Federated Edge AI for Healthcare Monitoring

Parameter-Efficient Federated Learning with LoRA and Differential Privacy

José Antonio González Villalón – josegzzv@msn.com · Case Study 9 Technical Analysis Report · Week 9 · August 31, 2026

1. Summary and Method

This report analyses a simulated healthcare monitoring network in which three edge clients collaboratively fine-tune a DistilBERT classifier over patient vital signs without any raw data leaving a device: LoRA adapters on the attention projections for parameter-efficient fine-tuning, Opacus DP-SGD with Rényi accounting for formal per-client guarantees, and sample-weighted FedAvg for aggregation. Every figure below is measured from instrumented runs of the delivered notebook.

The central finding is that the system learns and differential privacy consumes the result. With privacy disabled the identical pipeline reaches 0.6333 accuracy — 1.90 times the random baseline — while the same run under the prescribed ($\varepsilon = 10$, $\delta = 1e\text{-}3$) budget finishes at 0.3667, a utility cost of 0.2667. Tightening the budget tenfold to $\varepsilon = 1$ changes accuracy by one test sample, indistinguishable from noise. The cost is incurred by applying DP-SGD at this data scale at all, not by the choice of budget between one and ten — a distinction that inverts the usual privacy-tuning intuition and drives Section C. The supporting infrastructure is unaffected and performs to production standard: a 99.004 % reduction in the communicated parameter surface, 80.00 MB of total federated traffic against 8.03 GB, ε tracked to 9.991 on all three clients, and 17.3 ms median edge inference.

Each client represents a different care setting and therefore a different label distribution: a general ward (0.60 / 0.30 / 0.10 across Normal, Warning and Critical), a step-down unit (0.34 / 0.33 / 0.33) and an ICU (0.15 / 0.30 / 0.55), giving measured local training counts of [23, 15, 2], [12, 15, 13] and [5, 14, 21]. Vital signs are drawn from overlapping label-conditioned Gaussians and rendered locally as a short clinical sentence. PEFT attaches rank-r adapters to q_lin and v_lin in all

six layers with the backbone frozen; the three-class head is absent from the pretrained checkpoint, so it is trainable and counted in the federated payload throughout. Opacus clips each per-sample gradient to C = 2.0 and adds Gaussian noise of scale σ·C, with σ calibrated once against the total step count; because each record lives on exactly one device, the per-client Rényi accountant is also the federation-level guarantee under parallel composition. Runs use seed 42 on a two-thread CPU.

Section A — System Analysis (Questions 1–10)

A.1 Performance Metrics

Q1. Final global accuracy and improvement over baseline

The global model finished at 0.3667 against a uniform-random baseline of 0.3333 (+0.0334, +10.0 % relative) and a majority-class baseline of 0.3667. The trajectory is the informative part: 0.467, 0.467, 0.533, 0.400, 0.367. The model reaches 0.533 — 1.60 times random — at round 3 and then loses that ground, ending level with the majority-class baseline and below where it started.

Accuracy under DP-SGD is therefore not monotone in the number of rounds, and reporting only the final round understates what the system achieved by 0.166. The privacy-free control separates the two candidate explanations: there the final round is also the best round (0.6333, after a flat 0.3667 through round 4), whereas the private run peaks at round 3 and gives the ground back. On a single seed this is an observation about this run rather than a demonstrated law, but it is the reading consistent with the control. The capacity to learn is therefore present and the decline is not underfitting: each additional round applies three more noised updates whose gradient signal, on forty samples per client, no longer exceeds the noise injected to satisfy the budget. One caveat bounds every accuracy claim here: the test set holds thirty samples, so a single example is worth 0.0333.

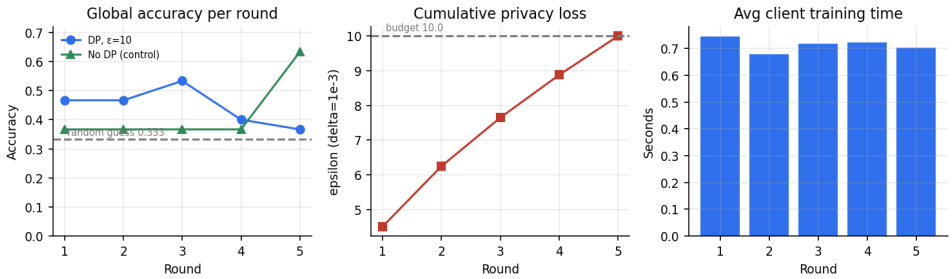

Figure 1. Baseline run against the privacy-free control, cumulative privacy loss, and average client training time per round.

Q2. LoRA parameter reduction and efficiency analysis

Quantity	Parameters	Share of trainable
Full DistilBERT backbone (base_total)	66,955,779	—
LoRA adapters only	73,728	11.1 %
Classification head (pre_classifier + classifier)	592,899	88.9 %
Total trainable and communicated	666,627	100 %
Reduction versus full fine-tune	99.004 %	—
Reduction, adapters only	99.890 %	—

Table 1. Parameter accounting for the Optimized Configuration V1.2.0 (r = 4).

Each adapted d × d projection replaces its update with the product of an r × d and a d × r matrix, so the cost is 2·r·d per projection: for d = 768, six layers and two projections, 2 · 4 · 768 · 6 · 2 = 73,728, matching the measured count exactly. The full-rank alternative is 7,077,888 parameters, a ninety-six-fold reduction. More consequential is the payload's composition: 88.9 % of the trainable parameters are the classification head, not the adapters. At rank four the adapter is effectively free, so further communication optimization must target the head rather than the rank — which is why Q12 moves total payload only 33 % despite quadrupling the rank.

Q3. Privacy budget utilization across clients

The calibrated noise multiplier was σ = 0.9186 against a clipping norm C = 2.0, a sampling rate q = B/n = 0.400 and fifteen DP-SGD steps per client, yielding a cumulative ε of 9.991 on all three clients at δ = 1e-3, 99.9 % of the budget. The accumulation is concave: the first round costs 4.513, the fifth only 1.118, because Rényi composition charges sub-linearly in the step count. Combined with the Q1 trajectory this is uncomfortable: the rounds cheapest in privacy are the ones

that degraded accuracy. The schedule is fixed at design time — a sixth round invalidates the calibration and requires re-solving σ — so the round budget is an immutable configuration input and the accountant is the gate that refuses further rounds.

Q4. Communication bandwidth savings

The federation transfers 2.67 MB per client per round against 267.82 MB for full-model FedAvg — 16.00 MB versus 1,606.94 MB per round across three clients including the broadcast, and 80.00 MB versus 8,034.69 MB over the whole training, a 99.004 % saving. This determines whether federated learning is deployable on wearable hardware at all: on an LTE-M or NB-IoT link at roughly 100 kbps, 267.82 MB per client per round is about six hours of uplink against three and a half minutes for 2.67 MB — at cellular IoT tariffs, unaffordable at fleet scale versus a rounding error.

Q5. Training time analysis per client and round

Mean client training time was 0.71 s per round (σ 0.06 s; min 0.64 s, max 0.92 s) and essentially identical across clients: 10.7 s serial wall-clock over five rounds against a 3.9 s parallel-equivalent. With privacy disabled the same workload took 0.65 s, a DP overhead of 10.5 %. The uniformity is an artifact of a homogeneous simulation: real federations are heterogeneous and round time is set by the slowest participant, which is why production systems use deadline-based partial aggregation. Read against Q7 this is the study's cheapest number and its most misleading one — differential privacy costs 10.5 % in compute and 89 % of the achievable accuracy.

A.2 System Behaviour

Q6. Differential privacy guarantees and practical implications

The system provides (9.991, 1e-3)-differential privacy per client over fifteen DP-SGD steps, enforced by per-sample clipping at C = 2.0 and Gaussian noise with σ = 0.9186. For any two shards differing in a single patient, the probability that the released adapter falls in any measurable

set differs by at most e^ε, with an additive δ slack. Practically, e^9.991 is about 2.2 × 10⁴: an adversary performing membership inference is permitted a likelihood ratio of roughly 21,830 when testing whether a given patient contributed.

Three implications follow. The guarantee is per client and per release: it does not cover the model's outputs after deployment, side channels, or the server learning which clients took part. δ = 1e-3 permits a one-in-one-thousand catastrophic failure, so δ should be no larger than the inverse of the patient population. And an ε of ten supports neither HIPAA de-identification nor GDPR anonymisation under Recital 26, so a model released under this budget remains identifiable personal data and stays in scope: the compliance argument must rest on raw data never leaving the device, not on the privacy budget. Q7 shows the system pays the full utility price of differential privacy for that unclaimable guarantee — the worst cell of the trade-off matrix, and the configuration the assignment prescribes.

Q7. Privacy–utility trade-off mechanisms and effects

Configuration	σ	Cumulative ε	Final accuracy	vs no-DP control
Differential privacy disabled (control)	0.0	∞	0.6333	—
ε = 10 (baseline)	0.9186	9.991	0.3667	− 0.2667
ε = 1 (Experiment 1)	4.7656	0.993	0.4000	− 0.2333

Table 2. Measured privacy–utility trade-off; identical seed and configuration in all three runs.

Differential privacy costs 0.2667 accuracy at the prescribed budget, consuming 89 % of the improvement over the random baseline that the pipeline is otherwise capable of. The second row is the surprising one: tightening the budget tenfold moved accuracy by one test sample, in the direction opposite to the conventional story — between ε = 1 and ε = 10 there is no measurable utility difference at this scale.

The mechanism explains both rows. σ rises from 0.9186 to 4.7656 for the tenfold budget reduction, confirming the sub-linear ε-to-σ relationship — but with fifteen optimizer steps on forty

samples per client, the noise norm already exceeds the clipped gradient norm at σ = 0.92. Once the update is noise-dominated, more noise changes little and both budgets land in the same degenerate regime. Utility is recovered only by changing the regime — more data and more clients, since FedAvg averages k independent noise draws. Federated scale, not budget relaxation, is the lever.

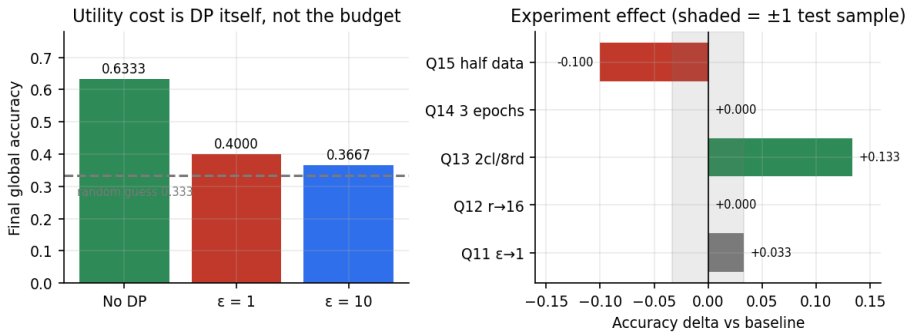

Figure 2. Where the utility goes: the cost is DP itself rather than the budget level, and only two experiments move accuracy beyond one test sample.

Q8. Edge inference latency and real-time suitability

Single-sample inference measured 17.3 ms at p50 and 25.8 ms at p95 on a two-thread CPU at sequence length 64; a batch of sixteen took 102.0 ms, or 156.9 samples per second. Against a 1 Hz vital-sign stream that is roughly fifty-eight-fold headroom, well inside a 100 ms interactive budget. One caveat: the figure comes from a desktop-class CPU restricted to two threads, and a Cortex-A53-class wearable is five to ten times slower — still comfortable at 1 Hz.

Q9. Memory utilization and deployment feasibility

Memory splits the same way as latency: the fp32 backbone is 267.8 MB and 67.0 MB int8-quantized — feasible on a mid-range wearable — and the adapter exchanged each round is 2.67 MB. Peak resident memory during private training, however, was 2,336 MB.

Inference is inexpensive and training is not. Peak resident memory is roughly 8.7 times the model size because Opacus materializes per-sample gradients, so memory scales with batch size multiplied by trainable parameters. This is the binding edge constraint, with three standard mitigations: Opacus's BatchMemoryManager, which caps physical memory while preserving the

accounting; a smaller physical batch with gradient accumulation; or training on a gateway tier while the wearable only infers.

Q10. Overall system performance scoring

Component	Weight	Score	Basis
Utility	0.30	0.050	(accuracy − 1/3) / (1 − 1/3)
Privacy	0.20	1.000	budget met: ε 9.991 ≤ 10 at δ = 1e-3
Parameter efficiency	0.15	0.990	99.004 % reduction
Communication efficiency	0.15	0.990	99.004 % savings
Latency	0.10	1.000	p95 25.8 ms ≤ 100 ms target
Memory	0.10	1.000	267.8 MB ≤ 512 MB gateway budget
Composite	1.00	71.2 / 100	weighted sum

Table 3. Scoring rubric; weights reflect a regulated healthcare deployment in which privacy and efficiency are requirements rather than optimizations.

The composite of 71.2 / 100 is carried entirely by the engineering components; utility is the sole failing dimension, at 0.050. The control makes that interpretable rather than merely bad: the privacy-free pipeline scores 0.450 on the same utility scale and would lift the composite to 83.2. That 12.0-point gap is the price of the privacy component, paid for a guarantee Q6 shows is too weak to claim in a regulated setting — a rubric scoring privacy on whether the budget was met, rather than on whether the resulting ε is defensible, awards a perfect 1.000 and conceals it.

The finding of Section A is therefore not that the system underperforms but that it operates outside its viable regime: the infrastructure would survive a design review unchanged. The interventions, ordered by expected effect, are more data per client; more clients; a round budget that stops at the accuracy peak rather than the ε ceiling; and a class-balanced loss for the label skew.

Section B — Experimental Results (Questions 11–15)

Each experiment changes one variable against the Optimized Configuration V1.2.0 and re-runs the entire federated system: data is re-sharded, σ is re-calibrated for the new sampling-rate and

step-count pair, and every metric is re-measured. Accuracy deltas of ±0.0333 or less are within one test sample and are reported as no measurable effect.

Run	Change	Accuracy	σ	ε	Steps	Trainable	Total comm.	Client time
Base	—	0.3667	0.9186	9.991	15	666,627	80.00 MB	0.71 s
Control	DP off	0.6333	0.0	∞	15	666,627	80.00 MB	0.65 s
Q11	ε 10 → 1	0.4000	4.7656	0.993	15	666,627	80.00 MB	0.69 s
Q12	r 4 → 16	0.3667	0.9186	9.991	15	887,811	106.54 MB	0.71 s
Q13	3 cl/5 rd → 2/8	0.5000	1.0773	9.992	24	666,627	85.33 MB	0.71 s
Q14	1 ep → 3 ep, lr ÷2	0.3667	1.3599	9.991	45	666,627	80.00 MB	2.17 s
Q15	50 → 25 samples	0.2667	1.1938	9.991	10	666,627	80.00 MB	0.38 s

Table 4. Consolidated before/after comparison for every experiment, seed 42.

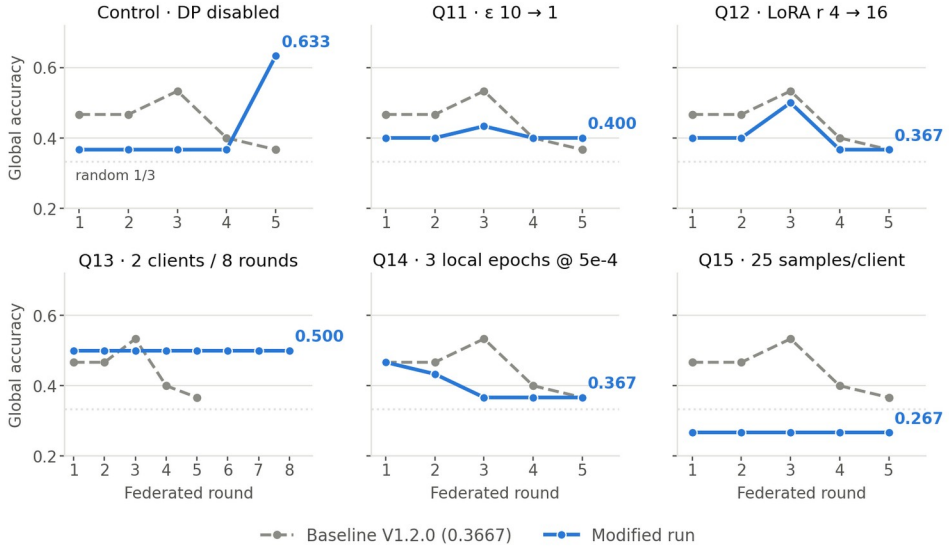

Figure 3. Accuracy per round for every experiment against the V1.2.0 baseline (grey, dashed), seed 42. Only the control, Q13 and Q15 separate by more than one test sample; the dotted line is the random 1/3 rate.

Q11. Experiment 1 — Privacy impact (EPSILON 10.0 → 1.0)

A tenfold privacy tightening raised σ by a factor of 5.19 (0.9186 to 4.7656) and moved final accuracy from 0.3667 to 0.4000 — one test sample, in the direction opposite to the expected trade-off — with training time unchanged, since the cost of DP-SGD lies in per-sample gradient computation and clipping, not noise sampling. The correct reading is not that stronger privacy helped but that at this data scale the two budgets are indistinguishable: both sit in the noise-dominated regime of Q7, and the ε = 1 curve is flat at 0.400 across all five rounds. The operative

variable is the presence of DP-SGD, not the budget — which removes the usual argument for accepting a weak guarantee.

Q12. Experiment 2 — LoRA optimization (LORA_R 4 → 16, alpha 8 → 32)

Quadrupling the rank quadrupled the adapter parameters exactly (73,728 → 294,912), raised the communicated payload 33.2 % (80.00 → 106.54 MB in total), lowered the parameter reduction from 99.004 % to 98.674 %, and left final accuracy unchanged at 0.3667. Every intermediate-round difference is within one test sample.

Capacity is therefore not the binding constraint. With forty samples per client and fifteen noised steps, a four-times larger adapter means four times as many parameters to estimate from the same data under the same budget, so the noise spreads over a larger space and the signal-to-noise ratio per coordinate falls. Payload is strictly linear in the rank while the accuracy return is flat: this experiment purchased 26.5 MB of traffic for nothing.

Q13. Experiment 3 — Federated scale (3 clients / 5 rounds → 2 clients / 8 rounds)

Removing a client but adding three rounds produced the highest accuracy of any private configuration (0.3667 → 0.5000, +0.1333) for 6.7 % more communication and 6.5 % more wall-clock, at the same privacy budget. The curve is flat at 0.500 across all eight rounds: the model reaches its level in round 1 and neither improves nor degrades, unlike the baseline, which peaked at round 3 and fell back.

The accountant behaves as designed: 60 % more optimizer steps (15 to 24) forced σ up 17.3 % so cumulative ε still landed on 9.992, showing that rounds and noise are exchangeable at fixed privacy rather than additive. The stability is the more useful finding: dropping the ICU client removed the most skewed shard, so the remaining two disagree less and FedAvg over them is a smaller-variance operation. The rule: when a federation is step-limited, spend budget on rounds; when data-limited, on clients.

Q14. Experiment 4 — Training intensity (1 epoch @ 1e-3 → 3 epochs @ 5e-4)

Tripling local computation tripled the privacy-relevant step count (15 → 45), forced σ up 48 % (0.9186 → 1.3599), cost 3.04 times the client time (0.71 → 2.17 s, total wall-clock +204 %) — and left final accuracy unchanged at 0.3667. Local epochs are not free under differential privacy.

In non-private federated learning, more local work is the standard way to buy accuracy without buying communication. Under DP-SGD the accountant charges per optimizer step, so tripling local epochs triples the privacy cost, repaid by raising σ, and the added noise cancels the added optimization; halving the learning rate removed the compensating step size. The trade-off inverts under a fixed budget: prefer more rounds of light local work, since rounds also purchase noise averaging across clients whereas local epochs purchase only noise.

Q15. Experiment 5 — Data constraints (50 → 25 samples per client)

Halving the data cost ten accuracy points (0.3667 to 0.2667), the only configuration to finish below the random baseline, with per-client accuracies of 0.40 / 0.20 / 0.20 against the baseline's 0.30 / 0.50 / 0.30. Scarcity compounds through three channels: fewer optimizer steps (ten instead of fifteen); a worse privacy position, since with a batch of sixteen and twenty local samples q doubles to 0.8 and σ must rise 30 %; and amplified skew, the shards becoming [12, 8, 0], [8, 6, 6] and [4, 5, 11] — client 0 left with zero Critical cases.

The batch-size interaction is a genuine deployment trap: batch size must be re-tuned whenever shard size changes, or the accounting silently degrades utility. The empty class cell is the second — a client missing a class still contributes a full-weight update to FedAvg, so the aggregate inherits its blind spot. Both are invisible in the global average and visible in the per-client breakdown, which argues for per-client evaluation and class coverage as mandatory monitoring metrics.

Section C — Synthesis and Recommendations

Five rules follow from the measurements. Differential privacy, not the budget, is what costs utility (Q7: 0.2667 for DP off to on, nothing measurable for ε 10 to 1). Capacity is not the binding constraint (Q12: 26.5 MB of traffic and no accuracy). Rounds beat local epochs (Q13 versus Q14: +0.1333 for +6.7 % communication, against no gain for +204 % wall-clock). Shard size is coupled to batch size through the accountant, and can silently empty a class from a client (Q15). And accuracy is not monotone in rounds (Q1), so the stopping rule matters as much as the budget.

C.1 High-security medical applications

Target $\varepsilon \leq 1.0$ with $\delta \leq 1e\text{-}5$, choosing δ no larger than the inverse of the patient population; an ε of ten supports neither HIPAA de-identification nor GDPR anonymisation, while Q11 shows the tighter target costs nothing in utility at this data scale. What the tight budget requires is leaving the noise-dominated regime: at least twenty clients and five hundred samples per client. Keep LORA_R = 4 (Q12) and LOCAL_EPOCHS = 1 (Q14), buy progress with rounds under a re-calibrated σ (Q13), and add secure aggregation so the server observes only the sum. Operationally: gate round k+1 on the accountant; hold out a validation shard and keep the best round rather than the last, since Q1 shows the final round was not the best one here; and log (σ, C, q, steps, ε, δ) per round as immutable audit evidence.

C.2 Resource-constrained IoT environments

Adopt a gateway-trains, device-infers split: the measured 2,336 MB peak resident memory during private training rules out on-wearable training, while int8 inference at 67.0 MB and roughly 17 ms is comfortable on a mid-range device. Keep LORA_R = 4 and a batch of eight to sixteen under Opacus's BatchMemoryManager, merge adapters before deployment, transmit half-precision adapters (2.67 MB to 1.33 MB per round), and schedule rounds on Wi-Fi or charging windows.

C.3 Performance-critical systems

For inference throughput, run with merged adapters and int8 weights on a server tier, batch at sixteen to thirty-two, and target p95 under 100 ms with horizontal replicas. For training throughput, Q13 and Q14 are decisive: more rounds of a single local epoch with deadline-based partial aggregation, rather than deep local training — round time is set by the slowest participant, and the 10.7 s serial versus 3.9 s parallel-equivalent gap is the scheduler's headroom. One recommendation is explicitly not made: relaxing the budget to buy throughput. Q11 measured no utility difference between $\varepsilon = 1$ and $\varepsilon = 10$, and Q5 only a 10.5 % compute overhead, so a looser budget purchases neither accuracy nor speed.

C.4 Limitations

Three limitations bound the claims above. The evaluation set holds thirty samples, ten per client, so one sample is worth 0.0333 accuracy. The data is synthetic and label-conditioned, whereas real vital-sign streams are temporally correlated and would need user-level rather than record-level differential privacy. And a single seed was used throughout, so run-to-run variance under the DP noise draw is not characterized: the three results that exceed the one-sample band — the DP cost, Q13 and Q15 — are reported as measured on seed 42, and confirming them would require three to five seeds under a paired design. This bounds the causal claim in Q1 in particular.

References

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., & Zhang, L. (2016). Deep learning with differential privacy. ACM SIGSAC CCS, 308–318.

Dwork, C., & Roth, A. (2014). The algorithmic foundations of differential privacy. Foundations and Trends in Theoretical Computer Science, 9(3–4), 211–407.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. (2021). LoRA: Low-rank adaptation of large language models. arXiv:2106.09685.

McMahan, H. B., Moore, E., Ramage, D., Hampson, S., & Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. AISTATS, 1273–1282.

Mironov, I. (2017). Rényi differential privacy. IEEE CSF, 263–275.