 2020], where $\mathfrak{q}$ is the SecAgg field modulus, we reserve q for the DP subsampling rate $q = k'/K$). Submission follows a SecAgg protocol [Bonawitz et al., 2017, Bell-Clark et al., 2025, Karthikeyan and Polychroniadou, 2025]: any instantiation suffices, provided the server receives only the modular sum of the aggregated subset $\sum_{i\in\mathcal{S}_t} \Delta_i \pmod{\mathfrak{q}}$ and cannot decompose it into individual contributions (concrete instantiations in Appendix F.2). Here $\mathcal{S}_t \subseteq \mathcal{B}_t$ is the *aggregated subset*, of size $k' := |\mathcal{S}_t| \le K$. Under standard buffered async FL all of $\mathcal{B}_t$ is aggregated ($\mathcal{S}_t = \mathcal{B}_t$, $k' = K$), §5.2 introduces a committee-subsampled $\mathcal{S}_t \subset \mathcal{B}_t$ with $k' < K$ on which the per-aggregation noise is committed. The server then updates $\theta_{t+1} = \theta_t + (1/k') \cdot \mathrm{Dequant}(\sum_{i\in\mathcal{S}_t} \Delta_i)$.

3.2 Distributed differential privacy

We work in the Gaussian-DP framework [Dong et al., 2022], in which a mechanism is μ-GDP if its hypothesis-testing tradeoff function is dominated by that of the test between $\mathcal{N}(0,1)$ and $\mathcal{N}(\mu,1)$. The privacy parameter is additive in squares under composition, and converts to a standard (ε,δ)-DP

guarantee through a closed-form expression in μ^2 and δ (Eq. 26, Appendix H). We measure all per-aggregation privacy costs in μ^2 throughout.

To realise a μ-GDP aggregation without a trusted server, the *distributed discrete Gaussian* mechanism (DDP) [Kairouz et al., 2021a] distributes the noise-sampling step across clients. The continuous-equivalent aggregate noise variance is $\sigma^2 = C^2/\mu^2$, contributed jointly by the k' clients in $\mathcal{S}_t$: each samples a discrete-Gaussian share of variance σ^2/k' (in continuous units), and the sum's privacy loss is dominated by that of a single central draw $\mathcal{N}(0, \sigma^2)$ [Kairouz et al., 2021a], giving the central-Gaussian-equivalent privacy guarantee. Integer-space sampling and the $(\Lambda/C)^2$ scaling are detailed in Appendix F. Under uniform participation $q = K/n$ over T aggregations, the system-level cost composes additively: $\mu^2_{\text{sys}} = Tq\mu^2$. The committee-subsampling refinement of §5.2 replaces this with $q = k'/K$ at the buffer-to-aggregate step via fixed-size sampling without replacement: every client in the buffer is included in the aggregated subset with marginal probability exactly q, recovering the standard subsampled-Gaussian amplification rate [Wang et al., 2019] at a wall-clock cost set by K.

A defining property of DDP under uniform-participation accounting is that the per-aggregation noise level σ^2 is *constant*: it is fixed before training from $(\varepsilon_{\text{target}}, \delta, T, q)$ and held across all T aggregations. This follows from client-side noise commitment: each client i samples its share η_i when preparing its update, before the buffer $\mathcal{B}_t$ has formed at the server. Neither the client nor any other party observes the buffer composition at that point, so the per-client variance σ^2/k' is determined ahead of time from the marginal sampling rate q, not from the realised buffer. Throughout we write σ^2 (no subscript) for this fixed DDP noise, §4 shows how the assumption fails under asynchronous participation, and §5 replaces it with a per-aggregation σ^2_t. We reserve $\mu^2_{i,t}$ for client i's *true* cumulative GDP cost under actual (non-uniform) participation and μ^2_{budget} for the per-client bound the protocol enforces ($\varepsilon_{\text{target}}$ is the corresponding (ε, δ)-DP bound).

4 The under- and over-privatisation problem

4.1 Latency-driven participation

Let $N_i := |\{t \in [T] : i \in \mathcal{B}_t\}|$ denote client i's actual participation count, with rate $q_i := \mathbb{E}[N_i]/T$. Each client's rate is determined by its speed: client i has base rate $\lambda_i = e^{X_i}$ with $X_i \sim \mathcal{N}(0, \sigma^2_{\text{lat}})$ ($\sigma_{\text{lat}} \approx 1.5$ is consistent with the order-of-magnitude device-speed ratios reported by Mohammadi et al. [2025] on heterogeneous edge testbeds and the qualitative production observations of Kairouz et al. [2021c]) and per-computation service time $\text{Exp}(1/\lambda_i)$. Buffered asynchronous selection picks the K fastest of n responders, inducing $q_i \approx K\lambda_i / \sum_j \lambda_j$, heavy-tailed: the fastest q_i exceeds the slowest by orders of magnitude.

4.2 Privacy disparity under uniform accounting

Standard composition assumes uniform participation $q = K/n$ over T aggregations and reports a single system-level GDP cost $\mu^2_{\text{sys}} = Tq\mu^2$ and a single $(\varepsilon_{\text{sys}}, \delta)$-DP guarantee shared by every client. Under asynchronous participation this no longer holds: client i's true cumulative cost after t aggregations is $\mu^2_{i,t} = N_{i,t} \cdot \mu^2$, where $N_{i,t}$ scales with q_i, not with q. The fixed σ^2 calibrated to q is then simultaneously too large for stragglers and too small for fast clients, silently exceeding $\varepsilon_{\text{target}}$.

Definition 1 (Under- and over-privatisation)**.** *Client i is* under-privatised *at aggregation t if $\mu^2_{i,t} > \mu^2_{sys,t}$, and* over-privatised *if $\mu^2_{i,t} \ll \mu^2_{sys,t}$. Equivalently, any client with $N_{i,t} > qt$ is under-privatised and any with $N_{i,t} \ll qt$ is over-privatised, since (ε, δ) is monotone in μ^2, the same inequalities hold for the per-client $\varepsilon_{i,t}$.*

The size of this violation is governed by the latency-heterogeneity scale σ_{lat}: at $\sigma_{\text{lat}} = 0$ participation is uniform and $\mu^2_{i,t} = \mu^2_{\text{sys},t}$ by construction, while heavy-tailed selection inflates the fastest client's true ε_i above the reported ε_{sys} at every federation size (§7.2, Appendix B). Overcoming the violation requires the calibrated noise to depend on the buffer $\mathcal{B}_t$ and on each client's remaining budget, which is impossible under DDP since noise is committed before the buffer is known (§3.2) [Kairouz et al., 2021c].

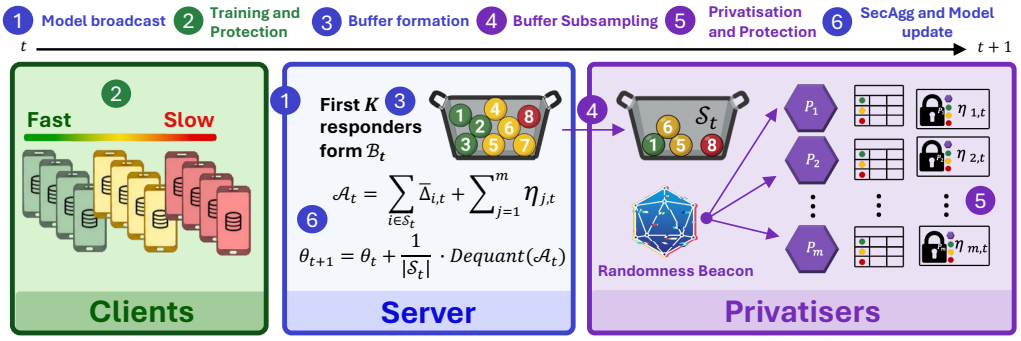

Figure 1: **One aggregation of the privatiser framework. (1)** server broadcasts θ_t; **(2)** clients train, clip, quantise, submit as soon as they are finished; **(3)** server forms $\mathcal{B}_t$ from the first K responders,**(4)** privatisers derive $\mathcal{S}_t \subseteq \mathcal{B}_t$ uniformly via the public beacon; **(5)** privatisers commit σ_t from the ledger and emit noise shares η_j; **(6)** server SecAgg-sums the $\mathcal{S}_t$-restricted partial aggregate and updates the model.

5 Method

The framework has four parts: a privatiser committee that relocates noise generation to nodes observing $\mathcal{B}_t$ *after* client submission (§5.1), a uniform random subset $\mathcal{S}_t \subseteq \mathcal{B}_t$ drawn from a public beacon, recovering rigorous subsampling amplification at the buffer-to-aggregate step (§5.2), a per-client budget plus safety cap that enforce $\varepsilon_i \leq \varepsilon_{\text{target}}$ under any spending policy (§5.3), and the threat model the construction operates under (§5.3).

5.1 Privatiser architecture

The under- and over-privatisation problem of §4.2 traces to a single architectural constraint: under DDP, the per-aggregation noise σ^2 must be committed before $\mathcal{B}_t$ is formed because each client samples its share at update-prep time. To make σ^2 depend on $\mathcal{B}_t$ and on per-client remaining budgets, we relocate noise generation to a committee of m *privatiser* nodes, repurposed from the persistent committees of one-shot SecAgg protocols Bell-Clark et al. [2025], Karthikeyan and Polychroniadou [2025] (Appendix F.2). The committee observes $\mathcal{B}_t$ *after* all K submissions arrive and only then commits σ_t^2 as a function of the buffer's per-client spent budgets.

Each privatiser j samples an integer-space noise share $\eta_{j,t} \sim \mathcal{N}_{\mathbb{Z}}(0, \tilde{\sigma}_t^2/m)$ with $\tilde{\sigma}_t^2 = (\Lambda/C)^2\sigma_t^2$ (Appendix F) and submits it through the SecAgg client-contribution interface. By Kairouz et al. [2021a], the privacy loss of $\sum_j \eta_{j,t}$ is dominated by that of a single central draw $\mathcal{N}(0, \sigma_t^2) = \mathcal{N}(0, (qC)^2/\mu_t^2)$ (Eq. (1), μ_t^2 is the delivered post-amplification GDP parameter, §5.3, $q = k'/K$): the privacy guarantee is identical to DDP, but σ_t^2 is chosen *after* observing $\mathcal{B}_t$. Each privatiser maintains the per-client ledger $\{\mu^2_{\text{spent},i}\}$ used by the spending policy (§5.3) to compute μ_t^2. After the committee subsamples $\mathcal{S}_t \subseteq \mathcal{B}_t$ (§5.2), retirement filters out near-exhausted clients to form the active aggregated subset $\mathcal{S}'_t \subseteq \mathcal{S}_t$ on which the spending policy and safety cap operate. Figure 1 shows one aggregation; Algorithm 2 (Appendix F) gives the pseudocode.

5.2 Committee-based amplification by subsampling

To our knowledge, no prior work delivers q^2-Gaussian amplification at the buffer-to-aggregate step of buffered async DP-FL (§2). The server fills a buffer $\mathcal{B}_t$ of size $K \geq k'$ from the first-to-arrive clients. Before any decryption is executed, the privatiser committee draws a uniformly random k'-subset $\mathcal{S}_t \subseteq \mathcal{B}_t$ and helps the server to reconstruct only $\sum_{i \in \mathcal{S}_t} \hat{\Delta}_i$, the remaining $K - k'$ ciphertexts stay sealed. From the privacy adversary's view, any client $i \in \mathcal{B}_t$ appears in the released aggregate with probability exactly $q = k'/K$, independent of i's identity or latency, so the standard subsampled-Gaussian analysis applies and the per-aggregation noise shrinks by q:

$$\sigma_t^2 \;=\; \frac{(C\,q)^2}{\mu_t^2} \;=\; \frac{(k'\,C/K)^2}{\mu_t^2}, \qquad q = k'/K. \tag{1}$$

K is a deployment parameter: $K = k'$ recovers the un-amplified mechanism, while $K = n$ is the synchronous endpoint, recovering DP-FedAvg's $q = k'/n$ amplification rate at the price of waiting for the n-th order statistic of latencies (the slowest client) every round. Any $K < n$ remains genuinely asynchronous: the round closes as soon as K ciphertexts arrive, so latency-tail clients can fail to finish without blocking progress.

Source of randomness. Following Ma et al. [2023], the privatiser committee obtains the per-round seed v_t either through a trusted source of randomness or by using a randomness beacon that is already deployed, such as Cloudflare's *drand* [Cloudflare, 2024]. Each privatiser derives $\mathcal{S}_t \leftarrow \textsc{ChooseSet}(v_t, \mathcal{B}_t, k')$ locally and deterministically. Full protocol pseudocode and the optional per-round rotation extension are deferred to Appendix F.2.

5.3 Per-client budget and safety cap

Per-client budget. Throughout this section and the rest of the paper, μ_t^2 denotes the *delivered, post-amplification* per-aggregation GDP parameter at step t: the amount charged to the per-client ledger when client i lands in $\mathcal{S}_t$. Equivalently, the underlying Gaussian mechanism's base parameter is μ_t/q, and committee subsampling at rate $q = k'/K$ amplifies it to μ_t (§5.2, Theorem 3). Each client is assigned a budget μ^2_{budget}, obtained by inverting the GDP-to-(ε, δ) conversion of §3.2 at the deployment target $(\varepsilon_{\text{target}}, \delta)$. The privatiser ledger tracks $\mu^2_{\text{spent},i} = \sum_t \mu_t^2 \cdot \mathbf{1}\{i \in \mathcal{S}_t\}$ and enforces $\mu^2_{\text{spent},i} \le \mu^2_{\text{budget}}$ at all times.

Retirement and safety cap. Two mechanisms enforce the per-client invariant. First, *retirement*: any client with remaining budget below $\theta_{\text{ret}} \cdot \mu^2_{\text{budget}}$ (default $\theta_{\text{ret}} = 0.05$) is dropped from the committee subsample $\mathcal{S}_t$ before the aggregate is decrypted, leaving the active aggregated subset $\mathcal{S}'_t \subseteq \mathcal{S}_t$. Second, a *hard cap* on the per-aggregation spend:

$$\mu_t^2 \;\leftarrow\; \min\Big(\mu_t^2,\; \min_{i \in \mathcal{S}'_t} \big(\mu^2_{\text{budget}} - \mu^2_{\text{spent},i}\big)\Big). \tag{2}$$

The safety guarantee (Theorem 2) holds for any choice of μ_t^2 paired with retirement and the cap.

Spending policies. We use FIXED, $\mu_t^2 = \mu^2_{\text{budget}}/(TK/n)$ constant with per-client tracking and retirement, as the default safe policy. UNSAFE is the current production practice [Kairouz et al., 2021a]: the same constant μ_t^2, but *without* per-client tracking, so the single ε_{sys} reported by the accountant assumes $\mathbb{E}[N_i] = TK/n$ uniformly and is silently violated under heterogeneous participation (§4.2). The architecture is not specific to constant μ_t^2: because the privatiser ledger exposes $(\mu^2_{\text{spent},i}, \hat{q}_i)$ for every $i \in \mathcal{S}'_t$, μ_t^2 can be chosen as any function of the active buffer, with per-client compliance to $\varepsilon_{\text{target}}$ following from the safety cap (Theorem 2) regardless of the function. Buffer-aware spending policies, not available in prior architectures, are therefore a separable design axis we leave to future work; Appendix J reports a preliminary exploration.

Threat model Following committee-assisted one-shot SecAgg protocols [Bell-Clark et al., 2025, Karthikeyan and Polychroniadou, 2025, Taiello et al., 2025], we assume a *static* malicious adversary that corrupts the server and up to a γ_p fraction of the privatiser committee at protocol start (adaptive corruption is out of scope). Privatisers receive $\mathcal{B}_t$ from the server and sign $(t, H(\mathcal{B}_t))$ for consistency [Bonawitz et al., 2017], derive $\mathcal{S}_t$ from the public beacon [Cloudflare, 2024], and then add discrete-Gaussian noise via the SecAgg protection. To tolerate up to $\gamma_p m$ faulty privatisers, each honest privatiser samples $\eta_{j,t} \sim \mathcal{N}_{\mathbb{Z}}(0, \sigma_t^2/\lceil(1-\gamma_p)m\rceil)$. The empirical cost of this m/t inflation on AG News is $\sim$2 pp at the honest-majority floor under amplification ($K \ge 2k'$), rising to $\sim$4.5 pp without amplification (Appendix F.5, Figure 3), amplification thus also offsets the cost of the trust regime. We defer to Ma et al. [2023] for details about privatiser committee selection. Privacy is enforced at the user level under the standard *add/remove* adjacency convention [McMahan et al., 2018a, Dwork and Roth, 2014]: (ε_i, δ) bounds the adversary's advantage between datasets differing in client i's entire $\mathcal{D}_i$ (Theorem 2, replacement convention absorbed as a $\sqrt{2}$ rescaling, Appendix A.2). Poisoning and denial-of-service attacks are outside the scope of this work.

6 Privacy and convergence guarantees

This section establishes three formal guarantees of the framework. *(i)* Per-client privacy holds under any spending policy, delivered by the hard safety cap of §5.3. *(ii)* The committee subselection of §5.2 delivers rigorous user-level subsampling amplification at rate $q = k'/K$. *(iii)* Standard convergence holds with the amplified σ_t (Appendix A.5).

6.1 Per-client privacy

Theorem 2 (Per-client DP under any spending policy)**.** *For every client $i \in [n]$ and aggregation $t \in [T]$, the model θ_t is $(\varepsilon_{i,t}, \delta)$-DP with respect to $\mathcal{D}_i$ with $\varepsilon_{i,t} \leq \varepsilon_{target}$, and $(\max_i \varepsilon_{i,t}, \delta)$-DP with respect to the full dataset.*

Sketch. The hard safety cap $\mu_t^2 \leftarrow \min(\mu_t^2, \min_{i \in \mathcal{S}_t'}(\mu^2_{\text{budget}} - \mu^2_{\text{spent},i}))$ prevents any aggregation from pushing an active client past its budget, per-client DP follows by adaptive GDP composition over i's participation aggregations (non-participating ones are post-processing). The bound holds for any σ_t produced by the spending policy, including the amplified $\sigma_t = (k'C/K)^2/\mu_t^2$ of (1). Full proof in Appendix A.

6.2 Amplification by committee subselection

Theorem 3 (Subsampled-Gaussian amplification)**.** *Fix any aggregation t and any buffer $\mathcal{B}_t$ with $|\mathcal{B}_t| = K \geq k'$. Let $q = k'/K$, and let $\mathcal{S}_t$ be a uniformly random k'-subset of $\mathcal{B}_t$, drawn from the committee's joint randomness independently of $\mathcal{B}_t$. Let the released aggregate be $A_t = \sum_{i \in \mathcal{S}_t} \hat{\Delta}_i + \eta_t$ with $\eta_t \sim \mathcal{N}_{\mathbb{Z}}(0, \sigma_t^2)$ and $\sigma_t = qC/\mu_t$ (Eq. (1); μ_t is the delivered post-amplification GDP parameter, §5.3). Then conditional on $\mathcal{B}_t$, A_t is μ_t-GDP with respect to any single client's data, and adaptive composition over T aggregations gives the system-level cost $\mu^2_{sys} \leq T\mu^2_{\max}$ with $\mu^2_{\max} = \max_t \mu_t^2$.*[2]

Sketch. Conditional on $\mathcal{B}_t$, the committee's joint randomness is independent of the data and yields a uniformly random k'-subset, so each client $i \in \mathcal{B}_t$ appears in $\mathcal{S}_t$ with probability exactly q. The post-amplification mechanism is therefore the subsampled discrete Gaussian at marginal rate q, whose μ_t-GDP guarantee derives from the fixed-size-without-replacement analysis of Wang et al. [2019]. The choice of $\mathcal{B}_t$ is data-dependent (latency-driven) but cancels in the conditional bound. Full proof in Appendix A.3.

6.3 Convergence

The framework inherits the standard convergence rate of clipped DP-SGD [Abadi et al., 2016, Koloskova et al., 2022]: under L-smoothness, bounded SGD variance σ^2_{sgd}, and bounded gradient heterogeneity Γ^2, the gradient-norm bound is

$$\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\|\nabla F(\theta_t)\|^2 \;\leq\; \mathcal{O}\left(\sqrt{\frac{L(F_0 - F^*)\,\bar{V}}{T}}\right), \quad \bar{V} = \tfrac{1}{T}\sum_t \left(\tfrac{\sigma^2_{\text{sgd}}}{k'} + \Gamma^2 + \tfrac{d(k'C/K)^2}{(k')^2\mu_t^2}\right), \tag{3}$$

where the third term in $\bar{V}$ is the DP noise variance under the amplified σ_t of Eq. (1): it shrinks by $q^2 = (k'/K)^2$ relative to the un-amplified bound, recovering the standard subsampled-Gaussian convergence rate.

7 Experiments

We structure our evaluation around three claims: the privacy violation under naïve accounting is real and severe (§7.2), it produces empirically measurable harm (§7.2), and our framework eliminates both the violation and the utility cost of enforcing safety (§7.3).

[2]The per-aggregation μ_t-GDP claim is the central-limit form of the fixed-size-without-replacement Rényi DP bound of Wang et al. [2019], converted to GDP via the Rényi-to-GDP map of Dong et al. [2022]; see Appendix A.3 for the explicit RDP→GDP bridge.

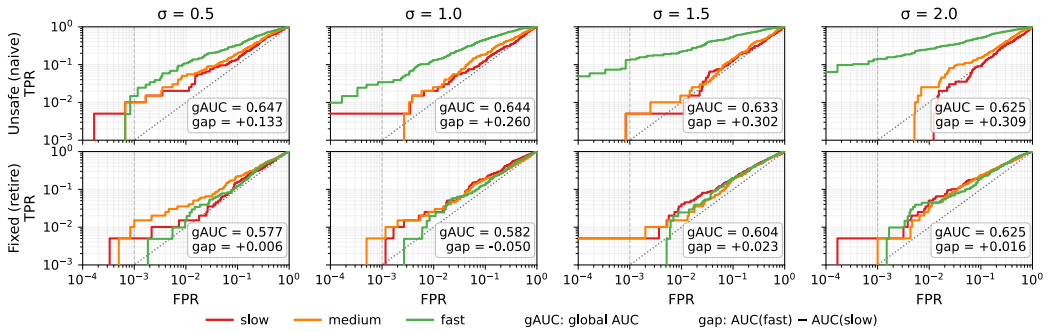

Figure 2: Per-tertile canary attack ROC across spending policies (rows) and latency heterogeneity σ_{lat} (columns). Each panel reports global AUC and disparity gap $\Delta\text{AUC} = \text{AUC}(\text{fast}) - \text{AUC}(\text{slow})$. Setup (Appendix D): parameter-space cosine attack [Andrew et al., 2024] on CIFAR-10, n=1000, k'=100, T=600, $\varepsilon_{\text{target}}$=10, K_{can}=200, S=2000, 3 seeds.

7.1 Setup

We organise the evaluation around three claims. *(i) Scale simulations* (§7.2) quantify the privacy violation under naïve accounting from $n = 20$ up to $n = 10^6$ via privacy-only GDP composition (no model training), making the result model- and dataset-independent. *(ii) A parameter-space canary audit* (§7.2) measures empirical disparate memorisation on CIFAR-10 ($n = 1000$, $k' = 100$, $T = 600$, $\varepsilon_{\text{target}} = 10$, $K_{\text{can}} = 200$, 3 seeds) across spending policies and four heterogeneity levels $\sigma_{\text{lat}} \in \{0.5, 1.0, 1.5, 2.0\}$. *(iii) Utility experiments* (§7.3) train models on AG News [Zhang et al., 2015] and DBpedia [Lehmann et al., 2015] (frozen DistilBERT linear heads) and CIFAR-10 [Krizhevsky, 2009] (frozen ResNet-18 linear head) at $n = 100$, $k' = 10$, $\delta = 10^{-5}$, server momentum 0.9, sweeping the buffer ratio $K \in \{k', 2k', 4k', n\}$ to characterise the wall-clock-versus-amplification trade-off. Architectures, ε targets, the latency-correlated Dirichlet construction, and extended $\varepsilon \in \{5, 10, 15\}$ tables are in Appendices G and I.

7.2 Privacy violation

Table 1 quantifies the under- and over-privatisation problem at $\sigma_{\text{lat}} = 1.5$, the regime documented in production cross-device deployments [Kairouz et al., 2021c, Mohammadi et al., 2025]. Stragglers use as little as 2–4% of their budget, the fastest client's true ε grows to 97× the target at production scale. The full $\sigma_{\text{lat}} \in \{0.5, 1.0, 1.5, 2.0\}$ sweep (Appendix B, Table 3) shows the violation factor growing monotonically and roughly quadratically with σ_{lat} at every scale, peaking at 192× (Gboard, $\sigma_{\text{lat}} = 2.0$). Both safe policies cap $\max_i \varepsilon_i \leq \varepsilon_{\text{target}}$ at every cell.

Table 1: Privacy violation under naïve async accounting ($\varepsilon_{\text{target}} = 8$, $\sigma_{\text{lat}} = 1.5$). Full sweep across $\sigma_{\text{lat}} \in \{0.5, 1.0, 1.5, 2.0\}$ in Appendix B, Table 3.

	n	T	Under-priv. (fast)		Over-pr.
			Max ε	Factor	Min bgt
Toy	20	50	11.8	1.5×	4%
Small	100	200	29.8	3.7×	5%
Med.	1K	500	35.5	4.4×	2%
Large	10K	1K	169.6	21×	10%
Prod.	100K	2K	363.5	45×	10%
Gboard	1M	5K	**774**	**97**×	20%

This is not merely an accounting error. Following Andrew et al. [2024]'s parameter-space canary construction (per-canary direction-projection, latency-tertile stratification, details in Appendix D), Figure 2 shows the privatiser architecture closes the disparate-vulnerability gap: under UNSAFE, the fast–slow ΔAUC grows from +0.13 to +0.31 as heterogeneity increases, while every safe policy reduces the gap.

7.3 Wall-clock and amplification trade-off

The buffer ratio K is the deployment parameter: larger K shrinks σ by $q = k'/K$ but delays the round to the K-th client's arrival. Table 2 reports end-of-training accuracy under FIXED across $K \in \{k', 2k', 4k', n\}$ at three privacy targets, mean ± std across 3 seeds. Accuracy rises monotonically with K (+33, +24, +38 pp on CIFAR-10 / AG News / DBpedia from $K = k'$ to $K = n$ at $\varepsilon = 10$),

Table 2: **Wall-clock ↔ amplification trade-off** under the safe FIXED spending policy. Left block: $n=100$, $k'=10$, mean ± std over 3 partitions × 3 latency heterogeneities × 3 seeds. Right block: $n=1000$, $k'=100$, $\sigma_{\text{lat}}=1.5$, Dir(0.5) correlated, single cell, 3 seeds. Wall-clock is total simulation time at $T_{\max}$ rounds, normalised per dataset to the $K=k'$ baseline. Full results with extended ε in Appendix I.

			Acc. (n=100)				Acc. (n=1000)			
Dataset	K	q	wc/wc_0	ε=5	ε=10	ε=15	wc/wc_0	ε=5	ε=10	ε=15
CIFAR-10	k'	1.00	1.0×	0.26 ± 0.02	0.32 ± 0.04	0.37 ± 0.04	1.0×	0.25 ± 0.02	0.34 ± 0.04	0.40 ± 0.04
	$2k'$	0.50	1.8×	0.34 ± 0.04	0.43 ± 0.05	0.47 ± 0.04	1.2×	0.37 ± 0.04	0.47 ± 0.05	0.54 ± 0.04
	$4k'$	0.25	4.6×	0.45 ± 0.04	0.53 ± 0.03	0.58 ± 0.03	3.3×	0.50 ± 0.04	0.61 ± 0.03	0.66 ± 0.03
	n	0.10	18×	0.59 ± 0.02	0.65 ± 0.01	0.68 ± 0.01	16×	0.67 ± 0.02	0.75 ± 0.01	0.78 ± 0.01
AG News	k'	1.00	1.0×	0.48 ± 0.07	0.56 ± 0.08	0.60 ± 0.07	1.0×	0.58 ± 0.07	0.65 ± 0.08	0.69 ± 0.07
	$2k'$	0.50	1.8×	0.58 ± 0.07	0.65 ± 0.06	0.68 ± 0.05	1.5×	0.66 ± 0.07	0.74 ± 0.06	0.78 ± 0.05
	$4k'$	0.25	3.7×	0.67 ± 0.05	0.72 ± 0.04	0.74 ± 0.03	2.8×	0.76 ± 0.05	0.82 ± 0.04	0.84 ± 0.03
	n	0.10	15×	0.76 ± 0.02	0.80 ± 0.02	0.82 ± 0.02	15×	0.84 ± 0.02	0.86 ± 0.02	0.87 ± 0.02
DBpedia	k'	1.00	1.0×	0.39 ± 0.03	0.56 ± 0.06	0.67 ± 0.04	1.0×	0.63 ± 0.03	0.76 ± 0.05	0.82 ± 0.04
	$2k'$	0.50	2.5×	0.60 ± 0.06	0.76 ± 0.07	0.82 ± 0.05	1.6×	0.78 ± 0.05	0.89 ± 0.07	0.93 ± 0.05
	$4k'$	0.25	4.3×	0.79 ± 0.04	0.86 ± 0.02	0.90 ± 0.02	4.2×	0.91 ± 0.04	0.96 ± 0.02	0.97 ± 0.02
	n	0.10	19×	0.90 ± 0.02	0.94 ± 0.02	0.96 ± 0.01	20×	0.97 ± 0.02	0.98 ± 0.02	0.98 ± 0.01

wall-clock grows super-linearly because the K-th order statistic is dominated by the slowest of K heavy-tailed latencies. Intermediate $K \in \{2k', 4k'\}$ is the operating regime: at $K=4k'$ the operator captures 74–84% of the maximum gain at $\sim 4\times$ rather than $\sim 17\times$ the no-amplification wall-clock cost. $K=n$ is the synchronous endpoint, included as a reference upper bound.

Per-client safety is unaffected by amplification. At $\varepsilon_{\text{target}}=10$ under UNSAFE, the fastest client's true ε exceeds the target by 1.4–1.5× even at $K=n$ and by 2.8–3.0× at $K=k'$ (Table 1, Appendix B). Amplification reduces σ uniformly but does not equalise participation, only the privatiser ledger (§5.1) closes the per-client gap. The two mechanisms are complementary, neither sufficient alone. On AG News in the configuration we test, committee subsampling at $K \geq 2k'$ also exceeds DP-FTRL: averaged over partitions, latency heterogeneities, and 3 seeds, $K=2k'$ exceeds DP-FTRL by 5–7 pp at every ε, and $K=n$ by 18–26 pp (Appendix E, Table 4).

7.4 Amplification offsets the cost of trust-robustness

The committee tolerates up to a γ_p fraction of faulty privatisers by inflating each share's variance by $1/(1-\gamma_p)$ (§5.3), equivalently, under threshold-cryptography with threshold t, the aggregate carries an m/t noise multiplier whether or not collusion occurs. Figure 3 shows the utility cost on AG News at $\varepsilon_{\text{target}}=10$, $\sigma_{\text{lat}}=1.5$, Dir(0.5). At the honest-majority floor $t=6$ the accuracy drop is ~ 4.5 pp without amplification, shrinking to ~ 2 pp at $K=2k'$. Larger K reduces the absolute extra noise that the m/t multiplier introduces, so deployments paying for collusion-resistance recover most of that cost through amplification. Privacy is identical at every t.

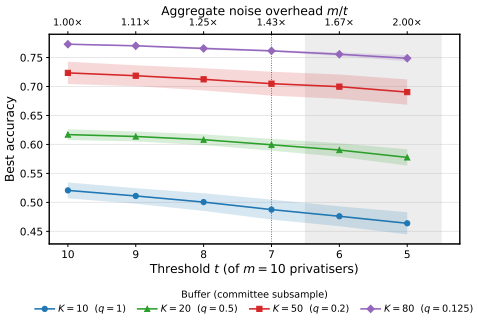

Figure 3: AG News accuracy vs. threshold t for $m=10$, swept across $K \in \{k', 2k', 5k', 8k'\}$. Vertical line: honest-majority floor ($t=6$).

8 Discussion and conclusion

Buffered async FL violates the uniform-participation assumption underlying DP accounting, and the violation tracks subgroup membership when device speed correlates with demographics. A single privatiser committee closes both gaps: a per-client ledger enforces $\varepsilon_i \leq \varepsilon_{\text{target}}$ under any spending policy, and a uniform k'-of-K subsample of the latency-biased buffer recovers user-level subsampling amplification. Limitations, broader impact, and open directions (buffer-aware spending, staleness-aware carry-over, per-round committee rotation [Ma et al., 2023]) are in Appendix K.

References

Local differential privacy for federated learning with fixed memory usage and per-client privacy. *arXiv:2510.12908*, 2025.

Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *CCS*, 2016.

Galen Andrew, Peter Kairouz, Sewoong Oh, Alina Oprea, H. Brendan McMahan, and Vinith M. Suriyakumar. One-shot empirical privacy estimation for federated learning. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2302.03098.

Marshall Ball, James Bell-Clark, Adria Gascon, Peter Kairouz, Sewoong Oh, and Zhiye Xie. Secure stateful aggregation: A practical protocol with applications in differentially-private federated learning. *arXiv:2410.11368*, 2024.

Borja Balle, Gilles Barthe, and Marco Gaboardi. Privacy amplification by subsampling: Tight analyses via couplings and divergences. In *NeurIPS*, 2018.

Borja Balle, Peter Kairouz, H. Brendan McMahan, Om Thakkar, and Abhradeep Thakurta. Privacy amplification via random check-ins. In *NeurIPS*, 2020.

James Bell-Clark, Adria Gascon, Baiyu Li, Mariana Raykova, and Phillipp Schoppmann. Willow: Secure aggregation with one-shot clients. In *CRYPTO*, 2025.

Franziska Boenisch, Christopher Mühl, Adam Dziedzic, Roy Rinberg, and Nicolas Papernot. Have it your way: Individualized privacy assignment for DP-SGD. In *NeurIPS*, 2023.

Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In *CCS*, 2017.

Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konečný, Stefano Mazzocchi, H. Brendan McMahan, Timon Van Overveldt, David Petrou, Daniel Ramage, and Jason Roselander. Towards federated learning at scale: A system design. In *MLSys*, 2019.

Clément L. Canonne, Gautam Kamath, and Thomas Steinke. The discrete Gaussian for differential privacy. In *NeurIPS*, 2020.

Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. Membership inference attacks from first principles. In *IEEE S&P*, 2022.

Yae Jee Cho, Jianyu Wang, and Gauri Joshi. Client selection in federated learning: Convergence analysis and power-of-choice selection strategies. In *AISTATS*, 2022.

Christopher A. Choquette-Choo, Arun Ganesh, Ryan McKenna, H. Brendan McMahan, Keith Rush, Abhradeep Thakurta, and Zheng Xu. (amplified) banded matrix factorization: A unified approach to private training. In *NeurIPS*, 2023a.

Christopher A. Choquette-Choo, H. Brendan McMahan, Keith Rush, and Abhradeep Thakurta. Multi-epoch matrix factorization mechanisms for private machine learning. In *ICML*, 2023b.

Cloudflare. Cloudflare randomness beacon (drand / league of entropy). https://developers.cloudflare.com/randomness-beacon/, 2024.

Ronald Cramer, Ivan Damgård, and Jesper Buus Nielsen. *Secure Multiparty Computation and Secret Sharing*. Cambridge University Press, 2015.

Sergey Denisov, H. Brendan McMahan, John Rush, Adam Smith, and Abhradeep Guha Thakurta. Improved differential privacy for SGD via optimal private linear prefix sums. In *SODA*, 2022.

Jinshuo Dong, Aaron Roth, and Weijie J. Su. Gaussian differential privacy. *Journal of the Royal Statistical Society: Series B*, 84(1):3–37, 2022.

Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3–4):211–407, 2014.

Hubert Eichner, Tomer Koren, H. Brendan McMahan, Nathan Srebro, and Kunal Talwar. Semi-cyclic stochastic gradient descent. In *ICML*, 2019.

Tom Farrand, Fatemehsadat Mireshghallah, Sahib Singh, and Andrew Trask. Neither private nor fair: Impact of data imbalance on utility and fairness in differential privacy. In *Workshop on Privacy in Machine Learning (PRIML), NeurIPS*, 2020.

Vitaly Feldman and Tijana Zrnic. Individual privacy accounting via a Rényi filter. In *NeurIPS*, 2021.

Yann Fraboni, Richard Vidal, Laetitia Kameni, and Marco Lorenzi. A general theory for federated optimization with asynchronous and heterogeneous clients updates. *Journal of Machine Learning Research*, 24(110):1–43, 2023.

Antonious Girgis, Deepesh Data, Suhas Diggavi, Peter Kairouz, and Ananda Theertha Suresh. Shuffled model of differential privacy in federated learning. In *AISTATS*, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

Dzmitry Huba, John Nguyen, Kshitiz Malik, Ruiyu Zhu, Mike Rabbat, Ashkan Yousefpour, Carole-Jean Wu, Hongyuan Zhan, Pavel Ustinov, Harish Srinivas, Kaikai Wang, Anthony Shoumikhin, Jesik Min, and Mani Malek. PAPAYA: Practical, private, and scalable federated learning. In *MLSys*, 2022.

Peter Kairouz, Sewoong Oh, and Pramod Viswanath. The composition theorem for differential privacy. In *International Conference on Machine Learning (ICML)*, 2015.

Peter Kairouz, Ziyu Liu, and Thomas Steinke. The distributed discrete Gaussian mechanism for federated learning with secure aggregation. In *ICML*, 2021a.

Peter Kairouz, H. Brendan McMahan, Shuang Song, Om Thakkar, Abhradeep Thakurta, and Zheng Xu. Practical and private (deep) learning without sampling or shuffling. In *ICML*, 2021b.

Peter Kairouz, H. Brendan McMahan, et al. Advances and open problems in federated learning. *Foundations and Trends in Machine Learning*, 14(1–2):1–210, 2021c.

Harish Karthikeyan and Antigoni Polychroniadou. OPA: One-shot private aggregation with single client interaction and its applications to federated learning. In *CRYPTO*, 2025.

Shiva Prasad Kasiviswanathan, Homin K. Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826, 2011.

Anastasia Koloskova, Sebastian U. Stich, and Martin Jaggi. Sharper convergence guarantees for asynchronous SGD for distributed and federated learning. In *NeurIPS*, 2022.

Antti Koskela, Joonas Jälkö, and Antti Honkela. Computing tight differential privacy guarantees using FFT. In *AISTATS*, 2020.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Bogdan Kulynych, Mohammad Yaghini, Giovanni Cherubin, Michael Veale, and Carmela Troncoso. Disparate vulnerability to membership inference attacks. In *PETS*, 2022.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian Bizer. DBpedia – a large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web*, 6 (2):167–195, 2015.

Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of FedAvg on non-IID data. In *ICLR*, 2020.

Ruiqi Liu et al. FLAME: Differentially private federated learning in the shuffle model. In *USENIX Security*, 2024.

Yunlong Lu, Xiaohong Huang, Yueyue Dai, Sabita Maharjan, and Yan Zhang. Differentially private asynchronous federated learning for mobile edge computing in urban informatics. *IEEE Transactions on Industrial Informatics*, 2020. Preprint: arXiv:1912.07902, 2019.

Yiping Ma, Jess Woods, Sebastian Angel, Antigoni Polychroniadou, and Tal Rabin. Flamingo: Multi-round single-server secure aggregation with applications to private federated learning. In *IEEE Symposium on Security and Privacy (S&P)*, 2023.

H. Brendan McMahan and Zheng Xu. Federated learning of Gboard language models with differential privacy. In *ACL (Industry)*, 2023.

H. Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. In *ICLR*, 2018a.

H. Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. In *ICLR*, 2018b.

Ilya Mironov. Rényi differential privacy. In *CSF*, 2017.

Samaneh Mohammadi, Iraklis Symeonidis, Ali Balador, and Francesco Flammini. Empirical analysis of asynchronous federated learning on heterogeneous devices: Efficiency, fairness, and privacy trade-offs. In *IJCNN*, 2025.

Milad Nasr, Jamie Hayes, Thomas Steinke, Borja Balle, Florian Tramèr, Matthew Jagielski, Nicholas Carlini, and Andreas Terzis. Tight auditing of differentially private machine learning. In *USENIX Security*, 2023.

John Nguyen, Kshitiz Malik, Hongyuan Zhan, Ashkan Yousefpour, Mike Rabbat, Mani Malek, and Dzmitry Huba. Federated learning with buffered asynchronous aggregation. In *AISTATS*, 2022.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv:1910.01108*, 2019.

Riccardo Taiello, Melek Önen, Clémentine Gritti, and Marco Lorenzi. Buffalo: A practical secure aggregation protocol for buffered asynchronous federated learning. In *CODASPY*, 2025.

Jianyu Wang, Zachary Charles, Zheng Xu, Gauri Joshi, H. Brendan McMahan, et al. A field guide to federated optimization. *arXiv:2107.06917*, 2022.

Yu-Xiang Wang, Borja Balle, and Shiva Kasiviswanathan. Subsampled Rényi differential privacy and analytical moments accountant. In *AISTATS*, 2019.

Cong Xie, Sanmi Koyejo, and Indranil Guha. Asynchronous federated optimization. *arXiv:1903.03934*, 2019.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *NeurIPS*, 2015.

Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. In *NeurIPS*, 2019.

A Full proofs

A.1 Budget compliance

Theorem 4 (Budget compliance)**.** *Under any spending policy, for all clients $i \in [n]$ and all aggregations $t \in [T]$:*

$$\mu^2_{spent,i,t} \;=\; \sum_{t' \le t,\, i \in \mathcal{S}'_{t'}} \mu^2_{t'} \;\le\; \mu^2_{budget}.$$

Proof. We show that $\mu^2_{\text{spent},i,t} \le \mu^2_{\text{budget}}$ for all clients $i \in [n]$ and all aggregations $t \in [T]$, under any spending policy that applies the safety cap below.

The proof proceeds by induction on aggregations. At $t=0$, $\mu^2_{\text{spent},i}(0) = 0 \le \mu^2_{\text{budget}}$ for all i. Assume the invariant holds at aggregation t. We show it holds at $t+1$.

The safety cap. Every safe spending policy applies a hard cap after computing μ_t^2 via its aggregation function:

$$\mu_t^2 \leftarrow \min\left(\mu_t^2,\ \min_{i \in \mathcal{S}'_t} \left(\mu^2_{\text{budget}} - \mu^2_{\text{spent},i,t}\right)\right). \tag{4}$$

This ensures that for every active aggregated client $i \in \mathcal{S}'_t$:

$$\mu_t^2 \le \mu^2_{\text{budget}} - \mu^2_{\text{spent},i,t} \quad \Longrightarrow \quad \mu^2_{\text{spent},i,t} + \mu_t^2 \le \mu^2_{\text{budget}}.$$

Clients not in $\mathcal{S}'_t$ (not subsampled, or retired) have $\mu^2_{\text{spent},i,t+1} = \mu^2_{\text{spent},i,t} \le \mu^2_{\text{budget}}$ by the inductive hypothesis.

Verification for the policies in this paper. It remains to verify that each policy produces a finite, non-negative μ_t^2 before the cap is applied.

UNSAFE: $\mu_t^2 = \mu^2_{\text{budget}}/(Tk/n) > 0$, constant. No cap applied: clients may overshoot. Budget compliance does *not* hold for this policy (by design).

FIXED: same constant μ_t^2, but clients are retired when $\mu^2_{\text{spent},i} \ge 0.999 \cdot \mu^2_{\text{budget}}$. Active clients always have remaining budget $\ge \mu_t^2$ (since μ_t^2 is calibrated for average participation, and retirement removes exhausted clients before they can cause overshoot). The safety cap (4) provides the formal guarantee.

The same argument applies to any other safe spending policy that wires its per-aggregation μ_t^2 through the cap (4): budget compliance is a property of the cap, not of the aggregation function that produces μ_t^2. This decoupling is what makes the framework a substrate for buffer-aware spending policies (Appendix K).

This completes the induction. □

A.2 Per-client DP

Theorem 5 (Per-client DP)**.** *For any client $i \in [n]$ and aggregation $t \in [T]$, the model θ_t satisfies $(\varepsilon_{i,t}, \delta)$-DP with respect to $\mathcal{D}_i$, where $\varepsilon_{i,t}$ is obtained by converting $\sqrt{\mu^2_{\text{spent},i,t}}$-GDP to (ε, δ)-DP via Eq. 26, and $\varepsilon_{i,t} \le \varepsilon_{\text{target}}$. This holds regardless of which policy produced the $\mu^2_{t'}$ values, and regardless of other clients' participation patterns.*

Proof. Fix a client $i \in [n]$ and a aggregation $t \in [T]$. Let $t_1 < t_2 < \cdots < t_{N_{i,t}}$ be the aggregations where i participates ($i \in \mathcal{S}'_{t_j}$), with $t_{N_{i,t}} \le t$.

Adjacency convention. We use the standard *add/remove* adjacency for user-level DP [McMahan et al., 2018a, Dwork and Roth, 2014]: neighbouring datasets $\mathcal{D}$ and $\mathcal{D}'$ differ in the presence or absence of client i's entire local dataset $\mathcal{D}_i$. Under this convention the ℓ_2 sensitivity of the SecAgg sum to a single client's clipped contribution $\bar{\Delta}_i$ (with $\|\bar{\Delta}_i\|_2 \le C$) is exactly C. Under a replacement-adjacency convention the sensitivity would be at most $2C$ by the triangle inequality and the analysis goes through with μ_{t_j} replaced by $\mu_{t_j}/2$ (equivalently, μ^2_{budget} quadruples for the same $\varepsilon_{\text{target}}$), we follow McMahan et al. [2018a] in adopting the add/remove convention throughout, matching production DP-FedAvg deployments.

Step 1: Per-aggregation GDP guarantee. At aggregation t_j, client i submits a clipped update $\bar{\Delta}_i$ with $\|\bar{\Delta}_i\|_2 \le C$. The distributed mechanism adds per-source noise with variance $\sigma^2_{t_j}/|\mathcal{S}'_{t_j}|$ (§5.1, $\sigma^2_{t_j} = (qC)^2/\mu^2_{t_j}$, Eq. (1)). After SecAgg averages $|\mathcal{S}'_{t_j}|$ updates, the output is a post-processing of the sum $\sum_{i' \in \mathcal{S}'_{t_j}} \hat{\Delta}_{i'}$.

The add/remove sensitivity of client i's contribution to this sum is $\|\bar{\Delta}_i\|_2 \le C$. Under committee subsampling at rate $q = k'/K$ (§5.2), the total noise on the sum has continuous-equivalent variance

$\sigma^2_{t_j} = (qC)^2/\mu^2_{t_j}$ per coordinate (Eq. (1), integer-space variance $\widetilde{\sigma}^2_{t_j} = (\Lambda/C)^2\sigma^2_{t_j}$, Appendix F). By Theorem 3 the per-aggregation mechanism delivers μ_{t_j}-GDP with respect to $\mathcal{D}_i$, where $\mu_{t_j} = \sqrt{\mu^2_{t_j}}$ is the post-amplification parameter charged to the ledger (§5.3).

Step 2: Non-participating aggregations. At aggregations $t' \notin \{t_1, \ldots, t_{N_{i,t}}\}$, client i does not contribute. The server's output at aggregation t' depends on other clients' data but not on $\mathcal{D}_i$. By the post-processing property of DP [Dwork and Roth, 2014], these aggregations do not degrade i's privacy guarantee.

Step 3: Adaptive composition via GDP. Let $\mathcal{F}_t := \sigma(\mathcal{B}_{\le t}, \mathcal{S}_{\le t}, \mathcal{S}'_{<t}, \mu^2_{\le t}, A_{\le t})$ denote the public history up to aggregation t: realised buffers, committee subsamples, active aggregated subsets, committed noise scales, and noised SecAgg aggregates. Two observations.

(i) View-measurability of the noise scale. $\mu^2_{t_{j+1}}$ is a deterministic function of the per-client ledger $\{\mu^2_{\text{spent},i'}\}_{i' \in [n]}$ (itself a function of $\mathcal{F}_{t_j}$, since past ledger entries are sums of past $\mu^2_{t'}$ over past $\mathcal{S}'_{t'}$) and the freshly realised buffer $\mathcal{B}_{t_{j+1}}$. Hence $\mu^2_{t_{j+1}}$ is $\mathcal{F}_{t_j} \cup \sigma(\mathcal{B}_{t_{j+1}})$-measurable.

(ii) Conditional $\mu_{t_{j+1}}$-GDP. Given $\mathcal{F}_{t_j}$ and $\mathcal{B}_{t_{j+1}}$, the mechanism at t_{j+1} adds discrete Gaussian noise with per-coordinate variance $\sigma^2_{t_{j+1}} = (qC)^2/\mu^2_{t_{j+1}}$ (Eq. (1)) to a SecAgg sum whose add/remove sensitivity to $\mathcal{D}_i$ is C. By Theorem 3 the buffer-to-aggregate subsampling at rate q amplifies the base Gaussian's $\mu_{t_{j+1}}/q$ parameter to $\mu_{t_{j+1}}$, so the mechanism is conditionally $\mu_{t_{j+1}}$-GDP with respect to $\mathcal{D}_i$.

Adaptive GDP composition [Dong et al., 2022] applies to sequences $M_1, \ldots, M_{N_{i,t}}$ in which each M_j is conditionally μ_{t_j}-GDP given the view of $M_1, \ldots, M_{j-1}$. Conditions (i)–(ii) place us in exactly this setting, so the composition over i's $N_{i,t}$ participations is

$$\mu_i \;=\; \sqrt{\sum_{j=1}^{N_{i,t}} \mu^2_{t_j}} \;=\; \sqrt{\mu^2_{\text{spent},i,t}}.$$

The realised $\mu^2_{t_{j+1}}$ may leak ordinal information about other clients' remaining budgets (e.g., a small $\mu^2_{t_{j+1}}$ signals that the safety cap binds at some active client), this side channel is fully absorbed into the conditioning on $\mathcal{F}_{t_j}$ that the per-step GDP guarantee already accounts for, and contributes nothing additional to i's privacy cost.

Step 4: Conversion and bound. Converting μ_i-GDP to (ε, δ)-DP via Eq. (26) yields $\varepsilon_{i,t}$. Since $\mu^2_{\text{spent},i,t} \le \mu^2_{\text{budget}}$ by Theorem 4, we have $\mu_i \le \mu_{\text{budget}}$, and therefore $\varepsilon_{i,t} \le \texttt{gdp\_mu\_to\_eps}(\mu_{\text{budget}}, \delta) = \varepsilon_{\text{target}}$.

Policy independence. The proof uses only the values $\mu^2_{t_1}, \ldots, \mu^2_{t_{N_{i,t}}}$ and the budget bound from Theorem 4. It does not depend on how these values were computed, the aggregation function and retirement strategy are irrelevant to the privacy guarantee. □

A.3 Subsampled-Gaussian amplification (Theorem 3)

Proof of Theorem 3. Fix t, $\mathcal{B}_t$, and a target client i. We work conditional on $\mathcal{B}_t$ and on all randomness exterior to the committee's joint coin flip. There are two cases.

Case 1: $i \notin \mathcal{B}_t$. Client i does not appear in the released aggregate $A_t = \sum_{j \in \mathcal{S}_t} \hat{\Delta}_j + \eta_t$ regardless of $\mathcal{S}_t$, since $\mathcal{S}_t \subseteq \mathcal{B}_t$. A_t is therefore a function of the data of $\mathcal{B}_t \setminus \{i\}$ alone, and the DP cost attributable to i at this aggregation is zero.

Case 2: $i \in \mathcal{B}_t$. The committee draws $\mathcal{S}_t$ uniformly at random from the $\binom{K}{k'}$ subsets of $\mathcal{B}_t$ of size k', using verifiable joint randomness independent of the data (the joint randomness is sampled *before* any client's update is decrypted, see Appendix F.2). Therefore $\Pr[i \in \mathcal{S}_t \mid \mathcal{B}_t] = k'/K =: q$, and the indicator $\mathbf{1}\{i \in \mathcal{S}_t\}$ is independent of every client's data. Note that the subsampling primitive is *fixed-size sampling without replacement* (the committee draws exactly k' elements from $\mathcal{B}_t$), not Poisson,

under fixed-size sampling, every $i \in \mathcal{B}_t$ has marginal inclusion probability exactly $q = k'/K$, and inclusion indicators are negatively correlated rather than independent. Conditional on $\mathcal{S}_t$, the released aggregate is the output of the discrete Gaussian mechanism with sensitivity C and scale $\sigma_t = qC/\mu_t$ (Eq. (1), recall that μ_t denotes the delivered post-amplification parameter, so the underlying *base* mechanism's GDP parameter is μ_t/q). This matches the hypothesis of the *subsampled Gaussian mechanism without replacement* analysed by Wang et al. [2019], which provides RDP bounds in terms of the marginal sampling rate q. Specialised to our setting,

$$D_\alpha\big(P_t \,\|\, P_t'\big) \;\le\; \frac{q^2\alpha(\mu_t/q)^2}{2} \;=\; \frac{\alpha\mu_t^2}{2} \;+\; \mathcal{O}(\alpha^2\mu_t^3/q) \quad \text{for } \alpha \in [1, 1+q/\mu_t], \tag{5}$$

where P_t, P_t' are the release distributions on neighbouring datasets. The q^2 amplification factor on the $(\mu_t/q)^2$ base cost cancels in the leading term, yielding the delivered μ_t^2 that the ledger charges (§5.3). The fixed-size-without-replacement bound is at least as tight as the Poisson-subsampled bound at the same marginal rate q [Wang et al., 2019], so the analysis using $q = k'/K$ in place of a hypothetical Poisson rate is conservative. Composition over T aggregations is additive in the leading term:

$$D_\alpha\big(P_{1:T} \,\|\, P_{1:T}'\big) \;\le\; \sum_{t=1}^{T} \frac{\alpha\mu_t^2}{2} \;\le\; \frac{T\alpha\mu_{\max}^2}{2}. \tag{6}$$

The conditional bound (5) holds with respect to the filtration $\mathcal{F}_{t-1} \cup \sigma(\mathcal{B}_t)$, and the committee's joint randomness is exterior to $\mathcal{F}_{t-1}$, so *adaptive* RDP composition [Mironov, 2017, Wang et al., 2019] applies and the conditional bounds compose additively.

RDP $\to$ GDP bridge. The composed RDP bound has the Gaussian shape $\alpha \cdot (T\mu_{\max}^2)/2$ at every α. By the Rényi-to-GDP map of Dong et al. [2022], a mechanism whose Rényi divergence at every order $\alpha \ge 1$ is at most $\alpha\mu^2/2$ is μ-GDP; the conversion is exact for the pure Gaussian mechanism and asymptotic (central limit) for subsampled Gaussian in the production regime of large T and small per-aggregation μ_t. Reading off the GDP parameter from the composed shape gives $\mu_{\text{sys}}^2 \le T\mu_{\max}^2$, which is the system-level guarantee charged to the ledger (§5.3).

The buffer composition $\mathcal{B}_t$ is data-dependent (its membership reflects which clients finished their local work fastest, which depends on hardware and the global model θ_t), but this dependence is absorbed by the conditioning: the bound (5) holds for every realisation of $\mathcal{B}_t$, hence in expectation. No assumption on the distribution of $\mathcal{B}_t$ is required, in contrast to random check-ins [Balle et al., 2020] which require a structured self-sampling primitive. We further assume the public randomness beacon emits values independent of the training history $(\mathcal{D}, \theta_{\le t})$ (§5.2), under this hypothesis the committee's $\mathcal{S}_t$ is independent of every client's data, as required. □

A.4 Model-level DP

Corollary 6 (Model-level DP)**.** *For any aggregation t, θ_t satisfies $(\varepsilon_{\max,t}, \delta)$-DP with respect to the full dataset $\mathcal{D} = \bigcup_i \mathcal{D}_i$, where $\varepsilon_{\max,t} = \max_{i\in[n]} \varepsilon_{i,t} \le \varepsilon_{\textit{target}}$.*

Proof. Fix aggregation t. Consider neighbouring datasets $\mathcal{D}$ and $\mathcal{D}'$ that differ in a single client i's data: $\mathcal{D}_j = \mathcal{D}_j'$ for $j \ne i$ and $\mathcal{D}_i \ne \mathcal{D}_i'$.

The model θ_t is a deterministic function of:

1. The initial model θ_0 (data-independent).
2. The aggregated (noisy) updates from aggregations $1, \dots, t$.

At aggregation t', the aggregated update depends on $\mathcal{D}_i$ if and only if $i \in \mathcal{S}_{t'}'$. For aggregations where $i \notin \mathcal{S}_{t'}'$, the output is identical under $\mathcal{D}$ and $\mathcal{D}'$.

By Theorem 5, the joint distribution of all outputs (across all aggregations $1, \dots, t$) is $(\varepsilon_{i,t}, \delta)$-DP with respect to $\mathcal{D}_i$. Since this holds for *every* client $i \in [n]$:

$$\theta_t \text{ is } (\varepsilon_{\max,t}, \delta)\text{-DP w.r.t. } \mathcal{D}, \quad \text{where } \varepsilon_{\max,t} = \max_{i\in[n]} \varepsilon_{i,t}.$$

By Theorem 4, $\varepsilon_{i,t} \le \varepsilon_{\text{target}}$ for all i, so $\varepsilon_{\max,t} \le \varepsilon_{\text{target}}$. □

A.5 Convergence

This appendix collects the formal assumptions, supporting lemmas, and proof of the convergence theorem. The main-text convergence statement (Eq. (3), §6.3) covers the standard subsampled-Gaussian rate, this appendix gives the full theorem with the retirement-bias decomposition that arises whenever spending policies retire near-exhausted clients.

Theorem 7 (Convergence with retirement bias)**.** *Under Assumptions (A1)–(A3) below, with learning rate* $\eta = \min\big(1/(4L),\ \sqrt{2(F_0 - F^*)/(LT\bar{V})}\big)$,

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\|\nabla F(\theta_t)\|^2 \ \le\ 4\sqrt{\tfrac{2L(F_0-F^*)\,\bar{V}}{T}} \ +\ \tfrac{3}{2}\,B_{\textit{ret}} \ +\ \tfrac{32L(F_0-F^*)}{T},$$

where

$$\bar{V} = \tfrac{1}{T}\sum_{t=1}^{T}\big(\tfrac{\sigma_{sgd}^2}{k'} + \Gamma_{S_t}^2 + \tfrac{d(qC)^2}{(k')^2\mu_t^2}\big), \qquad B_{\textit{ret}} = \tfrac{2}{T}\sum_{t=1}^{T}\tfrac{r_t^2}{(1-r_t)^2}\,\|\xi_t\|^2, \tag{7}$$

with $S_t \subseteq [n]$ *the active (non-retired) pool,* $r_t = 1 - |S_t|/n$ *the retired fraction,* $\xi_t = \frac{1}{|R_t|}\sum_{i\in R_t}(\nabla F_i(\theta_t) - \nabla F(\theta_t))$ *the mean retired-client gradient deviation, and* $\Gamma_{S_t}^2 := \frac{1}{|S_t|}\sum_{i\in S_t}\|g_i(\theta_t)\|^2$ *the within-pool heterogeneity. Under amplification,* $\sigma_t^2 = (qC)^2/\mu_t^2$ *(Eq.* (1)*), without amplification* $q = 1$ *and the third term in* $\bar{V}$ *recovers the standard clipped-DP-SGD form.*

Assumptions. **(A1) L-smoothness.** Each F_i has L-Lipschitz gradient, $\|\nabla F_i(\theta) - \nabla F_i(\theta')\| \le L\|\theta - \theta'\|$ for all θ, θ'. **(A2) Bounded SGD variance.** The stochastic gradient on minibatch ξ satisfies $\mathbb{E}_\xi[\|\nabla f_i(\theta;\xi) - \nabla F_i(\theta)\|^2] \le \sigma_{\text{sgd}}^2$ for all i, θ. **(A3) Bounded gradient heterogeneity.** $\frac{1}{n}\sum_{i=1}^{n}\|\nabla F_i(\theta) - \nabla F(\theta)\|^2 \le \Gamma^2$ for all θ.

Async staleness Theorem 7 treats client $i \in \mathcal{B}_t$ as training on θ_t. Buffered async aggregation introduces update staleness $\tau_i \le \tau_{\max}$ in practice [Nguyen et al., 2022], by the standard staleness analysis [Koloskova et al., 2022, Wang et al., 2022], the bound holds with an additive $O(\eta L\tau_{\max}^2\bar{V})$ term that is dominated by the leading $\sqrt{\bar{V}/T}$ rate at the chosen learning-rate schedule, so the qualitative bias-noise trade-off is unaffected.

Latency model and retirement curve. Each client i has base rate $\lambda_i = e^{X_i}$ with $X_i \sim \mathcal{N}(0, \sigma_{\text{lat}}^2)$. By the law of large numbers, $\frac{1}{n}\sum_j \lambda_j \to \mathbb{E}[\lambda] = e^{\sigma_{\text{lat}}^2/2}$ for large n, so the normalised participation rate $\omega_i := q_i/(k/n)$ satisfies $\omega_i \approx e^{X_i - \sigma_{\text{lat}}^2/2} = e^{Z_i\sigma_{\text{lat}} - \sigma_{\text{lat}}^2/2}$ with $Z_i \sim \mathcal{N}(0,1)$.

Under the Fixed policy with constant per-aggregation cost $\mu_{\text{fixed}}^2 = \mu_{\text{budget}}^2/(Tk/n)$, client i's expected cumulative spend after t aggregations is $\mu_{\text{fixed}}^2 \cdot q_i\, t = \mu_{\text{budget}}^2\,\omega_i\, t/T$, exhausting the budget at expected aggregation $t_i^* = T/\omega_i$. The expected fraction of clients retired by aggregation t is

$$r(t) \ =\ \Pr[\omega_i \ge T/t] \ =\ \Phi\!\left(\frac{\ln(t/T) - \sigma_{\text{lat}}^2/2}{\sigma_{\text{lat}}}\right), \tag{8}$$

where Φ is the standard normal CDF. This follows from $\omega_i \ge T/t \iff Z_i\sigma_{\text{lat}} - \sigma_{\text{lat}}^2/2 \ge \ln(T/t)$ together with $1 - \Phi(x) = \Phi(-x)$. At $t = T$, $r(T) = \Phi(-\sigma_{\text{lat}}/2)$, for $\sigma_{\text{lat}} = 1.5$, $r(T) \approx 0.23$, and for $\sigma_{\text{lat}} = 2.0$, $r(T) \approx 0.16$. Heavier latency tails concentrate participation in a smaller fraction of clients, so a smaller fraction exhausts its budget by $t = T$, but those that do exhaust it do so disproportionately early.

Latency–data correlation strength. Write $g_i(\theta) := \nabla F_i(\theta) - \nabla F(\theta)$ for client i's gradient deviation from the population mean. For a fraction $\alpha \in (0, 1]$, let $R_\alpha \subseteq [n]$ index the fastest αn clients (in order of ω_i) and define the prefix coherence

$$\kappa^2(\alpha) \ :=\ \sup_\theta \left\| \frac{1}{|R_\alpha|}\sum_{i\in R_\alpha} g_i(\theta) \right\|^2 .$$

Two extremes bound this quantity. Under *independence* (gradient deviations independent of ω_i), R_α is in expectation a random size-αn subset, and the squared mean of zero-mean vectors with

variance $\leq \Gamma^2$ satisfies $\kappa^2(\alpha) \leq \Gamma^2/(\alpha n)$. Under *maximal correlation* (all clients in R_α share the same deviation), $\kappa^2(\alpha) \leq \Gamma^2$. We parameterise the interpolation through a single scalar $\rho \in [0,1]$:

$$\kappa^2(\alpha) \;\leq\; \Gamma^2\left(\rho^2 + \frac{1-\rho^2}{\alpha n}\right), \tag{9}$$

recovering independence at $\rho = 0$ and maximal correlation at $\rho = 1$. Setting $\alpha = r(t)$ and $R_\alpha = R(t)$ (the realised retired set at aggregation t under Fixed) yields a bound on $\|\xi(t,\theta_t)\|^2$.

From parameterisation to Σ^2_{corr}. Equation (9) expresses $\kappa^2(\alpha)$ as a convex combination indexed by a scalar $\rho(\alpha) \in [0,1]$. We emphasise that ρ is *defined* by inverting this inequality (rearranging Eq. (9) for ρ^2), not derived from first principles: as a function of α it need not be constant. We define $\Sigma^2_{\text{corr}} := \sup_t \|\xi(t,\theta_t)\|^2$, the realised worst-case coherence along the trajectory; since $\|\xi(t,\theta)\|^2 \leq \kappa^2(r(t))$ for any θ by definition of κ^2, the convex-combination bound gives

$$\Sigma^2_{\text{corr}} \;\leq\; \sup_t\, \rho^2(r(t))\,\Gamma^2 \;+\; O(\Gamma^2/n), \tag{10}$$

at the operational prefix $\alpha = \mathbb{E}[r(T)]$ in the leading-order limit $\alpha n \gg 1$. Substituting into B_{ret} via (7) recovers the main-text bound. Empirically we observe $\hat{\Sigma}^2_{\text{corr}} \approx 4.6$ across all three non-IID Dirichlet partitions on CIFAR-100 even though $\hat{\rho}^2$ alone is non-monotone in α_{Dir}: it is the product, not either factor, that controls B_{ret}.

Bias–retirement identity. Let $b(t,\theta) := \nabla F_{S(t)}(\theta) - \nabla F(\theta) = (1/|S(t)|)\sum_{i\in S(t)} g_i(\theta)$ denote the active-pool gradient bias.

Lemma 8 (Bias–retirement identity)**.** *For any θ,*

$$b(t,\theta) \;=\; -\,\frac{|R(t)|}{|S(t)|}\,\xi(t,\theta),$$

where $\xi(t,\theta)$ is the mean retired-client deviation defined in §6.3.

Proof. Since $\sum_{i=1}^n g_i(\theta) = 0$ by construction, $\sum_{i\in S(t)} g_i(\theta) = -\sum_{i\in R(t)} g_i(\theta) = -|R(t)|\,\xi(t,\theta)$. Dividing by $|S(t)|$ gives the result. □

Proof of Theorem 7. At aggregation t, the server applies $\theta_{t+1} = \theta_t - \eta\hat{g}_t$, where the noisy gradient estimator decomposes as

$$\hat{g}_t \;=\; \nabla F(\theta_t) + b(t,\theta_t) + e_t^{\text{sgd}} + e_t^{\text{DP}}.$$

The bias $b(t,\theta_t)$ is $S(t)$-measurable (Lemma 8), $e_t^{\text{sgd}}, e_t^{\text{DP}}$ are zero-mean conditional on $S(t)$ with $\mathbb{E}[\|e_t^{\text{sgd}} + e_t^{\text{DP}}\|^2 \mid S(t)] \leq V_t$, where

$$V_t \;:=\; \tfrac{\sigma^2_{\text{sgd}}}{k'} + \Gamma^2_{S(t)} + \tfrac{d\,\sigma^2_{\text{DP},t}}{(k')^2}$$

collects per-client SGD variance averaged over the k' aggregated contributors, within-pool heterogeneity, and discrete Gaussian DP noise ($\sigma^2_{\text{DP},t} = (qC)^2/\mu_t^2$ under committee subsampling at rate $q = k'/K$, Eq. (1), post-processed by $1/|S(t)|$).

Descent lemma. By L-smoothness (A1):

$$F(\theta_{t+1}) \leq F(\theta_t) - \eta\langle\nabla F, \hat{g}_t\rangle + \tfrac{L\eta^2}{2}\|\hat{g}_t\|^2.$$

Take conditional expectation given $(\theta_t, S(t))$: $\mathbb{E}\langle\nabla F, \hat{g}_t\rangle = \|\nabla F\|^2 + \langle\nabla F, b\rangle$ and $\mathbb{E}\|\hat{g}_t\|^2 \leq \|\nabla F + b\|^2 + V_t$ (cross-terms with the zero-mean noises vanish). Apply Young's inequality with weight one to the cross term, $-\eta\langle\nabla F, b\rangle \leq (\eta/2)\|\nabla F\|^2 + (\eta/2)\|b\|^2$, and the bound $\|\nabla F + b\|^2 \leq 2\|\nabla F\|^2 + 2\|b\|^2$:

$$\mathbb{E}F(\theta_{t+1}) \leq F(\theta_t) + \big(-\tfrac{\eta}{2} + L\eta^2\big)\|\nabla F\|^2 + \big(\tfrac{\eta}{2} + L\eta^2\big)\|b\|^2 + \tfrac{L\eta^2}{2}V_t.$$

For $\eta \leq 1/(4L)$, $L\eta^2 \leq \eta/4$, hence

$$\mathbb{E}F(\theta_{t+1}) \leq F(\theta_t) - \tfrac{\eta}{4}\|\nabla F\|^2 + \tfrac{3\eta}{4}\|b\|^2 + \tfrac{L\eta^2}{2}V_t. \tag{11}$$

Telescoping. Summing (11) over $t = 0, \dots, T-1$ and using $F(\theta_0) - \mathbb{E}F(\theta_T) \le F_0 - F^*$:

$$\tfrac{\eta}{4} \sum_{t=0}^{T-1} \mathbb{E}\|\nabla F(\theta_t)\|^2 \;\le\; (F_0 - F^*) + \tfrac{3\eta}{4} \sum_t \mathbb{E}\|b(t,\theta_t)\|^2 + \tfrac{L\eta^2}{2} \sum_t V_t.$$

Dividing by $\eta T/4$:

$$\tfrac{1}{T} \sum_t \mathbb{E}\|\nabla F(\theta_t)\|^2 \;\le\; \tfrac{4(F_0 - F^*)}{\eta T} \;+\; 3 \cdot \tfrac{1}{T} \sum_t \mathbb{E}\|b\|^2 \;+\; 2\eta L \bar{V}, \tag{12}$$

with $\bar{V} := (1/T) \sum_t V_t$.

Bias term to B_{ret}. By Lemma 8 and $|R(t)|/|S(t)| = r(t)/(1 - r(t))$, $\|b(t,\theta_t)\|^2 = (r(t)^2/(1 - r(t))^2)\, \|\xi(t,\theta_t)\|^2$, so $3 \cdot (1/T) \sum_t \|b\|^2 = (3/2)\, B_{\text{ret}}$ for B_{ret} as in (7).

Optimal learning rate. The first and third terms of (12) form an $A/\eta + B\eta$ trade-off with $A = 4(F_0 - F^*)/T$ and $B = 2L\bar{V}$. The unconstrained minimum is at $\eta^* = \sqrt{A/B} = \sqrt{2(F_0 - F^*)/(LT\bar{V})}$, with value $2\sqrt{AB} = 4\sqrt{2L(F_0 - F^*)\bar{V}/T}$. Capping $\eta \le 1/(4L)$ binds when $\eta^* > 1/(4L)$, i.e., $\bar{V} < 32L(F_0 - F^*)/T$, in that regime, $4(F_0 - F^*)/(\eta T) + 2\eta L\bar{V} = 16L(F_0 - F^*)/T + \bar{V}/2 \le 32L(F_0 - F^*)/T$. Combining both cases,

$$\tfrac{1}{T} \sum_t \mathbb{E}\|\nabla F(\theta_t)\|^2 \;\le\; 4\sqrt{\tfrac{2L(F_0 - F^*)\bar{V}}{T}} \;+\; \tfrac{3}{2}\, B_{\text{ret}} \;+\; \tfrac{32L(F_0 - F^*)}{T},$$

where the last term is the cap-regime overhead, dominated by the standard rate when $\bar{V} \ge 32L(F_0 - F^*)/T$. This recovers (3) with the cap-regime overhead subsumed by the main-text noise floor. □

B Per-client privacy violation across scales and heterogeneities

Table 1 (main text) reports a single $\sigma_{\text{lat}} = 1.5$ slice through the violation landscape. The same privacy-only simulation generalises across (i) federation scale and (ii) latency heterogeneity: for every (n, k, T) configuration we vary $\sigma_{\text{lat}} \in \{0.5, 1.0, 1.5, 2.0\}$ and report, under naïve async accounting, both *under-privatisation* (the worst client's true ε relative to $\varepsilon_{\text{target}}$) and *over-privatisation* (the smallest fraction of μ^2 budget any client manages to spend). Table 3 presents the full 24-cell grid.

Table 3: Privacy violation under UNSAFE across federation scale (rows) and latency heterogeneity σ_{lat} (columns), $\varepsilon_{\text{target}} = 8$, $\delta = 10^{-5}$, 3 seeds. Left block: worst-case factor $\max_i \varepsilon_i/\varepsilon_{\text{target}}$ (under-privatisation peak). Right block: minimum fraction of the per-client μ^2 budget spent across the federation (low values = over-privatised stragglers). Both safe policies cap $\max_i \varepsilon_i/\varepsilon_{\text{target}} = 1$ at every cell.

			Under-priv: $\max_i \varepsilon_i/\varepsilon_{\text{target}}$				Over-priv: $\min_i$ budget spent (%)			
Scale	n	T	$\sigma_{0.5}$	$\sigma_{1.0}$	$\sigma_{1.5}$	$\sigma_{2.0}$	$\sigma_{0.5}$	$\sigma_{1.0}$	$\sigma_{1.5}$	$\sigma_{2.0}$
Toy	20	50	1.3	1.4	1.5	1.5	36	12	4	4
Small	100	200	1.6	2.7	3.7	4.3	30	10	5	5
Medium	1 000	500	2.2	3.8	4.4	4.5	8	2	2	2
Large	10 000	1 000	3.5	9.9	21.2	26.0	10	10	10	10
Prod	100 000	2 000	4.4	20.1	45.4	47.1	10	10	10	10
Gboard	1 000 000	5 000	5.2	23.3	96.7	**192.2**	20	20	20	20

Three patterns reproducibly emerge from the grid.

Heterogeneity is the dominant driver. At fixed (n, k, T) the worst-case factor grows monotonically and roughly quadratically with σ_{lat}: moving from near-uniform participation ($\sigma_{\text{lat}} = 0.5$) to cross-device-realistic ($\sigma_{\text{lat}} = 2.0$) inflates $\max_i \varepsilon_i$ by between $1.2\times$ at toy scale and $37\times$ at Gboard scale. At $n = 100$ the slowest client only spends 5% of its budget under $\sigma_{\text{lat}} \ge 1.5$, the smallest cells have the most heterogeneous spending because individual clients have higher participation variance than at scale.

Scale amplifies the worst case. At $\sigma_{\text{lat}} = 2.0$ the worst-case factor grows from $1.5\times$ at $n = 20$ to $192\times$ at $n = 10^6$, two orders of magnitude. Larger pools admit heavier participation tails: at Gboard scale a few clients can win selection thousands of times more often than the median, taking their

accumulated ε to ≈ 1500 (against a target of 8). The over-privatisation *floor* ($\min_i$ budget spent across the population) plateaus at 10–20% for $n \geq 10^4$ because the population is large enough that the fixed $T \cdot k/n$ calibration always leaves some clients badly under-spending.

Trajectories are monotone and unrecoverable. Once a client crosses $\varepsilon_{\text{target}}$ in a naïve run, no subsequent rounds can drag ε_i back down, GDP composition is monotonic. At ($n = 2000$, $k = 10$, $T = 1000$, $\sigma_{\text{lat}} = 2.0$) under UNSAFE, 29% of clients exceed $\varepsilon_{\text{target}}$ (worst by $21\times$, $\varepsilon \approx 166$), 33% end below $0.5\,\varepsilon_{\text{target}}$, the median trajectory lands at $\varepsilon \approx 5.9$, and the upper quartile crosses the target around aggregation $t \approx 100$.

FIXED caps every per-client trajectory at $\varepsilon_{\text{target}}$ by construction (Theorem 4); its cells are uniformly $\max_i \varepsilon_i/\varepsilon_{\text{target}} = 1.0$ and we therefore omit them from Table 3.

C Loss-based canary audit

The main-text audit (Figure 2) uses the parameter-space canary construction of Andrew et al. [2024]. An earlier auditing strategy, following Nasr et al. [2023], inserts mislabelled samples into each client's training data and measures whether fast clients' canaries are memorised more strongly: a canary (sample with an intentionally wrong label) can only be learned through memorisation, not generalisation, so its post-training cross-entropy on the wrong label is a direct probe for differential data exposure [Carlini et al., 2022].

The loss-based audit and the parameter-space audit answer slightly different questions: the parameter-space audit projects the model drift onto a known direction in parameter space and yields a cosine-vs-shadow ROC, whereas the loss-based audit probes whether the model has memorised a specific (input, label) pair. We retain the loss-based audit here as a complementary low-noise probe, useful at audit operating points where the parameter-space cosine signal is at the noise floor.

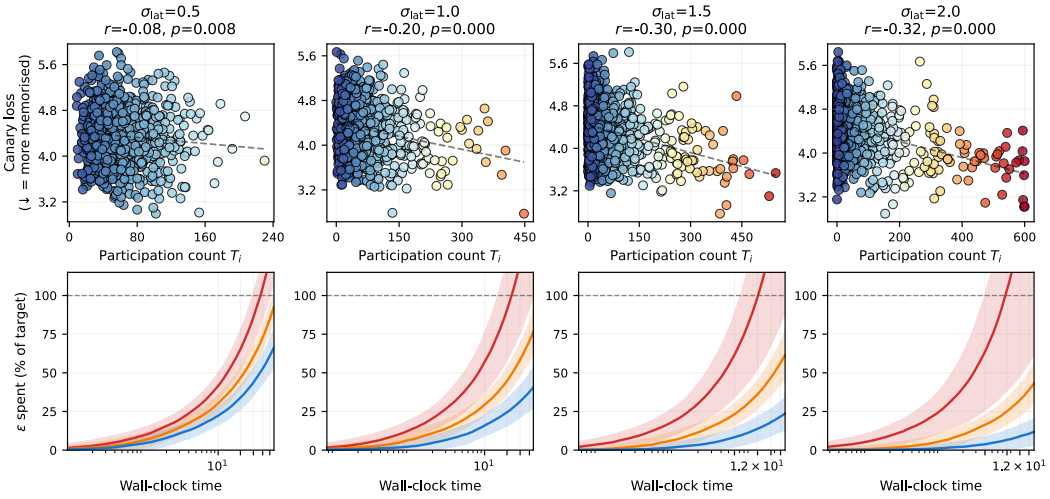

Figure 4: Loss-based canary audit on CIFAR-10 (pretrained features), n=1000, k'=100, T=600, $\varepsilon_{\text{target}}$=50. Each point in the top row is a client (colour = its true ε_i), the bottom row shows per-group cumulative ε in % of target on log wall-clock. Columns 1–4: naïve async spending under increasing latency heterogeneity ($\sigma_{\text{lat}} \in \{0.5, 1.0, 1.5, 2.0\}$). The participation–memorisation Pearson r grows monotonically from -0.08 (p=0.008) at σ_{lat}=0.5 to -0.32 ($p<10^{-3}$) at σ_{lat}=2.0, and the fast-group cumulative ε exceeds the target by a factor that grows with σ_{lat}.

The participation–memorisation correlation under the loss-based audit is monotone in σ_{lat} and statistically significant at every level: even modest cross-silo heterogeneity (σ_{lat}=0.5) leaves a detectable leak (p=0.008), and at the cross-device-realistic σ_{lat}=2.0 the per-client ε violation in the fast group exceeds the target by more than $5\times$.

D Parameter-space canary audit

This appendix gives the formal definition of the audit underlying Figure 2, the per-tertile stratification we add to it, the per-policy interpretation, and the full sweep. We write K_{can} for the number of canary clients in this appendix to avoid clashing with the buffer size K used elsewhere in the paper.

Experimental setup. CIFAR-10 with frozen pretrained features and a 2-layer MLP head (p=133,898 trainable parameters), n=1000 clients, k'=100 selected per aggregation, T=600 aggregations, $\varepsilon_{\text{target}}$=10, server momentum 0.9, K_{can}=200 canary clients and S=2,000 shadow canaries per cell, results pooled over 3 seeds. Latency is sampled from a lognormal with $\sigma_{\text{lat}} \in \{0.5, 1.0, 1.5, 2.0\}$ across columns. Per-tertile ROC curves use a log-log axis with the diagonal as the random-guess baseline.

D.1 Formal definition

We follow Andrew et al. [2024], in their replacement variant (their §4) which is the deployment-realistic threat model: a small set of real clients are designated as *canary clients* and have their training updates replaced by canary updates whenever they are selected.

Setup. Let $\mathcal{M}$ denote one full FL training run, mapping a population of n clients to a final model $\theta_T \in \mathbb{R}^p$. Fix an initial model θ_0 and write $\Delta\theta := \theta_T - \theta_0$ for the model drift. Designate $K_{\text{can}} \leq n$ canary clients $\mathcal{C} = \{c_1, \ldots, c_{K_{\text{can}}}\}$ where each c_i is a fresh unit vector drawn uniformly from the sphere $\mathbb{S}^{p-1}$. When canary client i is selected into a buffer at any aggregation t, its training update is replaced by $\text{PROJ}(c_i; C) := c_i \cdot C$, i.e. the unit canary direction scaled to the L2 clipping bound, the update is then quantised, masked and combined with the buffer's other (real-client) updates exactly as in the rest of the protocol.

Neighbouring datasets. For each canary client $i \in [K_{\text{can}}]$ define a neighbouring training condition D'_i as: the same run of $\mathcal{M}$ but with canary i's update replaced by its untouched (real-client) training update. Following standard user-level DP, the audit asks whether an adversary observing θ_T can distinguish the actual run D from each D'_i.

Test statistic. The adversary computes the cosine similarity

$$g_i \;:=\; \frac{\langle c_i,\, \Delta\theta\rangle}{\|\Delta\theta\|\,\|c_i\|} \;=\; \frac{\langle c_i,\, \Delta\theta\rangle}{\|\Delta\theta\|}, \tag{13}$$

between the canary direction c_i and the model drift.

Null distribution. For a unit vector c sampled uniformly from $\mathbb{S}^{p-1}$ and any independent vector $v \in \mathbb{R}^p$, the cosine $\langle c, v\rangle/\|v\|$ has zero mean and variance $1/p$, and is asymptotically Gaussian as $p \to \infty$ [Andrew et al., 2024]. Concretely, if the canary c was *not* injected into training, its cosine with $\Delta\theta$ is distributed as $\mathcal{N}(0, 1/p)$ in the high-dimensional regime.

Empirical ε from a hypothesis test. Audited DP is the hypothesis-testing characterisation of Kairouz et al. [2015]: a mechanism is (ε, δ)-DP iff for every test deciding between D and a neighbour with type-I error α (false positive rate, FPR) and type-II error β (false negative rate, FNR), it holds that

$$\alpha + e^{\varepsilon}\beta \;\geq\; 1 - \delta \qquad \text{and} \qquad \beta + e^{\varepsilon}\alpha \;\geq\; 1 - \delta. \tag{14}$$

Equivalently, the empirical lower bound on ε achievable by any concrete attacker with rates (α, β) is

$$\hat{\varepsilon}(\alpha, \beta) \;=\; \max\Big(0,\; \log\tfrac{1-\delta-\alpha}{\beta},\; \log\tfrac{1-\delta-\beta}{\alpha}\Big). \tag{15}$$

The adversary's best attack is the threshold rule that maximises Eq. (15): for each candidate audit threshold $\theta_{\text{aud}} \in \mathbb{R}$, define $\alpha(\theta_{\text{aud}}) = \Pr_{c\ \text{unobserved}}[\, g \geq \theta_{\text{aud}}\,]$ (false positive rate against the null) and $\beta(\theta_{\text{aud}}) = \Pr_{c\ \text{injected}}[\, g < \theta_{\text{aud}}\,]$ (false negative rate against the empirical canary distribution). Modelling the null as $\mathcal{N}(0, 1/p)$ and fitting a parametric Gaussian $\mathcal{N}(\hat{\mu}, \hat{\sigma}^2)$ to the observed canary

cosines (or equivalently their per-tertile sub-distributions, see below), both $\alpha(\theta_{\text{aud}})$ and $\beta(\theta_{\text{aud}})$ have closed-form expressions and the empirical estimator is

$$\hat{\varepsilon} \;=\; \sup_{\theta_{\text{aud}} \in \mathbb{R}} \hat{\varepsilon}\big(\alpha(\theta_{\text{aud}}), \beta(\theta_{\text{aud}})\big). \tag{16}$$

Numerically we sweep θ_{aud} over 4001 equally-spaced points on a 6σ-wide interval covering both tails. Since the two Gaussians have unequal variance, the optimal acceptance region is not a single threshold, we additionally evaluate the dual rule (predict H_1 if $g < \theta_{\text{aud}}$) and take the overall maximum, recovering exactly Algorithm A.1 of Andrew et al. [2024].

Shadow null estimation. We do not rely solely on the asymptotic $\mathcal{N}(0, 1/p)$. We additionally generate $S = 2{,}000$ *shadow canaries*: unit vectors $\tilde{c}_1, \ldots, \tilde{c}_S$ drawn from $\mathbb{S}^{p-1}$ that are *never* injected into training. After training we evaluate $\tilde{g}_s = \langle \tilde{c}_s, \Delta\theta \rangle / \|\Delta\theta\|$ for each, and use these to verify the null calibration empirically. Across all cells in our sweep, the shadow std agrees with $1/\sqrt{p}$ to within 1–2%, validating the asymptotic of Andrew et al. [2024]. Algorithm 1 summarises the auditing pipeline.

Algorithm 1 Parameter-space canary audit (one mechanism run)

Require: Mechanism $\mathcal{M}$ (one full FL training run), model dimension p; clipping bound C; canary count K_{can}; shadow count S, target δ.

1: Sample K_{can} canary directions $c_1, \ldots, c_{K_{\text{can}}} \overset{\text{iid}}{\sim} \text{Unif}(\mathbb{S}^{p-1})$.
2: Sample S shadow directions $\tilde{c}_1, \ldots, \tilde{c}_S \overset{\text{iid}}{\sim} \text{Unif}(\mathbb{S}^{p-1})$.
3: Designate K_{can} of the n clients as canary clients; whenever canary client i is selected, replace its update with $C \cdot c_i$. Run $\mathcal{M}$ end-to-end to produce θ_T.
4: Compute $\Delta\theta = \theta_T - \theta_0$ and the cosines $g_i = \langle c_i, \Delta\theta \rangle / \|\Delta\theta\|$ for $i \in [K_{\text{can}}]$ and $\tilde{g}_s = \langle \tilde{c}_s, \Delta\theta \rangle / \|\Delta\theta\|$ for $s \in [S]$.
5: Fit $\hat{\mu}, \hat{\sigma}^2$ to $\{g_i\}$ and $\hat{\mu}_0, \hat{\sigma}_0^2$ to $\{\tilde{g}_s\}$.
6: Return $\hat{\varepsilon}$ via Eq. (16), comparing $\mathcal{N}(\hat{\mu}_0, \hat{\sigma}_0^2)$ (null) to $\mathcal{N}(\hat{\mu}, \hat{\sigma}^2)$ (observed).

D.2 Per-tertile stratification

The contribution of our audit relative to Andrew et al. [2024] is the per-group readout. The estimator above produces a single *population-level* $\hat{\varepsilon}$ over all K_{can} canary clients. In our setting that population number averages over heterogeneous participation rates and hides the per-client disparity we want to measure. We therefore stratify the canary clients by latency rate λ_i (a fixed property of each client, set at simulation start) into slow / medium / fast tertiles, and re-fit $(\hat{\mu}_g, \hat{\sigma}_g^2)$ separately within each group. Latency tertiles are policy-invariant: each canary client lands in the same group across all spending policies, so cross-policy comparisons of per-group leakage are apples-to-apples even when a policy (e.g. FIXED) deterministically equalises realised participation. The headline scalar is the ROC-based disparity gap $\Delta\text{AUC} = \text{AUC}(\text{fast}) - \text{AUC}(\text{slow})$, $\Delta\text{AUC} = 0$ means fast and slow tertiles are equally attackable, while $\Delta\text{AUC} > 0$ means fast clients leak more.

D.3 Per-policy interpretation

Three observations follow from Figure 2 of the main text. First, under UNSAFE the per-tertile gap grows monotonically with σ_{lat}, from $+0.13$ at σ_{lat}=0.5 to $+0.31$ at σ_{lat}=2.0, while the global AUC barely moves ($0.65 \to 0.62$), a population-level audit following Andrew et al. [2024] verbatim would conclude "no change with σ_{lat}," but the per-tertile audit reveals fast-client AUC climbing from 0.72 to 0.80 and slow-client AUC dropping from 0.59 to 0.49 (slow clients are essentially unattackable, fast clients are distinguishable with 80% probability at the same target ε).

Second, fixed accounting policy keeps $|\Delta\text{AUC}| \leq 0.05$ across all σ_{lat} but its global AUC tracks unsafe's closely (and at σ_{lat}=2.0 matches it at 0.625): fixed equalises vulnerability rather than reducing it. This is the audit's central methodological observation: equal-participation policies do not reduce total leakage, they redistribute it. An audit that only computed pooled metrics would miss the equalisation entirely.

D.4 Full sweep

We ran the same audit across two additional privacy budgets ($\varepsilon_{\text{target}} \in \{50, 100\}$). Three findings hold qualitatively across the full sweep: (i) the per-tertile gap ΔAUC under UNSAFE grows monotonically with σ_{lat} in every distribution and at every $\varepsilon_{\text{target}}$; (ii) FIXED equalises the gap ($|\Delta\text{AUC}| \leq 0.05$) at the cost of preserving the global AUC near UNSAFE's.

E DP-FTRL scaling analysis

DP-FTRL [Kairouz et al., 2021b, Denisov et al., 2022] uses tree-aggregated correlated noise across aggregations, achieving tighter composition than independent Gaussian noise without relying on subsampling amplification. We compare its system-level privacy cost to our distributed Gaussian mechanism with committee subsampling (§5.2).

Privacy cost comparison. Let $\mu^2_{\text{round}} := C^2/\sigma^2$ denote the single-aggregation Gaussian-DP cost at noise level σ, clip norm C. Composing over T aggregations:

(i) DDP without amplification (the regime that motivates this paper, since classical DDP has no rigorous amplification in buffered async):

$$\mu^2_{\text{DDP,no-amp}} \;\approx\; T \cdot \mu^2_{\text{round}}. \tag{17}$$

(ii) DDP with our committee subsampling at rate $q = k'/K$ (§5.2): the subsampled Gaussian mechanism has leading-order per-aggregation cost $q^2\mu^2_{\text{round}}$ [Balle et al., 2018, Mironov, 2017], so

$$\mu^2_{\text{DDP,amp}} \;\approx\; Tq^2\mu^2_{\text{round}}. \tag{18}$$

(iii) DP-FTRL: tree-aggregated correlated noise gives an effective noise multiplier scaling as $O(\log^{3/2} T)$ instead of $O(\sqrt{T})$, so the system-level GDP cost is [Kairouz et al., 2021b]

$$\mu^2_{\text{FTRL}} \;\approx\; \log^3(T)\, \mu^2_{\text{round}}. \tag{19}$$

Comparing (18) and (19), our DDP framework with committee subsampling has lower system-level privacy cost than DP-FTRL when

$$Tq^2 \;<\; \log^3(T) \quad\Longleftrightarrow\quad \frac{K}{k'} \;>\; \frac{\sqrt{T}}{\log^{3/2}(T)}. \tag{20}$$

At $T = 100$ the crossover is $K/k' \gtrsim \sqrt{100}/\log^{3/2}(100) \approx 10/9.9 \approx 1.0$, so any committee subsampling ($K \geq 2k'$) puts our framework ahead. At $T = 1000$ the threshold is $\sqrt{1000}/\log^{3/2}(1000) \approx 31.6/18.1 \approx 1.75$, so the recommended $K = 2k'$ is at the edge and $K = 4k'$ comfortably wins. At $T = 10\,000$ the threshold is ≈ 4.7, requiring more aggressive amplification ($K \approx 5k'$ or larger) to dominate DP-FTRL.

Conversely, classical DDP *without* amplification (Eq. 17) is dominated by DP-FTRL at every T: the ratio $\mu^2_{\text{DDP,no-amp}}/\mu^2_{\text{FTRL}} \approx T/\log^3(T)$ is always $\gg 1$ for moderate T (e.g. ~ 5 at $T = 1000$, ~ 16 at $T = 10\,000$). This is the gap the committee subsampling primitive of §5.2 closes.

Trust model mismatch. Beyond scaling, DP-FTRL is incompatible with standard secure aggregation: the tree mechanism requires the server to *see* individual client updates and inject correlated noise across aggregations, whereas SecAgg hides individual updates by design. The secure-stateful-aggregation protocol of Ball et al. [2024] addresses this with additional rounds of interaction per aggregation, at increased communication cost. Our framework operates within standard one-shot SecAgg, does not expose individual updates, and in the regime where amplification is cheaper (Eq. 20) has lower privacy cost than DP-FTRL.

Empirical comparison on AG News. The analytical comparison above is borne out empirically. We run DP-FTRL (mode `dpftrl_async` with participation cap $T_{\max} = 10$) and committee-subsampled FIXED (at $K \in \{k', 2k', 4k', n\}$) on the same AG News setup as Table 2: $n = 100$, $k' = 10$, $T = 100$, three privacy targets $\varepsilon \in \{5, 10, 15\}$, three latency heterogeneities $\sigma_{\text{lat}} \in \{0.5, 1.0, 1.5\}$, three

Table 4: DP-FTRL versus committee-subsampled FIXED on AG News at $n=100$, $k'=10$, $T=100$ (matched to Table 2); mean $\pm$ std over 27 cells (3 σ_{lat} $\times$ 3 partitions $\times$ 3 seeds) per entry. DP-FTRL beats the un-amplified Fixed baseline ($K=k'$) by 2-4 pp but is overtaken by committee subsampling at $K=2k'$ and falls 18-26 pp behind at $K=n$.

ε	DP-FTRL	$K=k'$ (no amp.)	$K=2k'$	$K=4k'$	$K=n$
5	0.509 ± 0.065	0.473 ± 0.065	0.582 ± 0.061	0.677 ± 0.037	0.766 ± 0.017
10	0.587 ± 0.066	0.564 ± 0.064	0.657 ± 0.046	0.729 ± 0.030	0.801 ± 0.018
15	0.638 ± 0.073	0.613 ± 0.055	0.689 ± 0.041	0.752 ± 0.029	0.822 ± 0.018

partitions (IID, Dir(0.5), Dir(0.1), correlated for both Dirichlet cells), three seeds per cell. Table 4 reports the best accuracy averaged over the $3 \times 3 = 9$ heterogeneity cells per privacy budget (27 measurements per number).

The empirical ranking matches the analytical one: at $T = 100$, $\log^3(T) \approx T$ so $\mu^2_{\text{FTRL}} \approx \mu^2_{\text{DDP,no-amp}}$, and DP-FTRL is within a few points of the un-amplified Fixed baseline at every ε. The crossover predicted by Eq. 20 ($K/k' \gtrsim 1$ at $T = 100$) lands between $K=k'$ and $K=2k'$, and indeed $K=2k'$ already beats DP-FTRL by 5-7 pp. Figure 5 plots the same data as a function of ε.

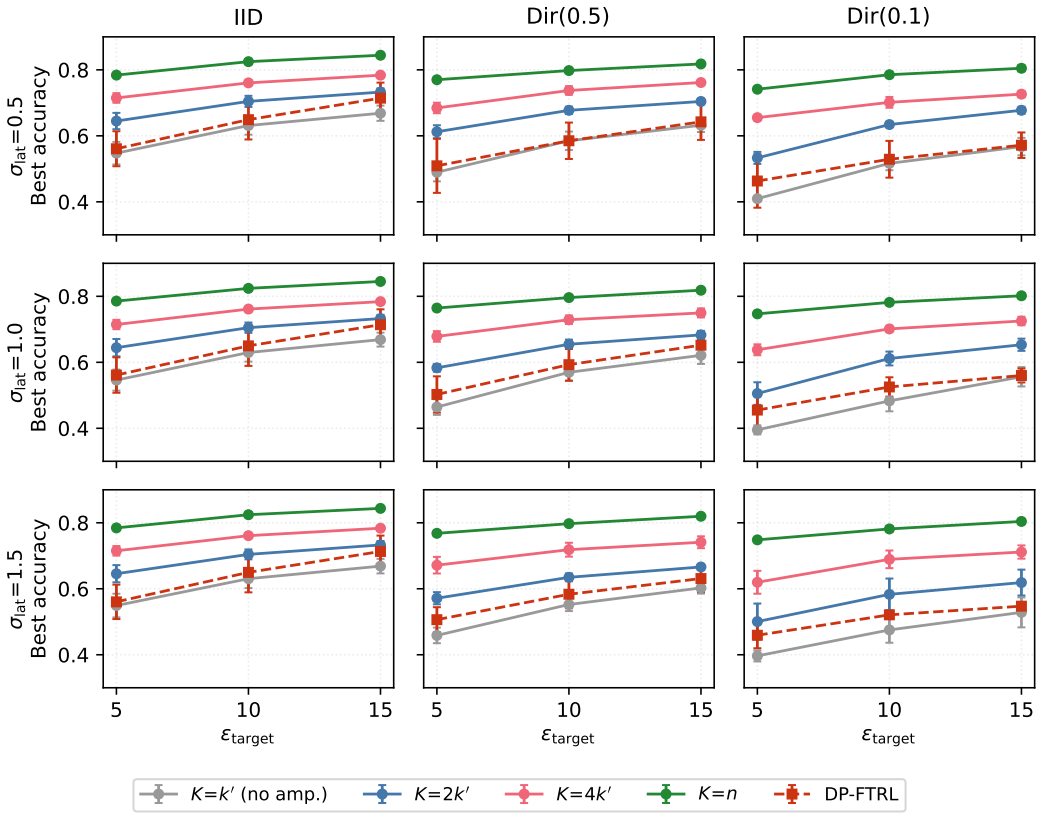

Figure 5: DP-FTRL versus committee-subsampled FIXED on AG News at $n=100$, $k'=10$, $T=100$ (matched to Table 2); 3×3 grid (rows: latency heterogeneity σ_{lat}; columns: data partition). Each point is mean $\pm$ std over 3 seeds at the exact $(\sigma_{\text{lat}}, \text{partition})$ cell, so the comparison is apples-to-apples within each panel. Committee subsampling at any $K \geq 2k'$ dominates DP-FTRL in every panel; the un-amplified Fixed baseline ($K=k'$) is the only configuration DP-FTRL beats. The gap widens with data heterogeneity (rightward across columns) and is largely invariant to σ_{lat} (downward across rows).

F Distributed noise protocol

This appendix provides a self-contained description of the cryptographic protocol underlying the privatiser architecture (§5.1), sufficient for replication.

F.1 Secure aggregation (SecAgg)

We assume familiarity with classical pairwise SecAgg [Bonawitz et al., 2017]: a protocol that lets a server recover $\sum_i x_i$ over k' clients without observing individual x_i, via antisymmetric pseudorandom masks that cancel under summation in $\mathbb{Z}_{\mathfrak{q}}$ (with SecAgg field modulus $\mathfrak{q}$, distinct from the DP subsampling rate q).

F.2 From pairwise masking to one-shot committee-based aggregation

The pairwise masking in classical SecAgg requires every pair of the K submitted clients to exchange keys, yielding $O(K^2)$ communication and at least three interaction rounds, and client synchronisation. For asynchronous FL this is a basic obstacle [Taiello et al., 2025], and recent one-shot committee-based protocols remove the need for synchronisation by adding a small committee of $m \ll n$ intermediate nodes, clients interact only with the committee and the committee jointly help the server aggregating without learning the individual inputs. We summarise three representative protocols, all candidate substrates for our privatiser architecture.

Willow [Bell-Clark et al., 2025]. Reduces client interaction to a single message via key-additive homomorphic encryption (KAHE). Each client samples a fresh KAHE key k_i and sends both $\mathrm{KAHE.Enc}(k_i, x_i)$ and an encryption of k_i under the committee's threshold public key. The server homomorphically aggregates both ciphertexts, the committee helps the server to recover $\sum_i k_i$, which deblinds the aggregate input ciphertext to yield $\sum_i x_i$. Trust: t-of-m honest committee. The committee is *persistent*, established once via distributed key generation and reused across aggregations.

OPA [Karthikeyan and Polychroniadou, 2025]. Achieves one-shot client interaction via a seed-homomorphic PRG (or threshold key-homomorphic PRF) instantiated from LWR. Clients mask inputs with PRG outputs and secret-share the seed among committee members, the committee combines partial seed evaluations to compute the aggregate mask without reconstructing individual seeds. Trust: t-of-m honest committee, with malicious-client security via abort. The committee is stateless by default but supports persistence when membership is fixed across aggregations.

Buffalo [Taiello et al., 2025]. A committee-based SecAgg protocol designed specifically for buffered asynchronous FL (FedBuff), handling dynamic client arrivals and departures natively within the buffer. Trust: t-of-m honest committee. Persistent committee within a training run.

Table 5: Committee-based SecAgg substrates relevant to our privatiser construction. Persistent committees naturally support our cross-aggregation budget ledger.

Protocol	Trust	Persistent committee	Client rounds
Willow [Bell-Clark et al., 2025]	$\geq t$ of m honest	yes	1
OPA [Karthikeyan and Polychroniadou, 2025]	$\geq t$ of m honest	toggleable	1
Buffalo [Taiello et al., 2025]	$\geq t$ of m honest	yes	1

From committee to privatiser. All three protocols expose a committee that participates in the aggregation pipeline and can therefore add a calibrated discrete Gaussian share to the aggregate before the server sees it (Algorithm 2). The architectural requirement specific to our framework is *persistent* committee state, since the per-client budget ledger $(\mu^2_{\text{spent},i}, \hat{q}_i)$ is updated across aggregations. Willow, persistent OPA, and Buffalo satisfy this directly.

Per-round rotation of the privatiser committee. The committee subsampling step (§5.2) is compatible with selecting a fresh privatiser set every round via the same beacon-driven ChooseSet

(see Section 4.3 [Ma et al., 2023]) primitive used to draw $\mathcal{S}_t$. Rotation of the privatisers, and their public state can be done using similar sharing part of Figure 12 [Ma et al., 2023].

F.3 Distributed discrete Gaussian mechanism

The discrete Gaussian distribution $\mathcal{N}_{\mathbb{Z}}(\mu, \sigma^2)$ over the integers has probability mass

$$\Pr[X = x] \propto \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right), \quad x \in \mathbb{Z}. \tag{21}$$

Kairouz et al. [2021a] show that the sum of m independent draws from $\mathcal{N}_{\mathbb{Z}}(0, \sigma^2/m)$ has its privacy loss bounded by that of a single continuous draw $\mathcal{N}(0, \sigma^2)$ (their Theorem 9, the discrete Gaussian family is not closed under convolution, but the *privacy loss* of the sum is dominated by the continuous-equivalent Gaussian). This *summation property* is the key enabler for distributed DP: if each of m committee members adds noise $\eta_j \sim \mathcal{N}_{\mathbb{Z}}(0, \widetilde{\sigma}_t^2/m)$, the aggregate noise $\sum_j \eta_j$ is privacy-equivalent to a single central draw of variance $\widetilde{\sigma}_t^2$.

Quantisation and integer-space noise. Before adding noise, each client quantises its clipped update $\bar{\Delta}_i$ to integers in $[-\Lambda, \Lambda]^d$ (typically $\Lambda = 2^{15}$) via stochastic rounding [Kairouz et al., 2021a]. The quantised update $\hat{\Delta}_i \in \mathbb{Z}^d$ is compatible with modular arithmetic in SecAgg. In integer space the per-client ℓ_2 sensitivity is $\Lambda\, q$ (with $q = k'/K$ the committee subsampling rate of §5.2, Λ otherwise), so the integer-space noise variance required for μ_t-GDP is

$$\widetilde{\sigma}_t^2 \;=\; (\Lambda\, q)^2/\mu_t^2, \tag{22}$$

and each committee member draws $\eta_j \sim \mathcal{N}_{\mathbb{Z}}(0, \widetilde{\sigma}_t^2/m)$. After dequantisation (dividing the aggregate by Λ/C), the continuous-space noise variance is $\sigma_t^2 = (C/\Lambda)^2\widetilde{\sigma}_t^2 = (C\, q)^2/\mu_t^2$, matching Eq. (1) in the main text. The quantisation error per coordinate is $O(C/\Lambda) \approx 3 \times 10^{-5}$, negligible relative to the DP noise. Calibrating noise in continuous units while sampling integer noise without the $(\Lambda/C)^2$ rescaling would under-noise the mechanism by exactly that factor and invalidate the privacy claim, our implementation takes the integer-space sensitivity Λ as input to avoid this pitfall.

F.4 Privatiser protocol (full specification)

Algorithm 2 gives the per-aggregation pipeline in full, including the committee subsampling step of §5.2. Roles in brackets indicate which party executes each step: SRV (the server), CLI i (client i), and PRIV (the privatiser committee, acting jointly under threshold cryptography). A one-time setup phase is assumed already complete: the privatiser committee holds threshold decryption-key shares (DKG, e.g. Flamingo's [Ma et al., 2023] §4.3) and is registered with the public randomness beacon [Cloudflare, 2024].

F.5 Deployment regimes: noise scaling versus trust

The privatiser committee can fail in two qualitatively different ways. *Liveness failure* (dropout): an honest privatiser goes offline, so its share never reaches the aggregate, the adversary learns nothing extra, but the committed noise must be repaired. *Confidentiality failure* (collusion): a malicious privatiser reveals its share to the adversary, the share still reaches the aggregate, but the adversary-unknown portion of the noise shrinks. Both failure modes are governed by a single design parameter: the per-share variance σ^2_{share} that each privatiser commits to at protocol setup. Three quantities depend on it:

$$\text{committed aggregate variance} \;=\; m \cdot \sigma^2_{\text{share}}, \tag{23}$$

$$\text{adversary-unknown variance} \;=\; |H| \cdot \sigma^2_{\text{share}}, \tag{24}$$

where $H \subseteq [m]$ is the set of privatisers neither colluding nor dropped. The privacy guarantee requires $|H| \cdot \sigma^2_{\text{share}} \geq \sigma_t^2$ in the worst case, utility experiences (23), which is what enters the SGD descent term in Theorem 7. Three regimes instantiate σ^2_{share} to span the trust spectrum.

Algorithm 2 One aggregation of the privatiser pipeline with committee subsampling.

Require: Round index t. Server holds model θ_t. Privatisers hold per-client ledger $\{\mu^2_{\text{spent},i}\}_{i=1}^n$ and, for buffer-aware spending, rate estimates $\{\hat{q}_i\}$. Buffer ratio $K \geq k'$, retirement threshold θ_{ret}, target μ^2_{budget}.

// Phase 1: client selection [SRV, CLI i]
SRV broadcasts θ_t.
for each client $i \in [n]$ in parallel as soon as θ_t arrives **do**
  $\Delta_i \leftarrow \theta_t^{(i)} - \theta_t$ {local SGD on $\mathcal{D}_i$}
  $\bar{\Delta}_i \leftarrow \Delta_i \cdot \min(1, C/\|\Delta_i\|_2)$ {clip}
  $\hat{\Delta}_i \leftarrow \text{STOCHASTICROUND}(\bar{\Delta}_i, \Lambda)$
  Submit $\text{SECAGG.ENCRYPT}(\hat{\Delta}_i)$ to SRV.
end for
SRV closes the buffer at the first K ciphertexts received: $\mathcal{B}_t \leftarrow \{i_1, \ldots, i_K\}$.
SRV publishes $(t, H(\mathcal{B}_t))$.
// Phase 2: buffer-id consistency check [PRIV]
for each privatiser $j \in [m]$ in parallel **do**
  Verify own view of $\mathcal{B}_t$ matches SRV's, sign $\text{SIG}_j(t, H(\mathcal{B}_t))$.
end for
Abort if fewer than $\lceil 2m/3 \rceil$ matching signatures (§5.3, equivocation defence).
// Phase 3: committee subsampling [PRIV]
Fetch the public beacon value v_t [Cloudflare, 2024].
$\mathcal{S}_t \leftarrow \text{CHOOSESET}(v_t, \mathcal{B}_t, k')$ {uniform k'-subset, deterministic given v_t}
// Phase 4: per-aggregation spend [PRIV]
Active subset: $\mathcal{S}'_t \leftarrow \{i \in \mathcal{S}_t : \mu^2_{\text{budget}} - \mu^2_{\text{spent},i} \geq \theta\, \mu^2_{\text{budget}}\}$. {retirement}
Pick μ_t^2 via spending policy π: $\mu_t^2 \leftarrow \pi(\{\mu^2_{\text{spent},i}\}_{i \in \mathcal{S}'_t})$. {FIXED: $\mu_t^2 = \mu^2_{\text{budget}}/(TK/n)$}
Safety cap: $\mu_t^2 \leftarrow \min\big(\mu_t^2,\, \min_{i \in \mathcal{S}'_t}(\mu^2_{\text{budget}} - \mu^2_{\text{spent},i})\big)$.
// Phase 5: noise injection [PRIV]
Continuous-equivalent aggregate noise variance with amplification: $\sigma_t^2 \leftarrow (k'C/K)^2/\mu_t^2$ {Eq. (1), $q = k'/K$, this is the variance after dequantisation.}
Integer-space per-share variance: $\widetilde{\sigma}^2_{t,j} \leftarrow (\Lambda\, q)^2 / \big(\mu_t^2 \cdot \lceil (1-\gamma_p)m \rceil\big)$ {$\Lambda = 2^{15}$ is the quantisation level (App. F); the integer-space sensitivity is $\Lambda\, q$, so $\widetilde{\sigma}^2 = (\Lambda/C)^2\sigma^2$.}
for each privatiser $j \in [m]$ in parallel **do**
  $\eta_{j,t} \sim \mathcal{N}_{\mathbb{Z}}\big(0,\, \widetilde{\sigma}^2_{t,j}\big)$ {d-dim discrete Gaussian, integer values}
  Submit $\text{SECAGG.ENCRYPT}(\eta_{j,t})$ to SRV.
end for
// Phase 6: partial aggregation and threshold decryption [SRV, PRIV]
SRV forms encrypted partial aggregate $C_t \leftarrow \sum_{i \in \mathcal{S}'_t} \text{SECAGG.ENCRYPT}(\hat{\Delta}_i) \boxplus \sum_{j=1}^m \text{SECAGG.ENCRYPT}(\eta_{j,t})$.
PRIV verify cross-check signatures match $(t, H(\mathcal{B}_t), H(\mathcal{S}_t), \mu_t^2)$, release threshold key shares.
SRV decrypts $A_t \leftarrow \text{SECAGG.DECRYPT}(C_t)$.
// Phase 7: model update and ledger [SRV, PRIV]
$\theta_{t+1} \leftarrow \theta_t + (1/k') \cdot \text{DEQUANT}(A_t, \Lambda, C)$.
for each $i \in \mathcal{S}'_t$ **do**
  $\mu^2_{\text{spent},i}$ += μ_t^2.
  Update $\hat{q}_i$ {maintained for buffer-aware spending; unused by FIXED}
end for

(R1) Trusted committee. All m privatisers are honest *and* online, so $|H| = m$. Setting $\sigma^2_{\text{share}} = \sigma_t^2/m$ gives a committed aggregate of exactly σ_t^2 and an adversary-unknown variance of exactly σ_t^2. Noise overhead is 1. This is the assumption made by DDP [Kairouz et al., 2021a] and the operating point adopted in our experiments (§7). It is appropriate when the committee is operated by a small set of trusted parties (e.g., the FL service provider plus one or two independent witnesses) and round-level liveness is guaranteed by the SecAgg layer.

(R2) Honest-with-detection (dropout-robust). Privatisers remain honest but may drop due to network failures or crashes, collusion is not in the threat model. The SecAgg layer [Bell-Clark et al., 2025, Karthikeyan and Polychroniadou, 2025, Taiello et al., 2025] detects dropout natively: a privatiser that fails to deliver its share within the round deadline is excluded from the aggregate. Two realisations recover the target noise variance: *(a) Pre-allocated margin.* Set $\sigma^2_{\text{share}} = \sigma_t^2/m_{\min}$, where $m_{\min} \le m$ is the deployment's guaranteed-online count. If $m_{\min}$ privatisers contribute, the adversary-unknown variance is exactly σ_t^2, more contributors yield mild over-noising of factor at most $m/m_{\min}$ in (23). Single round, no protocol change. *(b) Detect-and-resample.* σ^2_{share} starts at σ_t^2/m. After dropout detection the server announces the active count m_{active}, and survivors contribute a top-up share of variance $\sigma_t^2(m-m_{\text{active}})/(m\cdot m_{\text{active}})$ in a second round. Restores exact σ_t^2 aggregate at the cost of one additional round, with no extra noise in the common case.

(R3) Threshold-robust (collusion). At most γm privatisers may be semi-honestly colluding with $\gamma < 1 - t/m$, and their shares may be publicly revealed to the adversary. The threshold $t \le m$ is fixed at protocol setup (e.g., during distributed key generation in Willow / OPA / Buffalo) and cannot be lowered post-hoc, every privatiser always commits the same σ^2_{share} regardless of whether colluders actually appear in a given round. To keep (24) above σ_t^2 in the worst case,

$$\sigma^2_{\text{share}} \;=\; \sigma_t^2/t, \qquad \underbrace{m\cdot\sigma_t^2/t}_{\text{committed aggregate}} \;=\; (m/t)\,\sigma_t^2, \tag{25}$$

so the all-honest realisation injects an aggregate of $(m/t)\,\sigma_t^2$ into the SecAgg sum even when no privatiser actually colludes. The adversary's view is still covered by σ_t^2 of unknown noise, so the per-client DP guarantee is preserved exactly, the gap between (23) and (24) is the price the deployment pays for being robust against the worst-case collusion pattern. R3 collapses onto R1 only in the limit $t \to m$. The choice of t is substrate-dependent: the threshold primitives used in the SecAgg layer protect confidentiality against up to $t-1$ corrupted privatisers in the semi-honest model, and the standard honest-majority convention [Cramer et al., 2015], also adopted by the substrates we consider [Bell-Clark et al., 2025, Karthikeyan and Polychroniadou, 2025, Taiello et al., 2025], sets $t > m/2$ (e.g., $t \ge 6$ for $m = 10$), below this convention the primitive remains correct but the deployment loses the honest-majority security guarantee usually expected of it.

Algorithmic invariance. The three regimes are identical apart from the σ^2_{share} constant. All proofs in Appendix A and all spending-policy results in §5 go through unchanged with σ_t^2 replaced by the regime's adversary-unknown variance (24). The privacy accountant always calibrates against (24), the convergence bound always sees (23). R1 sets these equal, R2 keeps them equal in the common case, and R3 separates them by the m/t multiplier in (25).

Empirical noise overhead. Figure 3 reports the utility cost of the m/t multiplier on AG News under the safe FIXED policy at production-realistic latency heterogeneity ($\sigma_{\text{lat}} = 1.5$, Dir(0.5) correlated, $\varepsilon_{\text{target}} = 10$, $n = 100$, $k' = 10$, $T = 100$, 3 seeds, $m = 10$ privatisers). The privacy guarantee is identical at every t because (24) is held at σ_t^2 by construction, only (23), and hence the noise the trained model experiences, scales as m/t. We sweep four buffer ratios $K \in \{k', 2k', 5k', 8k'\}$ to expose the amplification-trust interaction.

The drop in best-accuracy relative to R1 ($t = m = 10$) is mild and grows with both $1/t$ and $1/q$ (less amplification makes the per-aggregation noise larger in absolute terms, so the additional m/t multiplier hurts more):

- $K = k'$ **($q = 1$, no amp):** -1.12, -2.59, -3.89, -4.51, -5.21 pp at $t = 9, 8, 7, 6, 5$.
- $K = 2k'$ **($q = 0.5$):** -0.09, -0.64, -1.64, -2.44, -3.72 pp.
- $K = 5k'$ **($q = 0.2$):** -0.54, -1.37, -2.29, -2.92, -3.92 pp.
- $K = 8k'$ **($q = 0.125$):** -0.31, -0.71, -1.26, -2.06, -3.07 pp.

At the honest-majority cryptographic floor $t = 6$ the worst-case cost is ~ 4.5 pp at $K = k'$ and shrinks to ~ 2 pp once amplification is enabled ($K \ge 2k'$). Below $t = 6$ (shown for trend only) the deployment loses the standard honest-majority security guarantee. Across the whole sweep the privacy guarantee is preserved ($\varepsilon_{\max} \le \varepsilon_{\text{target}}$) by construction.

Choosing a regime. R1 is the cheapest and matches the experimental setup of this paper. R2 adds liveness robustness at near-zero noise cost (pre-allocated margin) or one extra round (detect-and-resample) without changing the trust assumption. R3 adds robustness to semi-honest collusion at the cost of m/t extra noise, for $m=10$ the honest-majority floor $t=6$ costs ~ 2 pp on AG News under amplification ($K \geq 2k'$), rising to ~ 4.5 pp in the no-amplification regime. Stronger amplification reduces the absolute extra noise that an m/t multiplier introduces, so deployments adopting the threshold-robust regime recover most of that overhead by raising K.

F.6 Comparison with existing distributed DP mechanisms

Table 6 compares our framework with the two prior distributed DP mechanisms for FL on the dimensions that matter for buffered async deployment. The SecAgg substrate comparison (Willow / OPA / Buffalo) is in Table 5.

Table 6: Distributed DP mechanisms for federated learning.

	Noise injector	Trust requirement	Adaptive σ	Async-compatible
DDP [Kairouz et al., 2021a]	all K buffer clients	all K honest	no	yes
DP-FTRL [Kairouz et al., 2021b]	server	trusted server	yes	no
Ours	committee of m	$\geq t$ of m honest	yes	yes

vs. DDP [Kairouz et al., 2021a]. Same noise model (discrete Gaussian) and SecAgg compatibility. DDP commits per-client noise variance σ_t^2/k before the buffer is known and requires every selected client to add noise honestly. We move noise to the committee, removing the client-honest-noise assumption and exposing μ_t^2 as a buffer-dependent quantity (§5).

vs. DP-FTRL [Kairouz et al., 2021b]. DP-FTRL uses correlated tree-aggregated noise across aggregations, achieving tighter composition when $n/T > 10$, but requires a trusted server (the server must see individual updates to apply the correlated structure) and is stateful across aggregations, both incompatible with standard SecAgg. Our protocol is stateless per aggregation and operates under SecAgg.

G Experimental details

G.1 Hyperparameters

All training experiments use SGD with learning rate $\eta = 0.01$, gradient clipping norm $C = 1.0$, $\delta = 10^{-5}$, retirement threshold $\theta_{\text{ret}} = 0.05$, server momentum 0.9, $\Lambda = 2^{15}$ quantisation levels.

Wall-clock-vs-amplification sweep (Table 2, §7.3). $n = 100$, $k' = 10$, latency-correlated Dirichlet partitions, sweeping $\varepsilon_{\text{target}} \in \{5, 10, 15\}$, buffer ratio $K \in \{k', 2k', 4k', n\}$ (i.e. $\{10, 20, 40, 100\}$), partition $\in \{\text{IID}, \text{Dir}(0.5), \text{Dir}(0.1)\}$, and $\sigma_{\text{lat}} \in \{0.5, 1.0, 1.5\}$, 3 seeds. Per-dataset T and model:

Dataset	T	params	Model
CIFAR-10 (pretrained)	100	5,130	frozen ResNet-18 + linear head ($512 \to 10$)
AG News	100	$\sim$3K	frozen DistilBERT + linear head ($768 \to 4$)
DBpedia	100	$\sim$11K	frozen DistilBERT + linear head ($768 \to 14$)

Parameter-space canary audit (Figure 2, §7.2). CIFAR-10 with frozen ResNet-18 features and a trainable linear head ($512 \to 10$, 5,130 params), $n = 1000$, $k' = 100$, $T = 600$, $\varepsilon_{\text{target}} = 10$, $K = 200$ ($q = 0.5$), server momentum 0.9, 200 canary directions $+ 2{,}000$ shadow directions (parameter-space cosine attack of Andrew et al. [2024]), 3 seeds. Sweeps $\sigma_{\text{lat}} \in \{0.5, 1.0, 1.5, 2.0\}$ across UNSAFE, FIXED (with retirement).

Loss-based canary companion (Figure 4 of Appendix C). Same dataset/model/scale as the parameter-space audit, but with 50 mislabelled canaries per client (instead of direction vectors) and seeds $\{5, 10, 15, 20\}$.

Compute resources. All training experiments (utility design-space sweep, parameter-space canary audit, additional ablations) were run on a single NVIDIA A100 (40 GB) GPU. The privacy-only experiments (the scale-violation simulation in Appendix B and the threat-model utility ablation in Appendix F.5) require no model training and were run on CPU. Typical end-to-end cost per training cell ranges from a few minutes to roughly an hour at $T = 100$ on the A100, depending on dataset size and head dimensionality. The full sweep underlying Table 2 (3 datasets, 3 ε, 4 buffer ratios, 3 partitions, 3 σ_{lat}, 3 seeds) totalled approximately 90 GPU-hours.

G.2 Model architectures

CIFAR-10 pretrained. Frozen ResNet-18 [He et al., 2016] backbone (ImageNet weights) with a trainable linear head ($512 \to 10$, 5,130 params). Features are pre-extracted once and cached.

AG News. Frozen DistilBERT [Sanh et al., 2019] encoder (base-uncased, 66M frozen params). The $[\text{CLS}]$ embedding (768-dim) is extracted once and cached, and a trainable linear head $\text{FC}(768, 4)$ is added on top ($\sim$ 3,076 params).

DBpedia. Same DistilBERT $[\text{CLS}]$ feature extractor, with a trainable linear head $\text{FC}(768, 14)$ ($\sim$ 10,766 params).

CIFAR-100 pretrained (used only in the privacy-only / correlation analyses of Appendix B, not in the utility table). Frozen ResNet-18 [He et al., 2016] backbone (ImageNet weights) with a trainable linear head ($512 \to 100$, 51,300 params).

G.3 Data partitioning

IID. Training samples are shuffled and split evenly across n clients.

Dirichlet. Each client draws a class proportion vector $\mathbf{p}_i \sim \text{Dir}(\alpha)$ and receives samples according to $\mathbf{p}_i$ [Li et al., 2020]. Lower α produces more skewed partitions.

Correlated partition. Clients are sorted by latency rate λ_i (fastest first). The Dirichlet allocation assigns the first classes to fast clients and later classes to slow clients, so that device speed correlates with data distribution. This models the documented entanglement of systems and statistical heterogeneity in production FL [Kairouz et al., 2021c, Bonawitz et al., 2019, Cho et al., 2022].

G.4 Scale simulation methodology

Table 1 uses privacy-only accounting simulations (no model training). For each scale (n, T), we sample latency rates $\lambda_i \sim \text{LogNormal}(0, 1.5^2)$ for $i \in [n]$, simulate T aggregations of buffered selection (first $K = n/10$ responses), and compute per-client participation counts N_i. The true per-client ε_i is obtained via GDP composition over N_i aggregations, each with per-aggregation cost $\zeta_t = C^2/(2\sigma^2 K)$ calibrated for a system-level $\varepsilon_{\text{target}} = 8$. No gradient computation or model training is performed, the violation depends only on participation counts.

G.5 Privatiser overhead analysis

We analyse the cost of the privatiser layer relative to the *existing* cost of one aggregation of FedBuff with DDP [Kairouz et al., 2021a, Nguyen et al., 2022]. The baseline per-aggregation cost is dominated by: (a) each of K buffer clients performing local SGD and submitting a d-dimensional quantised update through SecAgg, and (b) the SecAgg protocol itself (key exchange, masking, aggregation). We show that our layer adds $O(K)$ scalar operations and zero additional communication aggregations.

Computation (committee side). The privatiser committee performs four operations per aggregation, the first three of which are independent of the model dimension d: *(i)* **Budget lookup and retirement check**: for each client $i \in \mathcal{B}_t$, read $\mu^2_{\text{spent},i}$ and test whether $\mu^2_{\text{budget}} - \mu^2_{\text{spent},i} < \theta_{\text{ret}} \cdot \mu^2_{\text{budget}}$. Cost: K comparisons. *(ii)* **Rate estimation update** (used only by buffer-aware spending policies; a no-op under FIXED): update $\hat{q}_i$ via exponential moving average. Cost: K multiply-adds. *(iii)* **Noise calibration**: under FIXED, $\mu_t^2 = \mu^2_{\text{budget}}/(TK/n)$ is a constant, the spending step reduces to one safety-cap evaluation (Eq. (2)), $O(k')$ comparisons. *(iv)* **Noise sampling**: each of m privatisers draws $\eta_j \sim \mathcal{N}_{\mathbb{Z}}(0,\, \sigma_t^2/m)$ as a d-dimensional vector over $\mathbb{Z}_{\mathfrak{q}}$. Cost: $O(d)$ per privatiser.

Operations (i)–(iii) total $O(K)$ scalar operations. At $K = 200$, this takes $< 10\,\mu$s on modern hardware, negligible relative to the $\sim$1 s wall-clock time of a FedBuff aggregation (dominated by client training and network transfer).

Operation (iv) is the only d-dependent cost, and it is *not new*: in standard DDP, each of k' clients samples $\mathcal{N}_{\mathbb{Z}}(0, \widetilde{\sigma}_t^2/k')$ and adds it to its update before SecAgg submission. We replace these k' client-side samples with m committee-side samples. The aggregate privacy guarantee is unchanged (Eq. (22) and the privacy-equivalent central-Gaussian bound of Appendix F.2), we merely relocate where the noise is generated. When $m < k$ (typical: $m \approx 3$–10 for threshold committees), the committee samples *fewer* noise vectors than DDP requires from clients.

Communication. The privatiser layer requires *zero additional communication aggregations* under any of the committee-based SecAgg protocols described in §F.2:

Willow [Bell-Clark et al., 2025]: each privatiser j adds η_j to its partial decryption of the aggregate KAHE ciphertext, so the server recovers $\sum_i x_i + \sum_j \eta_j$ instead of $\sum_i x_i$. The committee's message size is unchanged: one polynomial in R_q per member, independent of n and d.

OPA [Karthikeyan and Polychroniadou, 2025]: each committee member adds η_j to its partial mask evaluation before sending to the server, so the unmasked aggregate becomes $\sum_i x_i + \sum_j \eta_j$ with no change to the protocol structure.

Buffalo [Taiello et al., 2025]: noise injection occurs within the committee's aggregation step for each buffer, the protocol processes each buffer independently and naturally accommodates the per-aggregation σ_t^2 calibration.

The only new communication is a broadcast of the noise parameter σ_t^2 and the active aggregated subset $\mathcal{S}_t'$ from the committee to the server: $k+1$ scalars per aggregation. At $d \geq 10^4$ (typical model sizes), this is $< 0.01\%$ of the per-aggregation bandwidth.

Client-side changes. Clients are *unmodified*. In DDP, each client clips, quantises, adds local noise, and submits through SecAgg. Under our protocol, the local noise addition is removed: the client clips, quantises, and submits. This strictly *reduces* per-client computation (no sampling of $\mathcal{N}_{\mathbb{Z}}$) and simplifies the client implementation. Clients need not know the per-aggregation σ_t^2, the buffer composition, or whether they are close to retirement. The protocol is transparent to clients.

Storage. Each privatiser maintains per-client state: $\mu^2_{\text{spent},i} \in \mathbb{R}$ (8 bytes) and $\hat{q}_i \in \mathbb{R}$ (8 bytes) for each of n clients, plus a retirement flag (1 byte). Total: $17n$ bytes, replicated across m committee members. Table 7 gives concrete numbers.

Table 7: Privatiser storage cost per committee member.

n (clients)	Storage	Context
10^3	17 KB	research prototype
10^5	1.7 MB	production cross-device
10^6	17 MB	Gboard scale
10^7	170 MB	large-scale deployment

For comparison, the model parameters in a typical production FL task occupy $O(d \cdot 4)$ bytes (≈ 400 MB for $d = 10^8$). The per-client state is orders of magnitude smaller than the model itself.

Security considerations. The per-client state $(\mu^2_{\text{spent},i}, \hat{q}_i)$ is derived entirely from information already available to the committee: client identities in each buffer $\mathcal{B}_t$ (visible to the server in any SecAgg protocol) and the per-aggregation noise parameter μ_t^2 (a public protocol parameter). No information from individual client updates is used to compute the state. In particular, the retirement decision for client i depends only on $\mu^2_{\text{spent},i}$ (a running sum of public μ_t^2 values weighted by participation) and the threshold θ (a public hyperparameter). A semi-honest server that observes which clients are retired learns only that their cumulative participation cost approached μ^2_{budget}, information already inferable from the public buffer history $\{\mathcal{B}_t\}_{t=1}^T$. The retirement pattern therefore leaks no information beyond what the server already observes.

Comparison with alternatives. DP-FTRL [Kairouz et al., 2021b] requires the server to maintain a tree-structured noise state of size $O(d \log T)$ and to observe *individual* client updates (incompatible with SecAgg). Our approach requires $O(n)$ scalar state (independent of d) and operates entirely within SecAgg. Standard DDP [Kairouz et al., 2021a] has no per-client state but also no per-client accounting: the privacy guarantee is the uniform-participation approximation that our work identifies as flawed. Our overhead ($O(k)$ compute and $O(n)$ storage) is the minimal cost of *any* scheme that tracks per-client privacy budgets in the distributed DP model.

H Accounting design choices

GDP-to-(ε, δ) conversion. A μ-GDP mechanism (§3.2) satisfies (ε, δ)-DP for the closed-form $\delta(\varepsilon) = \Phi(-\varepsilon/\mu + \mu/2) - e^{\varepsilon}\,\Phi(-\varepsilon/\mu - \mu/2)$, where Φ is the standard-normal CDF [Dong et al., 2022], the inverse $\varepsilon(\mu, \delta)$ yields the per-client budget μ^2_{budget} used throughout the paper.

$$\delta(\varepsilon) = \Phi\Big(-\tfrac{\varepsilon}{\mu} + \tfrac{\mu}{2}\Big) - e^{\varepsilon}\,\Phi\Big(-\tfrac{\varepsilon}{\mu} - \tfrac{\mu}{2}\Big). \tag{26}$$

Subsampling amplification: where it does and does not apply. In synchronous FL with uniform random client selection ($q = k/n$), the per-aggregation cost benefits from privacy amplification by subsampling [Balle et al., 2018], substantially tightening composition. Buffered asynchronous selection at the population layer is *not* Poisson: the first K responders form the buffer, inclusion probabilities q_i depend on device latency, and they vary across clients (§4.2). Standard amplification results [Balle et al., 2018, Mironov, 2017] require identical, independent inclusion probabilities and do not apply at this layer.

Our framework recovers rigorous amplification *at the buffer-to-aggregate step* via committee subsampling (§5.2): the privatiser committee draws a uniformly random k'-subset $\mathcal{S}_t \subseteq \mathcal{B}_t$ from a public randomness beacon, and each client $i \in \mathcal{B}_t$ appears in $\mathcal{S}_t$ with probability exactly $q = k'/K$. Conditional on the buffer $\mathcal{B}_t$, this is a uniform Bernoulli inclusion independent of client identity or latency, so the subsampled-Gaussian RDP bound applies and the per-aggregation cost is reduced by q^2 (Theorem 3, Appendix A.3). The q in our accounting is therefore k'/K, set by the operator's choice of buffer ratio, not the population-level rate k/n.

Population-layer amplification (treating each client's average participation rate q_i as if it were a uniform Poisson rate) remains an open problem for heterogeneous async, we charge participating clients the full unamplified-at-the-population-layer cost (multiplied by the q^2 reduction at the buffer-to-aggregate layer), making our reported ε_i a conservative upper bound on the true privacy cost.

GDP vs RDP vs PLD. We choose GDP over Rényi DP (RDP) [Mironov, 2017] and Privacy Loss Distributions (PLD) [Koskela et al., 2020] for two reasons. First, μ^2 composition is exactly additive, making per-client tracking a single floating-point addition per aggregation per client, RDP requires optimising over the Rényi order α at each query, and PLD requires maintaining per-client discretised loss distributions ($O(n \cdot B)$ storage for B bins). Second, the (ε, δ)-DP conversion (Eq. 26) is closed-form, so the safety cap can be evaluated without numerical optimisation. The cost of this choice is that GDP composition may be slightly looser than PLD for non-Gaussian mechanisms, for the discrete Gaussian mechanism we use [Kairouz et al., 2021a], the gap is negligible [Dong et al., 2022].

I Wall-clock and amplification trade-off

Why amplification is the missing piece in async DP-FL. Privacy amplification by subsampling is the workhorse of modern DP training: the Poisson-subsampled Gaussian mechanism underlying DP-SGD [Abadi et al., 2016] relies on each record being included in a minibatch with a known, fixed probability q, and the resulting per-step privacy cost shrinks proportionally to q [Kasiviswanathan et al., 2011, Balle et al., 2018, Mironov, 2017, Wang et al., 2019]. Synchronous DP-FedAvg inherits this benefit when the server samples a uniform k'-of-n subset of clients per round, with $q = k'/n$ [McMahan et al., 2018a, Kairouz et al., 2021c]. In buffered asynchronous FL, by contrast, the server does not control which clients participate: the buffer fills with whichever clients finish their local work first, so participation is deterministic given hardware rather than uniformly

random [Nguyen et al., 2022, Huba et al., 2022, Xie et al., 2019, Wang et al., 2022]. Fast clients are systematically over-represented and slow clients under-represented, breaking both prerequisites of the standard amplification proof: clients do not appear with a known probability, and consecutive rounds are not independent draws.

Existing routes around the gap. The async-DP literature has so far avoided the problem rather than solved it. One line of work downgrades the privacy granularity to local DP, where each client perturbs its update before transmission and amplification at the server is no longer needed, this includes DP-FedBuff with local DP [lrd, 2025], multi-stage adaptive DP for async edge training [Liu et al., 2024], and the original async-DP edge proposals [Lu et al., 2020]. These designs lose the $1/\varepsilon$ noise scaling of central DP and tolerate large ε to recover utility. A second line replaces Poisson sampling with a different randomisation primitive: *random check-ins* [Balle et al., 2020] have each client self-sample a single round inside a fixed window, recovering a clean Bernoulli structure but constraining each client to one participation per window, *shuffled check-ins* [Girgis et al., 2021] extend this with a trusted shuffler. Neither covers the streaming buffered-async setting where fast clients contribute repeatedly. A third line drops amplification entirely and substitutes correlated noise via tree aggregation or matrix factorisation, as in DP-FTRL [Kairouz et al., 2021b], multi-epoch matrix mechanisms [Choquette-Choo et al., 2023b], and the banded variant [Choquette-Choo et al., 2023a], these are deployed in production [McMahan and Xu, 2023] but are designed for synchronous federations and do not address the asynchronous participation pattern of FedBuff-style systems. In short, prior work covers either (a) sync FL with rigorous amplification, (b) async FL with weaker (local) granularity, or (c) sync FL with no amplification, rigorous user-level amplification for buffered async FL is the cell that remained empty.

The committee subselection step (§5.1) draws a uniform k'-subset out of the K-sized latency-biased buffer (fixed-size sampling without replacement, marginal inclusion rate $q = k'/K$), yielding rigorous subsampled-Gaussian amplification at the buffer-to-aggregate step [Wang et al., 2019]. The amplification rate is a free design parameter of the deployment: it is set by the operator's choice of K, the buffer size that the privatiser committee waits for before authorising decryption. Larger K shrinks σ by q, so each round delivers more useful signal, smaller K shortens the per-round wall-clock, since the round completes once the K-th client arrives rather than the n-th.

Table 2 reports this trade-off under the safe *fixed* spending policy across three datasets (CIFAR-10, AG News, DBpedia), three target privacy budgets ($\varepsilon \in \{5, 10, 15\}$), and four buffer ratios ($K \in \{k', 2k', 4k', n\}$ for $k' = 10, n = 100$). Each accuracy is the mean of best-accuracy across 3 partitions $\times$ 3 latency heterogeneities $\times$ 3 seeds. The wall-clock column reports total simulation time at $T_{\max}$ rounds (in latency units), normalised per dataset to the $K = k'$ baseline.

Two patterns are visible across all three datasets and all three ε values. First, accuracy rises monotonically with K: at $\varepsilon = 10$, fixed gains $+33$ pp on CIFAR-10 ($0.320 \to 0.650$), $+24$ pp on AG News ($0.556 \to 0.798$), and $+38$ pp on DBpedia ($0.562 \to 0.943$) when the operator pays the cost of $K = n$. Second, the wall-clock cost grows faster than linearly in K: under heavy-tailed latency distributions, the K-th order statistic is dominated by the slowest of K clients, so $K = n$ is roughly $15-19\times$ the $K = k'$ wall-clock. Intermediate $K \in \{2k', 4k'\}$ recovers most of the accuracy gain at a fraction of the wall-clock cost.

I.1 Stratified results

The averaged numbers in Table 2 hide two pieces of structure that we expose here. First, the wall-clock-vs-K *multiplier* is approximately invariant to latency heterogeneity σ_{lat} (Table 8), only the absolute wall-clock changes. Second, accuracy under amplification is invariant to σ_{lat} on IID partitions but degrades on correlated Dirichlet partitions as σ_{lat} grows (Table 9), this residual coupling between latency and data heterogeneity is a small remaining gap of the FIXED default.

J Buffer-aware spending: preliminary prototype

An early exploration of buffer-aware spending policies is explroed in thsi section. At every aggregation, sets μ_t^2 to a robust statistic of the active aggregated subset's ledger entries $\{\mu^2_{\text{ideal},i,t}\}_{i \in \mathcal{S}'_t}$, where

Table 8: Wall-clock multiplier $\mathrm{wc}(K)/\mathrm{wc}(K{=}k')$ under FIXED, $\varepsilon = 10$, stratified by latency heterogeneity σ_{lat}. The multiplier is approximately invariant to σ_{lat} (the K-th order statistic scales with the same shape regardless of latency dispersion), the absolute wall-clock grows with σ_{lat} but the relative cost of larger K does not.

Dataset	σ_{lat}	$\frac{wc(2k')}{wc(k')}$	$\frac{wc(4k')}{wc(k')}$	$\frac{wc(n)}{wc(k')}$
CIFAR-10	0.5	2.18×	4.78×	22.3×
	1.0	2.03×	4.51×	18.9×
	1.5	1.67×	4.62×	17.5×
AG News	0.5	1.87×	3.80×	17.0×
	1.0	1.68×	3.55×	14.5×
	1.5	1.82×	3.72×	14.6×
DBpedia	0.5	1.95×	3.88×	19.3×
	1.0	2.04×	3.83×	17.9×
	1.5	2.78×	4.71×	19.5×

Table 9: Accuracy under FIXED at $K = 2k'$, $\varepsilon = 10$, stratified by partition and latency heterogeneity σ_{lat}, 3 seeds. IID accuracy is invariant to σ_{lat} (amplification + uniform data renders fast/slow client identity irrelevant), accuracy on correlated Dirichlet partitions degrades as σ_{lat} grows because retirement of fast clients biases the active pool away from classes those clients hold.

Dataset	σ_{lat}	IID	Dir(0.5)	Dir(0.1)
CIFAR-10	0.5	0.468	0.466	0.410
	1.0	0.466	0.436	0.367
	1.5	0.466	0.403	0.351
AG News	0.5	0.714	0.680	0.626
	1.0	0.714	0.638	0.594
	1.5	0.713	0.622	0.565
DBpedia	0.5	0.828	0.805	0.698
	1.0	0.828	0.793	0.684
	1.5	0.827	0.758	0.643

$\mu^2_{\mathrm{ideal},i,t}$ spreads client i's remaining budget over its expected future participations.[3] The prototype runs *without* committee subsampling ($K = k'$), so its setting is the un-amplified regime (ε=20 to avoid large sums of noise); we report it here as evidence that the future-work direction (§5) admits non-trivial designs. Table 10 reports the final accuracy and federation-time speedup against FIXED.

Table 10: **Final accuracy** and federation-time speedup of the buffer-aware (BA) prototype over FIXED (Fix.) at the same per-client privacy ceiling. Bold = best safe-policy final accuracy (within 0.5 pp). EMNIST is included as a fourth dataset to widen the prototype evidence; the main-text experiments use the three-dataset subset (CIFAR-10, AG News, DBpedia).

		AG News			**DBpedia**			**CIFAR-10**			**EMNIST**		
σ_{lat}	Part.	Fix.	BA	SpUp	Fix.	BA	SpUp	Fix.	BA	SpUp	Fix.	BA	SpUp
0.5	IID	**.854**±.000	**.843**±.000	6.3×	**.973**±.002	**.969**±.002	4.4×	**.710**±.002	.670±.002	6.3×	**.674**±.002	.681±.001	1.2×
	Dir(0.5)	.691±.003	**.847**±.002	3.9×	.954±.002	**.968**±.003	6.6×	.591±.003	**.675**±.001	4.7×	.647±.002	**.657**±.002	1.2×
	Dir(0.1)	.682±.002	**.846**±.002	3.9×	.942±.002	**.965**±.001	5.5×	.620±.002	**.679**±.003	4.5×	.621±.003	**.643**±.002	1.3×
1.0	IID	**.853**±.002	**.842**±.000	30×	**.973**±.002	**.968**±.000	19×	**.707**±.004	.671±.000	27×	**.661**±.003	**.672**±.000	6.9×
	Dir(0.5)	.756±.000	**.814**±.000	13×	.946±.002	**.965**±.003	34×	.586±.003	**.661**±.003	20×	**.438**±.003	**.447**±.003	6.4×
	Dir(0.1)	.699±.000	**.832**±.000	16×	.926±.003	**.959**±.002	21×	.614±.003	**.666**±.003	18×	.393±.003	**.440**±.003	5.6×
1.5	IID	**.853**±.000	**.840**±.000	192×	**.973**±.002	**.968**±.002	115×	**.709**±.003	.668±.002	154×	**.655**±.003	**.657**±.002	33×
	Dir(0.5)	.728±.002	**.798**±.002	76×	.946±.003	**.961**±.002	221×	.601±.002	**.649**±.003	97×	.424±.002	**.493**±.002	37×
	Dir(0.1)	.753±.002	**.817**±.003	117×	.914±.002	**.952**±.003	94×	.617±.002	**.656**±.002	85×	.378±.002	**.439**±.002	40×

[3]The specific statistic in this prototype is the median, robust to outliers in both tails of the active-buffer distribution.

At matched per-client privacy, the prototype matches FIXED on IID partitions (within 1.3 pp) and delivers $+1.4$ to $+16$ pp on correlated Dir(α) partitions with $\alpha \leq 0.5$, with 3 to $220\times$ federation-time speedups depending on σ_{lat}. Two caveats. First, the speedups derive from the un-amplified regime ($K = k'$); under committee subsampling the absolute σ shrinks for any policy and the accuracy gap collapses (§7.3), so the prototype is most informative in low-K, high-heterogeneity, wall-clock-bound deployments. Second, this is a single design point in a larger space; characterising which buffer-aware policies pay off, and in what amplification regime, is the motivation of the future work pointed to from §5.

K Limitations and broader impact

K.1 Limitations

Empirical scope. The evaluation has two halves, matching the two contributions of the framework. Safety: the per-client violation under naive accounting and its removal under FIXED, reported across federation scale and latency heterogeneity in Table 1 and Appendix B, and audited empirically by the parameter-space canary in §7.2. Utility: the wall-clock $\leftrightarrow$ accuracy trade-off under committee subsampling, covering three datasets (CIFAR-10, AG News, DBpedia), all with frozen-backbone linear heads ($n = 100$, $k' = 10$, $T = 100$), three seeds, and sweep $K \in \{k', 2k', 4k', n\}$ at three privacy targets $\varepsilon \in \{5, 10, 15\}$ (Table 2). Production-scale validation ($n \geq 10^6$) is out of scope for the current evaluation: the privacy-only scaling simulation (Table 1, Appendix B) extends to that regime without model training, but committee-protocol overhead at that scale is studied analytically rather than empirically (Appendix G.5).

Threat-model boundaries. We inherit the static-adversary model from one-shot SecAgg substrates [Bell-Clark et al., 2025, Karthikeyan and Polychroniadou, 2025, Taiello et al., 2025, Ma et al., 2023]: the corrupted set γ_p is fixed at protocol start. Adaptive corruption that changes the corrupted set mid-round is out of scope and would require the adaptive variants of Flamingo [Ma et al., 2023] or the dynamic state-transfer of Ball et al. [2024]. We assume the public randomness beacon emits values independent of the training history $(\mathcal{D}, \theta_{\leq t})$, in practice the drand / League-of-Entropy threshold-BLS construction satisfies this [Cloudflare, 2024], but a malicious beacon would break the amplification claim. Poisoning and denial-of-service attacks on the underlying SecAgg substrate are also outside our scope.

Theoretical scope. The amplification result (Theorem 3) applies at the buffer-to-aggregate step. Population-layer amplification (treating each client's expected participation rate as if it were a uniform Poisson rate) remains an open problem under heterogeneous async, and we charge the full unamplified-at-population-layer cost to the per-client ledger, this makes our reported ε_i a conservative upper bound (Appendix H). Our convergence statement inherits clipped DP-SGD bounds and assumes L-smoothness, bounded SGD variance, and bounded gradient heterogeneity (Appendix A.5), finer analysis under correlated participation across rounds and under buffer-induced staleness is left for follow-up work.

Buffer-aware spending policies. We evaluate the framework with a single safe spending policy, FIXED, which sets μ_t^2 to a constant calibrated for the average per-aggregation participation rate. The architecture supports a strictly larger design space: because the per-client ledger $(\mu^2_{\text{spent},i}, \hat{q}_i)$ is committee-resident and updated every aggregation, any function from the active buffer's ledger state to a non-negative μ_t^2 is a candidate spending policy, and budget compliance follows from the safety cap of Theorem 4 regardless of the choice of function. Identifying buffer-aware spending policies that exploit this freedom (for instance to allocate more budget to rounds whose buffer composition predicts low retirement bias) and characterising the regimes in which they improve on FIXED is left for follow-up work; a preliminary single-policy prototype is reported in Appendix J.

K.2 Broader impact

Closing a deployment-realistic privacy gap. The framework targets production cross-device FL deployments that today choose between (i) standard DP-FedAvg with unenforced uniform-participation accounting (silently violating per-client privacy under heterogeneous arrival), or (ii)

DP-FTRL (incompatible with secure aggregation, requires central trust). Our committee-driven amplification keeps the SecAgg trust model intact while restoring rigorous per-client guarantees, removing the trade-off operators face between meaningful privacy claims and deployment realism.

Disparate vulnerability. A core motivation is the disparate-vulnerability framing of Kulynych et al. [2022]: when device speed correlates with demographics [Bonawitz et al., 2019, Cho et al., 2022], uniform accounting transfers extra information about the most-active subgroups. Our framework closes this empirical gap (Figure 2) and lets operators enforce $\varepsilon_i \leq \varepsilon_{\text{target}}$ for every individual, rather than reporting a population-average ε_{sys} that hides the disparity.

Risks. The same primitive can in principle be deployed by adversaries who claim DP-style privacy guarantees while operating in threat models that depart from ours (e.g. trusting a single party with the privatiser ledger). We emphasise that the guarantee depends on the threshold-trust assumption $\gamma_p < 1/3$ on the privatiser committee, the integrity of the public randomness beacon, and the underlying SecAgg substrate, weakening any of these collapses the claim. Practitioners adopting the framework should be explicit about which threat model they assume and avoid presenting our (ε_i, δ)-DP guarantee as a categorical privacy claim divorced from those assumptions.

Energy and compute. Our experiments totalled approximately 30 A100 GPU-hours (Appendix G), preliminary and discarded configurations not reported in the paper add an estimated $\sim 2\times$ overhead. This is small relative to typical DP-FL evaluation budgets and we expect production adoption (where the privatiser committee runs alongside existing SecAgg infrastructure) to add less than constant overhead per aggregation (Appendix G.5).

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and §1 state three contributions (per-client violation in buffered async DP-FL, the privatiser-committee architecture with safety cap, and the first user-level subsampling-amplification result for buffered async DP-FL); all three are formalised in §4–§5.3 and validated in §7.

   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: A dedicated limitations paragraph discusses the trust assumption on the privatiser committee, the public-randomness-beacon dependency, and the latency model.

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: Assumptions are stated inline with each theorem in §6; full proofs of the per-client guarantee, amplification theorem, and convergence bound are in Appendices A.2, A.3, A.5.

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: Datasets, model architectures, optimiser, hyperparameters, latency model, and seeds are reported in §7 and Appendix G; the full code is available via an anonymous GitHub link.

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: Code is released through an anonymous GitHub repository linked from the paper, with reproduction commands for each table and figure; all datasets used (CIFAR-10, AG News, DBpedia) are publicly available and accessed via standard loaders.

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: Splits, model, optimiser, learning rate, batch size, clipping bound, noise calibration, committee size, and latency parameters are specified in §7 and Appendix G.

   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: Accuracy and wall-clock numbers report mean and standard deviation across 3 seeds (and across partitions and latency heterogeneities); canary AUC is reported per stratum.

   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: Compute setup (single-GPU simulation, hardware, runtime per sweep) is reported in Appendix G.

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: The work studies privacy in federated learning using public benchmark datasets, with no human-subject data, and conforms to the NeurIPS Code of Ethics.

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: A broader-impact paragraph discusses the positive impact (tighter per-client privacy in deployed FL, exposing latency-driven disparate vulnerability) and negative risk (overstated guarantees if the trust assumption on the privatiser committee is weakened).

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 release consists of analysis code and a simulation harness; no pretrained generative models, scraped datasets, or other high-misuse-risk assets are released.

    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: All datasets and pretrained backbones used are cited; their licences permit research use and are respected.

    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 released code includes a README documenting reproduction commands for each table and figure.

    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: No crowdsourcing or human-subject experiments are conducted.

    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: No human-subject research is conducted, so no IRB review applies.

    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: LLMs are not a component of the core methods; any LLM use was limited to writing assistance and does not impact the methodology, rigor, or originality.

    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.