Toward Privacy-Efficient and Interaction-Aware Hyperparameter Tuning for Differentially Private Federated Learning

Devaganthan Sivakumar Srirangan
MBZUAI, Abu Dhabi
devaganthan.ss@mbzuai.ac.ae

Abstract

Hyperparameter search in federated learning (FL) must balance two conflicting forces: utility, and the privacy budget imposed by differential privacy (DP). We revisit the Poisson-repetition framework of Papernot and Steinke [5]—originally developed for centralized-DP setting—and adapt it to a Central-DP-FL pipeline. Our implementation on MNIST, using a 50-client convolutional benchmark, confirms that Poisson repetition preserves a favourable privacy–utility trade-off once appropriate clipping and noise parameters are chosen. Complete hyperparameter sweep reveals, non-monotonic interactions among noise multiplier, clipping norm, client-participation rate, and learning rate, exposing limitations of "one-knob-at-a-time" tuning and motivating theory that captures these couplings. To address multi-parameter search, we propose factored Poisson sampling: independent Poisson draws for each hyperparameter that maintain full coverage while potentially reducing the additive privacy cost. We outline open questions, both empirical and theoretical, needed to formalise this idea and to derive interaction-sensitive convergence bounds. The code for the base algorithm is available.

1 Introduction

Federated Learning (FL) has emerged as a powerful paradigm for collaboratively training machine learning models across a large number of decentralized devices or servers, each holding local data samples, without exchanging the data itself [3]. By enabling model training without raw data sharing, FL addresses growing concerns around data privacy and regulatory compliance. This decentralized approach has found applications in domains such as healthcare [6], mobile devices [2], and finance [8], where sharing sensitive data directly would pose significant legal and ethical challenges.

However, despite this inherent privacy advantage, model updates themselves can still leak sensitive information. This vulnerability has motivated the integration of formal privacy guarantees into FL systems, most notably through Differential Privacy (DP). By introducing carefully calibrated noise into model updates and employing tight privacy accounting mechanisms (e.g., the Moments or Rényi accountant [1], [4]), DP ensures that any single device's contribution remains indistinguishable, even to adversaries monitoring the training process.

While addressing privacy concerns is crucial, another significant challenge in federated learning is **hyperparameter tuning**. Achieving optimal model performance and privacy guarantees requires careful selection of several critical hyperparameters, which can be broadly categorized into three groups:

1. **Optimization hyperparameters**, such as server learning rate, client learning rate, and number of local epochs;

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

2. **Participation hyperparameters**, including client selection fraction and total number of participating clients;
3. **Privacy hyperparameters**, such as clipping norm and noise multiplier in differentially private settings.

Tuning these hyperparameters is notoriously difficult: the performance landscape is highly non-convex, high-dimensional, and heavily dependent on the underlying task. Further compounding this difficulty, recent studies have shown that publishing tuned hyperparameters can itself leak private information about the underlying data [5]. Consequently, hyperparameter tuning must also be performed under explicit privacy guarantees. While privacy-preserving hyperparameter tuning has received attention in centralized private learning, its exploration in federated settings remains limited.

Addressing hyperparameter tuning under differential privacy constraints in federated learning introduces several unique challenges:

First, each hyperparameter evaluation — even preliminary ones — involves executing federated training rounds and injecting noise to ensure privacy. This results in non-trivial privacy costs: each trial consumes a portion of the finite overall privacy budget, directly impacting the budget available for final model training.

Second, the search space in FL is particularly vast and intricate. Optimization, participation, and privacy hyperparameters interact in complex, often unpredictable ways, creating a rugged and sensitive performance landscape. Traditional tuning strategies such as grid search or random search become inefficient, especially under constrained communication and privacy budgets.

Together, these challenges underscore the need for specialized approaches to hyperparameter tuning in differentially private federated learning — an essential step toward realizing the full promise of secure, efficient, and scalable FL systems.

In this work, we take a step toward addressing these challenges by adapting and extending the framework proposed in *Hyperparameter Tuning with Rényi Differential Privacy* [5] to the federated setting. Our contributions are threefold:

- **Adaptation and validation:** We demonstrate how this method can be adapted to FL systems with central privacy guarantee.
- **Empirical analysis of hyperparameter interactions:** Through empirical experiments, we show that hyperparameters in DP-FL settings interact in highly complex and nontrivial ways—small changes in one hyperparameter can substantially impact the optimal settings of others. This finding motivates the need for a deeper optimization-theoretic understanding of the DP-FL hyperparameter landscape.
- **Exploring scalable strategies for multi-parameter tuning:** While the original Poisson-repetition method from [5] remains effective, its direct application to high-dimensional hyperparameter spaces may be suboptimal in terms of privacy efficiency. We explore a factored Poisson sampling approach—where each hyperparameter is sampled independently—as a potential refinement. This direction raises interesting theoretical and practical questions about whether such factorization can lead to tighter privacy guarantees or improved utility in multi-parameter DP-FL tuning.

To lay the groundwork for our approach, we begin by reviewing key concepts in differential privacy and its integration within federated learning systems.

2 Background

2.1 Differential Privacy and Its Role in Federated Learning

Differential Privacy (DP) provides a formal framework to quantify the privacy guarantees of randomized algorithms operating on sensitive data. An algorithm $\mathcal{M} : \mathcal{X}^n \to \mathcal{Y}$ is said to satisfy (ε, δ)-differential privacy if, for any two neighboring datasets x, x' differing in a single individual's data and any subset of outputs $S \subseteq \mathcal{Y}$,

$$\Pr[\mathcal{M}(x) \in S] \leq e^{\varepsilon} \Pr[\mathcal{M}(x') \in S] + \delta. \tag{1}$$

Here, ε measures the worst-case privacy loss, while δ bounds the probability of a larger privacy breach. Smaller values of ε and δ indicate stronger privacy guarantees.

To enable tighter privacy accounting over multiple adaptive steps, Rényi Differential Privacy (RDP) has been introduced as a relaxation of DP. A randomized mechanism $\mathcal{M}$ satisfies (λ, ε)-RDP for order $\lambda > 1$ if, for any adjacent inputs x, x',

$$D_\lambda(\mathcal{M}(x) \| \mathcal{M}(x')) \leq \varepsilon, \tag{2}$$

where $D_\lambda(P\|Q)$ denotes the Rényi divergence of order λ between distributions P and Q. RDP naturally composes over multiple mechanisms and can be converted to approximate (ε, δ)-DP guarantees, making it particularly useful for analyzing iterative algorithms such as differentially private stochastic gradient descent (DP-SGD).

In the federated learning (FL) setting, DP is typically integrated by modifying the training process: each client's local update is first clipped to control sensitivity and then noise is added either before or after aggregation. The cumulative privacy loss over communication rounds is tracked using privacy accountants, most notably the Moments Accountant or the RDP-based accountant.

There are two primary models for applying differential privacy in federated systems:

- **Central Differential Privacy (Central DP):** Noise is added at the server after aggregating the local updates. The trust assumption is that the server honestly aggregates the unperturbed updates and only releases privatized information.
- **Local Differential Privacy (Local DP):** Each client adds noise to their updates before sending them to the server. This eliminates the need for trusting the server but often results in a significant utility loss compared to Central DP.

In this work, we focus on the Central DP model for federated learning, which achieves a better privacy-utility tradeoff while still protecting individual client contributions through formal privacy guarantees.

With the differential-privacy framework in place, we next turn to the specific challenge of tuning hyperparameters under these constraints.

2.2 Hyperparameter Tuning in DP-FL (A Brief Recap)

Tuning optimization, participation, and privacy hyperparameters in federated learning is uniquely constrained by (i) privacy budget consumption per trial, (ii) non-linear interactions among hyperparameters under differential privacy, and (iii) the significant computational and system resources required for each federated training run. These coupled challenges motivate for specialized algorithms that are both privacy-accounted and resource-efficient. We now survey the principal lines of work addressing these needs.

3 Related Work

We first discuss a privacy efficient hyperparameter search by Papernot and Steinke [5], which rigorously analyzes privacy accounting during hyperparameter search. We then review the *Cooperative SGD* framework by Wang and Joshi [7], which provides theoretical insights for finding the optimal hyperparameters

3.1 Hyperparameter Tuning with Rényi Differential Privacy

While each training run in differential privacy may individually protect sensitive data, the process of hyperparameter selection itself can leak information. Papernot and Steinke [5] demonstrate that the optimal hyperparameter choice, such as a regularization weight, can shift noticeably with small changes in the training set, enabling membership inference attacks.

This phenomenon is illustrated in Figure 1: introducing a few outliers into the dataset significantly alters the accuracy curve as a function of the regularization parameter α, making hyperparameter choices indirectly reveal information about data presence. These observations highlight the need for hyperparameter tuning strategies that preserve differential privacy end-to-end.

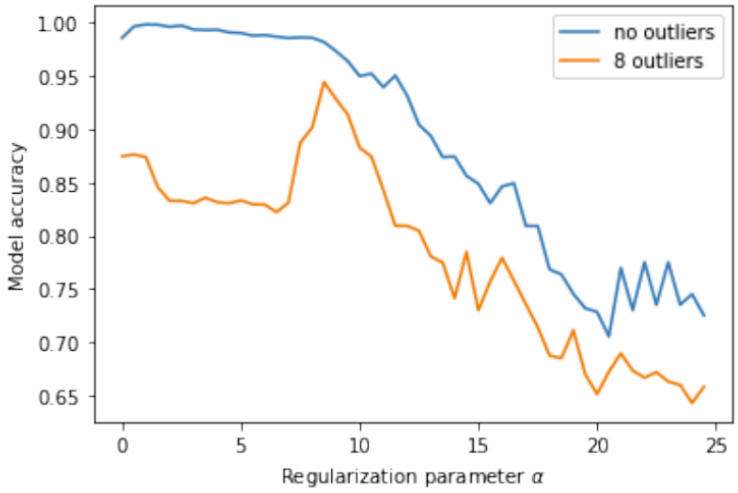

Figure 1: Effect of data outliers on hyperparameter tuning. The optimal regularization parameter α shifts in the presence of outliers, revealing sensitivity of tuning to dataset membership (adapted from Papernot and Steinke [5]

They formalize private hyperparameter tuning as selecting the best outcome from a set of differentially private base algorithms, each corresponding to a candidate hyperparameter setting. A naïve approach—running all candidates a fixed number of times—accumulates privacy loss linearly, leading to poor overall guarantees. To mitigate this, they propose randomly sampling the number of repetitions according to a carefully chosen distribution, adding uncertainty that improves privacy bounds.

Their key insight is that if each base algorithm is individually Rényi differentially private, then repeating a random number of runs drawn from a light-tailed distribution (e.g., Poisson) results in significantly tighter composition. Specifically, under a Poisson distribution with mean μ for the number of runs, the overall procedure satisfies a new (λ, ε')-RDP guarantee, where ε' grows slowly with μ rather than linearly with the number of candidates.

We now briefly state this main result.

Poisson–RDP Guarantee Let $Q : \mathcal{X}^n \to \mathcal{Y}$ be a randomized mechanism satisfying both (λ, ε)-Rényi Differential Privacy (RDP) and $(\hat{\varepsilon}, \hat{\delta})$-Differential Privacy (DP) for some $\lambda > 1$, $\varepsilon, \hat{\varepsilon}, \hat{\delta} \geq 0$. Suppose the number of repetitions K is drawn from a Poisson distribution with mean $\mu > 0$. Then, the mechanism that runs Q independently K times and outputs the best result satisfies (λ, ε')-RDP, where

$$\varepsilon' = \varepsilon + \mu\hat{\delta} + \frac{\log \mu}{\lambda - 1}.$$

This result shows that under Poisson sampling, the privacy cost grows only *logarithmically* with the mean number of repetitions μ, rather than linearly with the number of hyperparameter candidates, thus achieving much tighter privacy guarantees during hyperparameter search.

Figure 2 shows that using a Poisson distribution for randomizing the number of runs achieves significantly tighter privacy bounds compared to naïve repetition. This confirms the effectiveness of Poisson-based random repetition for enabling privacy-preserving hyperparameter tuning under tight budgets.

3.2 Choosing Hyperparameters via a Non-Private Convergence Bound

To ground our DP-FL tuning strategy, we first examine a *non-private* yet analytically way of optimising the hyperparameters [7]. Their framework unifies periodic averaging, elastic-averaging,

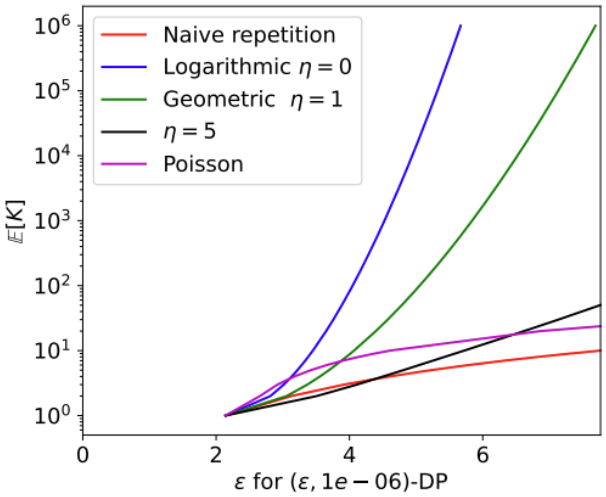

Figure 2: Comparison of privacy guarantees (converted to $(\varepsilon, 10^{-6})$-DP) for different repetition strategies, plotted as a function of expected number of repetitions $E[K]$. The Poisson distribution achieves substantially lower privacy loss compared to naïve fixed-count repetition, especially when the number of runs grows (adapted from [5]).

and decentralized SGD, and delivers a single convergence upper bound that isolates how each hyperparameter affects the final optimization error.

Key bound and an explicit rule for α: For elastic averaging with one auxiliary variable, the error bound in Corollary 1 simplifies to

$$\underbrace{\mathcal{O}\left(\frac{1}{\sqrt{mK}}\right)}_{\text{fully-sync term}} + \underbrace{\eta_{\text{eff}}^2 L^2 \sigma^2 \left(1+\zeta^2\right)\left(\frac{m}{m+2}\tau - 1\right)}_{\text{network error}}, \tag{3}$$

where $\zeta = \max\{|1-\alpha|, |1-(m+1)\alpha|\}$ and m is the number of clients. Minimizing the *network-error* term yields the closed-form elasticity

$$\alpha^* = \frac{2}{m+2}. \tag{4}$$

(Lemma 1), which indeed produces the lowest loss in practice (see Fig. 4 of the original paper). This illustrates how a tight analytical bound can translate directly into a concrete, *best-in-class* hyperparameter choice.

Why a DP-aware bound could be useful Once differential privacy is enforced, the picture changes: noise multiplier, clipping norm, client participation rate, and α begin to interact nonlinearly. Without a *Central-DP-FL* convergence analysis, we cannot predict how the optimal α^* (or learning rate, local epoch count, etc.) should shift under privacy noise—a phenomenon we empirically observe in Section 5, where a small tweak in one knob forces a drift in another.

Thus, the resulting bound could captures the full coupling between optimization, participation, and privacy hyperparameters, enable achieving a resource-efficient, privacy-accounted tuning algorithm.

In the next section, we take a first step toward a privacy-accounted tuning algorithm for the FL setting

4 Proposed Approach: Private Hyperparameter Tuning in FL

Papernot and Steinke [5] show that if each *base algorithm* Q is (λ, ε)-RDP, then repeating Q a Poisson(μ) number of times and returning the best result inflates the privacy cost only by

$$\varepsilon' = \varepsilon + \mu\delta + \frac{\log \mu}{\lambda - 1}, \tag{5}$$

a logarithmic—rather than linear—penalty.

In our Central-DP-FL setting, we keep the same Poisson-repetition skeleton but *redefine* the base mechanism to one complete FL round in which (i) each client clips its local update, (ii) the server aggregates un-noised updates, and (iii) a single Gaussian noise vector (with multiplier σ) is added *centrally* before the model is broadcast.

Since each training round already satisfies $(\lambda, \varepsilon_{\text{round}})$-RDP for any $\lambda > 1$, we can directly apply the Poisson accountant to get the overall privacy guarantee.

$$(\lambda, \varepsilon_{\text{round}} + \mu\delta + \frac{\log \mu}{\lambda - 1})\text{-RDP}$$

for the entire hyperparameter search.

The only new design knob is the Poisson mean μ, which we will tune to balance privacy budget, search coverage, and wall-clock cost; Algorithm 1 formalizes the procedure.

Algorithm 1 Poisson Repetition (High-Level)

Input:
  Poisson mean μ
  Central DP-FL Algorithm $\mathcal{M}$
  Privacy Parameters
    Noise Multiplier s
    Clipping Norm C
    Sample Rate q
  Optimisation Parameters
    Learning Rate α
    communication Rounds T
Initialize:
  best_score $\leftarrow -\infty$
  Candidate Hyperparameters $\theta_1, \theta_2, \ldots, \theta_m$
for each repetition $i = 1$ to Poisson(μ) **do**
  Sample hyperparameter θ_i
  Run $\mathcal{M}(\theta_i)$ (see Algorithm 2)
  Measure metric u_i
  if $u_i >$ best_score **then**
    Update best score and hyperparameters
  end if
end for
Output:
  Best hyperparameter setting and Cumulative Privacy Cost (From Equation (5))

Algorithm 2 Federated Learning Iteration (Expanded)

Input: Initial model w^0, hyperparameters θ
for round $t = 0$ to $T - 1$ **do**
  Sample clients q fraction of total clients
  Broadcast Global Model
  for each sampled client **in parallel do**
    Local SGD
    $\Delta \leftarrow$ Local Model - Global Model
    clip(Δ, C)
    Send update
  end for
  Server aggregates updates
  Add_Noise(s, C) to Aggregate
  Global Model $\leftarrow$ Noisy Aggregate + Global Model
  Update RDP accountant(s, q)
end for
Output: Final model w^T with metric

Our experimental setup closely follows the setup of Papernot and Steinke [5], with the main difference stemming from the central DP setting. As a result, the privacy parameters—such as the noise multiplier, clipping norm, and sampling strategy are adjusted to reflect the aggregation and accounting methods specific to this model of privacy.

Experimental details: We train an all-convolutional network (five 3×3 layers with 32–32–64–64–64 filters) on MNIST for 600 communication rounds (one local epoch per round) across 50 simulated clients, with 5 clients (10%) sampled each round and a minibatch size of 256. Privacy is enforced with a noise multiplier $\sigma = 0.5$, clipping norm $C = 0.005$, and target $\delta = 10^{-5}$; the Rényi accountant converts the cumulative RDP bound to (ε, δ) after training. The server learning rate is the sole search variable, varied logarithmically from 0.0005 to 0.02; all other optimisation, participation, and privacy parameters remain fixed throughout the sweep. In Figure 3, we plot the maximal accuracy achieved during the hyperparameter search for Poisson distributions considered previously as a function of the total privacy budget expended by the search. The experiment is repeated 5 times and the mean result reported. This experiment validates the Privacy-Utility tradeoff for this adaptation.

Having validated Poisson repetition in our Central-DP-FL pipeline, we now investigate how the broader hyperparameter landscape behaves in practice.

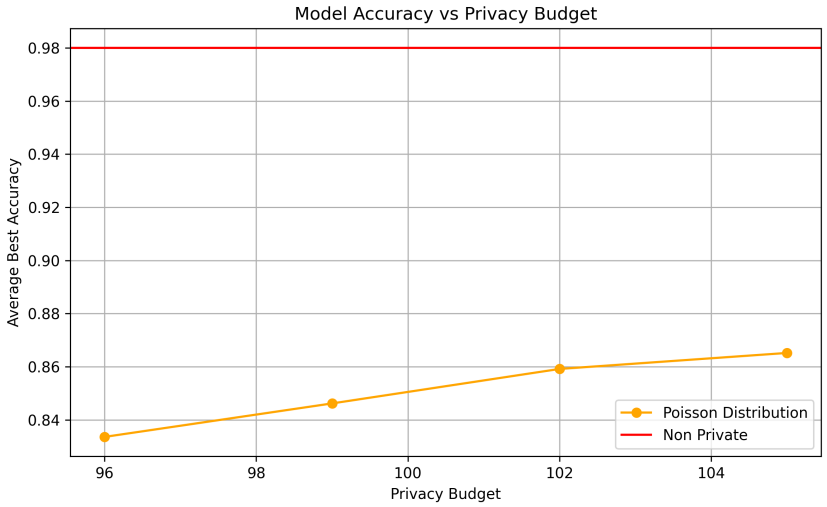

Figure 3: Accuracy of the CNN model in Central-DP-FL setting obtained at the end of the hyperparameter search, for the poisson distributions on the number of repetitions K we considered. We report the mean over 5 trials of the experiment.

5 Exploring the Hyperparameter Space

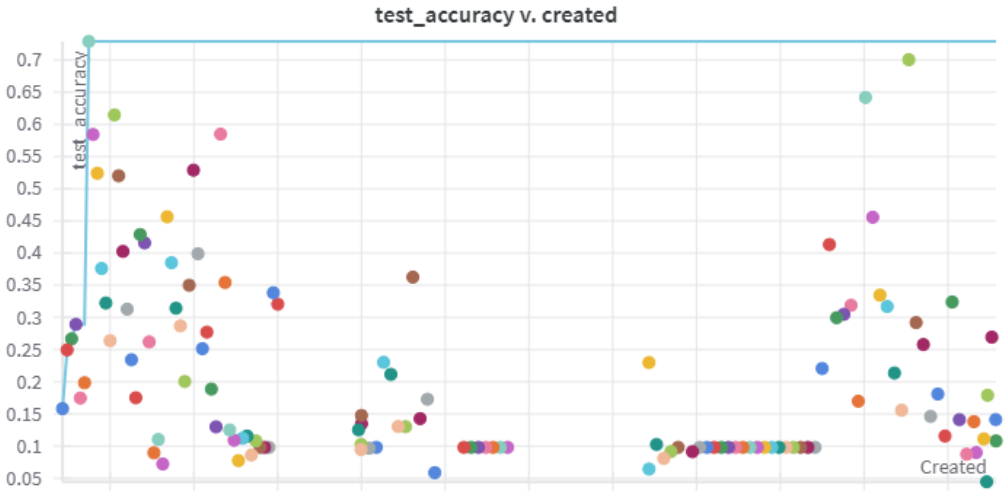

Figure 4: Test accuracy for different $(\sigma, , C, , \text{lr})$ configurations. Most settings yield divergence, forming a dense band near 0.1 or sub-50% accuracy.

Before launching the Poisson-repetition search, we first had to establish a stable baseline configuration for all hyper-parameter including the learning rate. This proved surprisingly difficult: as illustrated in Figure 4, most $(\sigma,\ C,\ learningRate)$ combinations drove the model below 50% test accuracy, and many diverged altogether—evidence of the brittleness and tight coupling among DP-FL hyperparameters. After extensive trial-and-error we settled on $\sigma = 0.5$, $C = 0.005$, $q = 0.10$; only with this anchor in place we could fine tune the learning rate.

Yet even this "fixed-everything-but-one" strategy exposed a deeper issue. Figure 5 plots the *optimal* learning rate obtained at each noise multiplier in the range $\sigma \in [0.5, 1.0]$. The curve has a zig-zag

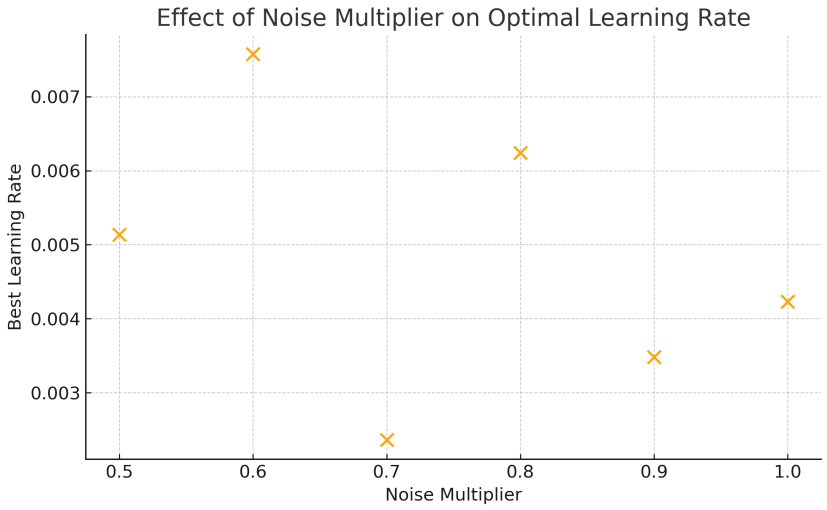

Figure 5: The optimal learning rate zig-zags across $\sigma \in [0.5, 1.0]$, showing non-monotonic behavior

characterstic: the best rate jumps from $\approx 5 \times 10^{-3}$ at $\sigma = 0.5$ to 7×10^{-3} at 0.6, plummets below 3×10^{-3} at 0.7, rises again past 6×10^{-3} at 0.8, and so on. This three-fold swing, with alternating peaks and troughs, could be an evidence that once DP noise is introduced the learning-rate landscape becomes highly non-monotonic and noise-dependent.

This empirical observation motivates why a DP-aware convergence bound could provide useful insights. Without a Central-DP-FL analogue of the Cooperative-SGD analysis, we cannot predict how an "optimal" setting (learning rate, local epochs, elasticity α, etc.) should adapt as privacy noise or participation parameters change. Bridging this gap—by deriving interaction-sensitive bounds that explain and guide these shifts—remains an open and important direction for future work.

The observed interaction suggest that tuning a single parameter in isolation is inadequate; this motivates an extension of Poisson repetition to multi-dimensional search, which we introduce next

6 Factored Poisson Sampling for Multi-Hyper-parameter Search

In this section, we propose a simple yet potentially more privacy-efficient extension to the Poisson-repetition framework for hyperparameter tuning when multiple parameters must be optimized simultaneously. By factoring the sampling process across dimensions, we aim to reduce the cumulative privacy cost while maintaining effective coverage of the hyperparameter space.

Setting Assume we must tune two hyper-parameters—say learning-rate (θ_1) and noise-multiplier (θ_2)—each with 20 discrete candidates, yielding $20 \times 20 = 400$ possible pairs.

Papernot–Steinke baseline In the original framework every *pair* (θ_1, θ_2) is treated as a separate base algorithm, so the Poisson repetition must have mean

$$\mu \approx 400$$

to visit each pair with reasonable probability.

Proposed factored scheme

1. Draw two *independent* counts

$$K_1 \sim \text{Poisson}(20), \qquad K_2 \sim \text{Poisson}(20).$$

2. Run $K_1 \times K_2$ training trials.
   - At each trial independently sample one value of θ_1 from its 20-element grid.
   - Independently sample one value of θ_2 from its own grid.

The expected number of trials remains $\mathbb{E}[K_1]\mathbb{E}[K_2] = 400$, preserving coverage.

Open questions

- Is the proposed factored scheme *exactly* equivalent to a single 400-mean Poisson repetition, or does the factorisation yield a provably tighter bound?
- How does the scheme extend to d hyper-parameters, each with m candidates (draw d independent Poisson(m) counts and perform $\prod_i K_i$ trials)?
- What is the empirical privacy–utility trade-off when the bound is converted to (ε, δ)-DP in realistic FL workloads?

Answering these questions will determine whether **factored Poisson sampling** merely matches, or can actually improve upon, the Papernot & Steinke method for high-dimensional hyper-parameter tuning.

7 Conclusion

We set out to adapt privacy-aware hyper-parameter search to the federated-learning setting, where differential-privacy guarantees must coexist with system constraints and a vastly larger configuration space. Our first contribution was **adapting the Poisson-repetition framework of Papernot & Steinke to Central-DP-FL**; experiments on MNIST confirmed that, with appropriate clipping and noise, the method retains a favourable privacy–utility trade-off. Second, an **empirical exploration of the search space** revealed pronounced, non-monotonic interactions among noise multiplier, clipping norm, client-participation rate, and learning rate—underscoring the need for theory that captures these couplings. Finally, we **sketched a factored Poisson sampling scheme** that may scale the framework to many hyper-parameters with a smaller additive privacy cost, and we outlined open questions needed to formalise and validate this idea.

8 Limitations and Future Works

Our findings highlight potential fragility of current DP-FL tuning methods. They point toward two immediate research directions:

(i) developing convergence bounds that are *interaction-sensitive*, in the spirit of Cooperative SGD but adapted to privacy-aware settings, and

(ii) rigorously analysing factored sampling (and other structured search strategies) to determine when they yield provably tighter or more practical privacy guarantees.

That said, this work is not without limitations. While our experiments empirically suggest coupling between hyperparameters, the evidence is not strong enough. The number of trials per configuration is limited (typically 5), and the observed fluctuations in optimal settings could in part reflect stochastic variance rather than true dependencies. Additionally, although we motivate the need for interaction-sensitive convergence bounds, it remains unclear whether such theoretical constructs will accurately capture the interactions between the hyper parameters.

9 Acknowledgements

I would like to thank Dr. Praneeth Vepakomma and Dr. Samuel Horvath for their guidance through the course of this project. I am also grateful to Hamad Jaseem and Joseph Geo for helping me get started with federated learning simulations.

References

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS'16. ACM, October 2016. doi: 10.1145/2976749.2978318. URL http://dx.doi.org/10.1145/2976749.2978318.

[2] Andrew Hard, Kanishka Rao, Rajiv Mathews, Swaroop Ramaswamy, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. Federated learning for mobile keyboard prediction, 2019. URL https://arxiv.org/abs/1811.03604.

[3] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data, 2023. URL https://arxiv.org/abs/1602.05629.

[4] Ilya Mironov. Rényi differential privacy. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, page 263–275. IEEE, August 2017. doi: 10.1109/csf.2017.11. URL http://dx.doi.org/10.1109/CSF.2017.11.

[5] Nicolas Papernot and Thomas Steinke. Hyperparameter tuning with renyi differential privacy, 2022. URL https://arxiv.org/abs/2110.03620.

[6] Nicola Rieke, Jonny Hancox, Wenqi Li, Fausto Milletarì, Holger R. Roth, Shadi Albarqouni, Spyridon Bakas, Mathieu N. Galtier, Bennett A. Landman, Klaus Maier-Hein, Sébastien Ourselin, Micah Sheller, Ronald M. Summers, Andrew Trask, Daguang Xu, Maximilian Baust, and M. Jorge Cardoso. The future of digital health with federated learning. *npj Digital Medicine*, 3(1), September 2020. ISSN 2398-6352. doi: 10.1038/s41746-020-00323-1. URL http://dx.doi.org/10.1038/s41746-020-00323-1.

[7] Jianyu Wang and Gauri Joshi. Cooperative sgd: A unified framework for the design and analysis of communication-efficient sgd algorithms, 2019. URL https://arxiv.org/abs/1808.07576.

[8] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications, 2019. URL https://arxiv.org/abs/1902.04885.