Federated Learning of Gboard Language Models with Differential Privacy

Zheng Xu*, **Yanxiang Zhang***, **Galen Andrew**, **Christopher A. Choquette-Choo**, **Peter Kairouz**, **H. Brendan McMahan**, **Jesse Rosenstock**, **Yuanbo Zhang**
Google

Abstract

We train language models (LMs) with federated learning (FL) and differential privacy (DP) in the Google Keyboard (Gboard). We apply the DP-Follow-the-Regularized-Leader (DP-FTRL) (Kairouz et al., 2021b) algorithm to achieve meaningfully formal DP guarantees without requiring uniform sampling of client devices. To provide favorable privacy-utility trade-offs, we introduce a new client participation criterion and discuss the implication of its configuration in large scale systems. We show how quantile-based clip estimation (Andrew et al., 2021) can be combined with DP-FTRL to adaptively choose the clip norm during training or reduce the hyperparameter tuning in preparation for training. With the help of pretraining on public data, we train and deploy more than twenty Gboard LMs that achieve high utility and ρ−zCDP privacy guarantees with $\rho \in (0.2, 2)$, with two models additionally trained with secure aggregation (Bonawitz et al., 2017). We are happy to announce that all the next word prediction neural network LMs in Gboard now have DP guarantees, and all future launches of Gboard neural network LMs will require DP guarantees. We summarize our experience and provide concrete suggestions on DP training for practitioners.

1 Introduction

FL and Gboard LMs. In cross-device federated learning (FL), client devices collaboratively train a model without directly exchanging their local data (Kairouz et al., 2019). Google Keyboard (Gboard) was an early adopter of FL to train models that improve the user experience, following data minimization principles (Bonawitz et al., 2021) to protect users' privacy from some risks. Language models (LMs) are trained with FL to support various features in Gboard, including Next Word Prediction (NWP), Smart Compose (SC), and On-The-Fly rescoring (OTF). As illustrated in Fig. 1, NWP (Hard et al., 2018) uses an LM to suggest a word, which is triggered after a previous word is committed; SC provides longer inline suggestions to accelerate typing, which can be triggered per character when the confidence is high; OTF is used to re-rank the candidate words generated during typing before a word is committed.

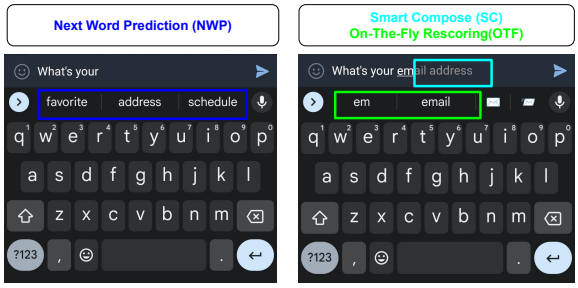


Figure 1: Gboard features supported by language models: NWP for next word, SC for inline suggestion, and OTF for candidates re-ranking.

Models, metrics and tasks. We train LMs with the same neural network (NN) architecture described in (Hard et al., 2018): a one-layer LSTM/CIFG of 670 hidden neurons, with input and output word-based embeddings of dimension 96. OTF LMs use a larger vocabulary ($\sim$ 30K words) compared to NWP LMs ($\sim$ 10–20K words); the number of parameters for models with a 10K/20K/30K vocabulary is 2.4M/4.4M/6.4M, respectively. SC is a downstream task that reuses NWP LMs without any retraining from data. We train NWP LMs and OTF LMs from populations of devices categorized by language and location. For example, en-US NWP denotes the task of training NWP model on data generated by devices using English in the United States.

Federated Averaging (FedAvg) (McMahan et al., 2017) and variants (Wang et al., 2021) are popular FL training algorithms in practice. In each communication *round*, the server will orchestrate a small subset of client devices for training and

* Equal contribution, alphabetical order. Correspondence to {xuzheng,zhangyx}@google.com.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 5: Industry Track, pages 629–639

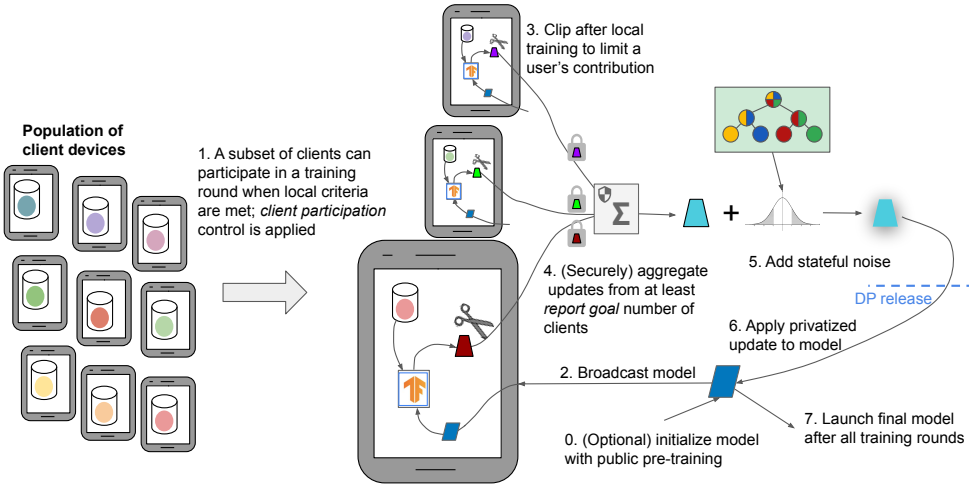


Figure 2: System overview of federated learning of Gboard language models with differential privacy and secure aggregation.

aggregate the resulting model deltas to update the global model. In a successful round, the system guarantees the number of clients participating in training is at least as large as the configured *report goal* (Bonawitz et al., 2019). A model is typically tested and deployed after training for several thousands of rounds. Top-1 in-vocab accuracy is used to track the utility during training and additional metrics for A/B testing are introduced in Sec. 3.

DP and DP-FTRL. Differential privacy (DP) can be combined with FL to provide a formal guarantee that the trained model will not memorize specific users' data, which provides stronger privacy protection by executing data anonymization principles (Bonawitz et al., 2021; Wang et al., 2021). Ramaswamy et al. (2020) applied DP-FedAvg (McMahan et al., 2018; Geyer et al., 2017), a variant of DP-SGD (Abadi et al., 2016) for user/client-level DP, to train production LMs in FL. Ramaswamy et al. (2020) demonstrated anonymization via empirical auditing techniques by Carlini et al. (2019) but did not provide a formal DP guarantee. Achieving a strong formal DP guarantee for DP-FedAvg would require privacy amplification-by-sampling, which necessitates sampling clients uniformly at random on each round. However, a cross-device FL system has limited control over client sampling as devices have to satisfy local criteria such as being charging and connected to an unmetered network to be eligible for participation (Bonawitz et al., 2019; Balle et al., 2020). In contrast, we deploy a recent algorithm, DP-FTRL (Kairouz et al., 2021b), allowing us to achieve strong privacy and utility for production models without uniform sampling assumptions.

Contributions. We discuss our strategy and experience of training Gboard LMs with FL and DP. We introduce an algorithm that enables adaptive clipping (Andrew et al., 2021) in DP-FTRL (Kairouz et al., 2021b) (Sec. 2.1), which can reliably estimate the clip norm to reduce hyperparameter tuning. We discuss the impact of scaling up computation and limiting client participation (Sec. 2.2), and identify the algorithm and system configurations for the regime of strong privacy and utility. We also successfully apply pre-training (Sec. 2.3) to improve privacy and utility, which is (to the best of our knowledge) the first time pretraining is applied to training a DP model directly from users' data.

We combine DP-FTRL with secure aggregation (SecAgg) to further strengthen the data minimization properties of our approach (Sec. 2.4). Fig. 2 provides a system overview of the techniques for training Gboard language models with federated learning and differential privacy. Finally, we summarize concrete suggestions for practitioners training differentially private models to deploy in production in (Sec. 2.5), and present and analyze twenty Gboard LMs trained with formal DP guarantees (Sec. 3). We are happy to announce that all the next word prediction neural network LMs in Gboard now have DP guarantees, and all future launches of Gboard neural network LMs will require DP guarantees.

2 DP FL in Practice

2.1 DP-FTRL and adaptive clipping

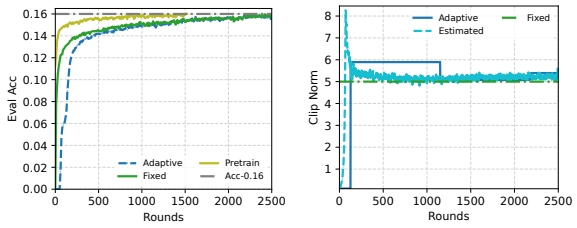
As described in Alg. 1, we apply DP-FTRL in FL by modifying the FedAvg algorithm: clip the model update Δ, and add noise when updating the global

Algorithm 1 Federated DP-FTRL with adaptive clipping

input : report goal m, learning rate for model weights on client η_c and on server η_s, momentum $\beta = 0.9$, noise multiplier for model delta z_Δ, total number of rounds T, restart rounds $\mathcal{R} = \{128 + 1024i, i = 0, 1, \ldots\}$, quantile based norm estimation C^0, target quantile $\gamma = 0.5$, learning rate for norm $\eta_\gamma = 0.2$, noise stddev for clip estimation $\sigma_b = m/20$

Initialize model θ^0, momentum buffer $\bar{\Delta}^0 = 0$, clip norm $C_\theta = C^0$
Initialize tree $\mathcal{T}_\theta$ with z_Δ, C_θ, and $\mathcal{T}_b$ with σ_b
for each round $t = 0, 1, 2, \ldots, T$ **do**
  $\mathcal{Q}^t \leftarrow$ (at least m users for this round)
  for each user $i \in \mathcal{Q}^t$ **in parallel do**
    $(\Delta_i^t, b_i^t) \leftarrow \text{ClientUpdate}(i, \theta^t, \eta_c, C_\theta, C^t)$
  //Update model weights with noise addition
  $\tilde{\Delta}^t = \frac{1}{m}\text{PrivateSum}\left(\mathcal{T}_\theta, \sum_{i\in\mathcal{Q}^k} \Delta_i^k, k \in [0, t]\right)$
  $\bar{\Delta}^t = \beta\bar{\Delta}^{t-1} + \tilde{\Delta}^t$, $\theta^{t+1} \leftarrow \theta^0 + \eta_s\bar{\Delta}^t$
  //Estimate quantile-based norm
  $\tilde{b}^t = \frac{1}{m}\text{PrivateSum}\left(\mathcal{T}_b, \sum_{i\in\mathcal{Q}^k} b_i^k, k \in [0, t]\right)$
  $C^{t+1} \leftarrow C^0 \cdot \exp\left(-\eta_\gamma(\tilde{b}^t - t\gamma)\right)$
  //Restart and adjust clip norm
  if $t \in \mathcal{R}$ **then**
    $C_\theta \leftarrow C^{t+1}$
    Restart tree $\mathcal{T}_\theta$ and $\mathcal{T}_b$ with updated C_θ

function ClientUpdate(i, θ_0, η, C_θ, C)
  $\theta \leftarrow \theta_0$
  $\mathcal{G} \leftarrow$ (user i's local data split into batches)
  for batch $g \in \mathcal{G}$ **do**
    $\theta \leftarrow \theta - \eta\nabla\ell(\theta; g)$
  $\Delta \leftarrow \theta - \theta_0$
  $b \leftarrow \mathbb{I}_{||\Delta||\le C}$
  $\Delta' \leftarrow \Delta \cdot \min\left(1, \frac{C_\theta}{||\Delta||}\right)$ //Clipping
  return (Δ', b)



(a) Evaluation accuracy (b) Clip norm

Figure 3: DP training of the en-GB NWP model. Adaptive clipping performs similar to fixed clipping, while achieves slightly weaker guarantees. Pre-training significantly reduces the number of rounds to reach the utility target, and achieves stronger guarantees.

model. Two additional hyperparameters are introduced for DP: the *clip norm* C, which bounds the norm of Δ, and the *noise multiplier* z, which determines the standard deviation zC for the added Gaussian noise. We discuss clip norm in this section and defer the discussion of noise multiplier and other privacy related hyperparameters to Sec. 2.2.

Andrew et al. (2021) introduced an adaptive clipping method that automatically adjusts the clip norm each round by privately estimating the norm of the model delta at a targeted quantile. However, adaptive clipping cannot be directly applied to DP-FTRL as the tree-based noise addition in DP-FTRL assumes a fixed clip norm across rounds. We integrate adaptive clipping in DP-FTRL through restarts, where the quantile estimate C^t is continually tracked but only becomes an active clip norm C_θ upon tree restarting. As both the aggregated model delta $\tilde{\Delta}^t$ and the quantile $\tilde{b}^t$ use tree-based noise, we can directly use the privacy accounting in (Kairouz et al., 2021b) by applying the noise transformation in Thm. 1 in App. A.

In practice, Alg. 1 slightly inflates the noise for the model from zC to $z_\Delta C$ and requires *restarts* that complicate the privacy accounting for DP-FTRL. Moreover, we find that a fixed clip norm can achieve comparable or slightly better model utility, and is more robust in experiments with large report goal. For example, adaptive clipping for the de-DE NWP model experiences catastrophic failure and makes no progress in the first 1000 rounds.

Nevertheless, adaptive clipping can reduce hyperparameter tuning for many tasks when privacy budget allows. Fig. 3 shows the evaluation accuracy and corresponding clip norm for DP training the en-GB NWP model with report goal 6500 and noise multiplier 7. The adaptive clip curve starts from a small initial clip norm to avoid catastrophic failure due to large initial noise and eventually catches up on accuracy. The estimated clip norm (quantile $\gamma = 0.5$) stabilizes and we can fix the clip norm to 5 based on the estimated value. The clip norm is relatively insensitive, especially when tuning to-

gether with the server learning rate. However, clip norm can have a wide tuning range across tasks and models, and quantile-based estimation is still useful for estimating a clip norm to be fixed.

2.2 DP parameters and system configuration

The privacy guarantees of DP-FTRL (Kairouz et al., 2021b) are affected by several factors: noise multiplier z, number of total rounds T, max participation (MaxP) of a client, and min separation (MinS) of rounds between the participation of the same client. The noise multiplier is a conventional parameter for controlling privacy-utility trade-off: large noise achieves strong privacy guarantees but can potentially hurt the utility. Achieving the same utility with smaller rounds T can significantly improve the privacy guarantees. Next, we discuss the effect of MaxP and MinS, and the privacy-utility-computation trade-off for system configuration.

Client participation. DP-FTRL achieves strong privacy if each client only participates once during training, or the number of client participation is limited when a client can participate multiple times. Two parameters are introduced to characterize client participation for DP-FTRL: the maximum participations (MaxP) of a client in all training rounds and the minimum round separation (MinS) between any single client's two participations. MaxP and MinS are correlated as MaxP is upper bounded by rounds T divided by MinS. In general, for fixed rounds T, decreasing MaxP and increasing MinS can lead to stronger privacy guarantees without changing utility. In addition, Cho et al. (2023) suggests potential advantage of increasing MinS for utility.

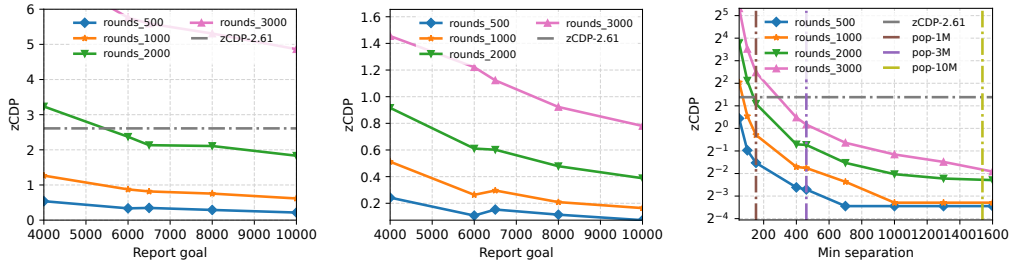
When using the worst-case MaxP estimated by rounds T divided by MinS, Fig. 4c shows increasing MinS can achieve stronger privacy measured by smaller zCDP values. However, the maximum MinS is limited by the population size divided by the number of clients per round lower bounded by the report goal. For example, when the report goal is 6500 for small population of around 10^6, MinS has to be smaller than 153 rounds, so strong privacy guarantees are difficult to achieve when training for 3000 rounds. While we cannot measure the precise population size in the FL system due to client dynamics, we estimate the population size of various Gboard tasks as ranging from 0.8 million to 16.6 million in Tab. 1.

Report goal. We study report goal for privacy-computation trade-off based on a hypothesis used in (McMahan et al., 2018; Kairouz et al., 2021b; Xu et al., 2022): for sufficiently large data, the utility is approximately non-decreasing if the noise multiplier and clients per round (lower bounded by report goal) proportionally increase. We provide empirical justification to this hypothesis by comparing the evaluation accuracy of two training runs: one with a report goal of 500 and noise multiplier of 0.54, versus another of report goal 6500 and noise multiplier 7. On more than three Gboard language tasks, we observed that the final utility remains similar, or slightly better for larger report goals. Moreover, using a larger report goal speeds up learning at the beginning of training. Based on the hypothesis, we plot Figs. 4a and 4b by linearly increasing report goal and noise multiplier, and assuming the MinS is set to the maximum possible value (population divided by report goal) for strong privacy. Though a large report goal can limit the MinS, it generally leads to stronger privacy guarantees for reasonable population size and total rounds.

System configuration. According to Figs. 4a and 4b, we choose a large report goal of 6500 supported by the large scale FL systems and aim for maximum MinS for DP-FTRL. To control MinS in practice, a timer is introduced on clients in the FL system so that a client will only become eligible to participate in training (again) after a certain period of time has passed. McMahan and Thakurta (2022) used a timer period of 24 hours to train the es-ES NWP model, which led to an observed MinS of 313. The MinS of es-ES is upper bounded by $4.21M/6500 \sim 647$ and can be potentially improved by increasing the timer period. We increase the timer period in the unit of 24 hours due to the uneven diurnal participation pattern (Yang et al., 2018; Zhu et al., 2022), and generally observe that MinS can proportionally increase with the timer period before reaching the possible maximum. However, there are many factors in the FL system that may affect the wall clock training speed, which makes it challenging to optimize the timer period to maximize MinS.

2.3 Public pretraining

We explore pretraining on public data for production models, which were shown to substantially improve model utility in DP simulations (Li et al., 2021; De et al., 2022; Xu et al., 2022; Wang et al.,



(a) Population 1M (b) Population 3M (c) Report goal 6500

Figure 4: The effect of population size, number of rounds, report goals, and min separation on DP-FTRL privacy guarantees. For a fixed number of rounds to achieve utility target, increasing report goal and min separation can achieve stronger guarantees measured by smaller zCDP.

2023). We pretrain a model for each Gboard language task using the multi-lingual C4 dataset (Raffel et al., 2019; Xue et al., 2020) collected from public web pages. Fig. 3a shows that pretraining can reduce ~ 1000 rounds to reach a given utility threshold under the same noise multiplier, which can significantly improve the privacy guarantees as shown in Fig. 4.

We additionally observe that: (1) it is challenging to fine-tune from a pretrained model when the word embeddings are shared for input and output to reduce the parameter size of LMs for on-device deployment; (2) the accuracy may decrease in the first a few rounds of fine-tuning; (3) pretraining helps with diminishing marginal returns: at some point further pretraining does not necessarily improve the final performance. Therefore, we use models with separate input and output embeddings and pretrain with half of the C4 dataset for Gboard LMs.

2.4 Combining with secure aggregation

Secure aggregation (SecAgg) (Bonawitz et al., 2017) ensures that the central server can only access the aggregated update from a large set of clients, preventing inspection of individual client updates. We combine SecAgg and DP-FTRL to provide strong data minimization and anonymization protection (Bonawitz et al., 2021). To avoid the suboptimal privacy cost from the ℓ_2 norm increase of the discretized vector in SecAgg, we follow the protocol of (Kairouz et al., 2021a) for discretizing, flattening, and modularly clipping[1] the client model updates—this introduces minimal norm inflation later accounted in DP-FTRL. The large report goal requirement for strong DP guarantees is challenging for SecAgg in practice, which requires a slightly different system configuration. The SecAgg training speeds we observe are still notably slower, and we leave for future work potential improvements such as compression for communication efficiency (Chen et al., 2022), new DP methods to reduce report goal (Choquette-Choo et al., 2022), and embedding compression to reduce round time (Shu and Nakayama, 2017).

[1]In our current implementation, there is a very small chance that modular operator in SecAgg will inflate the sensitivity. The problem will be fixed by an additional element-wise clipping of the flattened vector.

2.5 Recommended strategies and practices

We summarize our strategy for training Gboard LMs with DP. (1) Pre-train the model on public datasets if possible. (2) Choose the maximum noise multiplier that meets the utility target based on small report goal simulation experiments on public datasets that is similar to the production task. (3) Based on the target number of rounds and estimated population, linearly increase the report goal and noise multiplier to meet the privacy target, and choose a large report goal supported by the system. If the privacy target is unachievable, fix the report goal to maximum, and increase the noise multiplier to target on a model with suboptimal utility. (4) Estimate the possible maximum MinS based on chosen report goal and estimated population, and configure the timer period to approach the MinS; use previous experience of model training speed if applicable. (5) If the hyperparameters (e.g., learning rates) are known from previous experiments or simulation on public datasets, apply DP-FTRL with adaptive clipping (Alg. 1) without manual tuning to try meet the privacy and utility goals. Note that Alg. 1 needs to account the noise inflation and restart for privacy guarantees. (6) If Alg. 1 fails or stronger privacy and utility are desirable, we can run a few small report goal experiments with Alg. 1 that tune quantile γ and server learning rate η_s, select the best learning rate, and fix the clip norm

based on the estimation; and run DP-FTRL with large report goals. (7) SecAgg can be used for all experiments, and precise MaxP and MinS are computed by post-processing for privacy accounting.

3 Deploying DP LMs

A/B test metrics. We introduce metrics in A/B test to measure the utility of Gboard LMs. (1) *Picked Rate (PRate)*: the ratio of picked candidates among the NWP predictions; or SC predictions when it is triggered. (2) *Accuracy (Acc)*: the ratio of candidates matching the final committed words among the NWP model predictions. (3) *Trigger Rate*: the ratio of words with SC triggered among all committed words, which is an important metric when PRate is fixed. (4) *Word Modified Ratio (WMR)*: the ratio of words being modified during typing or after committed; improvement is shown by reduction. (5) *Word Per Minute (WPM)*: the number of committed words per minute.

Privacy guarantees. Same as (McMahan and Thakurta, 2022), the zero-one device neighboring relationship ((Kairouz et al., 2021b, definition 1.1)) is adopted for DP. For user's with a single device, device-level DP corresponds directly to user-level DP. Our privacy guarantee holds for all well-behaved clients during training, and we do not account for privacy cost of modest amount of hyperparameter tuning. DP is measured by the zero-Concentrated DP (zCDP) (Bun and Steinke, 2016) guarantee that has been used by US census bureau (US Census Bureau, 2021), and can be easily converted to (ϵ, δ)-DP. We use the privacy accounting in (Kairouz et al., 2021b, appendix D) implemented in Tensorflow Privacy (TFP Authors, 2022), and follow the guidelines outlined in (Ponomareva et al., 2023, Sec. 5.3) to report detailed narratives of privacy guarantees in App. C.

Experimental setup. We use the implementation in App. B, and apply the strategy in Sec. 2.5 to train Gboard LMs with DP. We present NWP results in Tab. 1, and OTF results in Tab. 2. As Smart Compose (SC) reuses NWP LMs, SC has the same DP guarantees as NWP models by the post-processing property (Dwork et al., 2014). Following es-ES NWP model in (McMahan and Thakurta, 2022), we choose noise multiplier 7 and report goal 6500 based on simulation in (Kairouz et al., 2021b) on public StackOverflow dataset (TFF Authors, 2022b). We pretrain the models on public datasets and configure the timer period to control client participation, separately for different tasks. We use DP-FTRL with adaptive clipping and small report goal 500 to tune server learning rate and estimate the clip norm. Interestingly, we observe the learning rate and clip norm to be consistent for various Gboard LMs, and tuning seems to be unnecessary. DP-FTRL with fixed clip and large report goal is used to run the final model for deployment.

Result analysis. All NWP and OTF models in Tabs. 1 and 2 are trained with stronger guarantees (smaller zCDP) compared to zCDP > 2.6 used by US Census Bureau (US Census Bureau, 2021). For five NWP models in Europe (DE, GB, FR, IT, PT), the DP NN models significantly improve the utility compared to previous N-gram models. On en-US, pt-BR and en-IN, DP NN models also achieve comparable, or slightly better utility compared to their non-private versions as the strong models. SecAgg is successfully applied to en-US and es-ES, and can achieve good privacy-utility trade-off with a smaller number of rounds, likely due to the system configuration that results in more clients per round. However, SecAgg is also notably slower. There is a general positive correlation between the estimated population size and privacy guarantees.

However, only a few tasks approach the possible maximum MinS for strong privacy guarantees, which highlights the challenge of both estimating population and controlling client participation. Longer training rounds are often used for NWP (compared to OTF) as the non-private NN baselines are strong, and to improve the downstream SC performance. As an example, we train es-ES NWP for 1900 rounds with a pretrained model, while the previous models (McMahan and Thakurta, 2022) is trained for 2000 rounds without pretraining. Our es-ES NWP model slightly improves the utility measured by PRate and Acc, and improves the zCDP bound from 0.81 to 0.35 due to the larger MinS by timer configuration. We highlight that our es-ES model at round 1240 already achieves similar NWP utility and a strong privacy guarantee, but the utility of SC keeps improving with training. Compared to the previous model in (McMahan and Thakurta, 2022), our model improves the SC trigger rate by 4.23% at round 1240, and 9.51% at round 1900.

4 Concluding remarks

We discuss our experience and summarize our strategy for training production Gboard LMs with FL

NWP	Rounds	Utility		Privacy		Est. Pop. (M)	BaseModel
		PRate(+%)	Acc(+%)	MinS/MaxP/Timer	zCDP		
de-DE	930	8.28	12.49	212 / 4/ 48h	0.48	3.24	N-gram
en-GB	980	3.26	7.72	226 / 4 / 72h	0.48	2.38	
fr-FR	1280	3.78	8.50	180 / 5 / 72h	0.89	2.79	
it-IT	1620	3.98	9.86	303 / 5 / 72h	0.71	3.32	
pt-PT	530	3.99	7.82	54 / 8 / 48h	1.86	0.83	
es-ES	1900	0.29	0.48	526 / 3 / 144h	0.35	4.21	zCDP 0.81
es-ES*	1750	0.32	0.56	349 / 4 / 144h	0.52		
en-US	2800	-0.39	0.11	371 / 7 / 48h	1.31	13	No-DP NN
en-US*	1360	-0.30	0.15	622 / 2 / 144h	0.25		
pt-BR	3600	0.18	0.29	909 / 3 / 144h	0.45	16.6	
en-IN	1290	0.19	0.40	170 / 6 / 96h	1.14	7.72	
es-MX	1980	-0.15	0.29	343 / 5 / 96h	0.64	9.96	
es-AR	640	0.25	3.50	90 / 5 / 96h	0.84	4.09	Mix

Table 1: Live A/B tests of DP NWP models. Utility shows the improvement from previously deployed models; privacy shows the key parameters and corresponding device-level zCDP; all models are trained by DP-FTRL with report goal of 6500 and noise multiplier of 7; en-US*/es-ES* are trained with SecAgg in addition to DP; the base model in AR is a mix of N-gram and No-DP NN models.

OTF	Rounds	Utility		Privacy		
		WMR(-%)	WPM(+%)	MinS/MaxP/Timer	zCDP	DP-$\epsilon(\delta = 10^{-10})$
de-DE	1170	1.01	0.59	206 / 5 / 48h	0.89	9.01
en-GB	1220	1.99	0.38	206 / 5 / 72h	0.89	9.01
es-ES	1280	1.03	0.60	197 / 5 / 48h	0.89	9.01
fr-FR	1300	1.83	0.67	290 / 4 / 72h	0.61	7.31
it-IT	1360	1.39	0.80	188 / 5 / 48h	0.89	9.01
ru-RU	870	0.72	0.34	327 / 3 / 48h	0.32	5.13
pt-PT	430	1.71	0.32	54 / 7 / 48h	0.99	9.56

Table 2: Live A/B tests of DP OTF models. Utility shows the WMR decrease and WPM increase; privacy shows the key parameters and corresponding zCDP bound; all models are trained with DP-FTRL with report goal of 6500 and noise multiplier of 7; estimated population for ru-RU is 6.63M and other tasks can be found in Tab. 1.

and DP. We propose an algorithm applying adaptive clipping (Andrew et al., 2021) in DP-FTRL (Kairouz et al., 2021b) to reduce the hyperparamter tuning. We discuss the impact on privacy and utility of several important factors: the clip norm, report goal, client participation, and pre-training. Our study highlights the importance of system and algorithm co-design for differential privacy in practice, the challenges of tuning in FL systems, and opportunities to improve the scalability and stability of FL with DP and/or SecAgg. More than twenty LMs with formal DP guarantees are trained and launched to support Gboard NWP, SC, and OTF features, including en-US and es-ES NWP models additionally with SecAgg. Our experience demonstrates the possibility of training DP models for practical applications when a large scale system is available for large scale data. Therefore, Gboard is introducing and enforcing a new policy: DP has to be applied in all future training and launching of Gboard LMs.

Acknowledgement

The authors would like to thank Stanislav Chiknavaryan, Adria Gascon, Zachary Garrett, and Timon Van Overveldt for infrastructure configuration support; Swaroop Ramaswamy, Om Thakkar, Abhradeep Thakurta for early discussion on models and algorithms; Jeremy Gillula for internal review process; Xu Liu, Shumin Zhai, and Daniel Ramage for leadership support.

References

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and

Li Zhang. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318.

Galen Andrew, Om Thakkar, H Brendan McMahan, and Swaroop Ramaswamy. 2021. Differentially private learning with adaptive clipping. *Conference on Neural Information Processing Systems (NeurIPS)*.

Borja Balle, Peter Kairouz, Brendan McMahan, Om Thakkar, and Abhradeep Guha Thakurta. 2020. Privacy amplification via random check-ins. *Advances in Neural Information Processing Systems*, 33:4623–4634.

Kallista Bonawitz, Peter Kairouz, Brendan McMahan, and Daniel Ramage. 2021. Federated learning and privacy: Building privacy-preserving systems for machine learning and data science on decentralized data. *Queue*, 19(5):87–114.

Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konečnỳ, Stefano Mazzocchi, Brendan McMahan, et al. 2019. Towards federated learning at scale: System design. *Proceedings of machine learning and systems*, 1:374–388.

Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. 2017. Practical secure aggregation for privacy-preserving machine learning. In *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1175–1191.

Mark Bun and Thomas Steinke. 2016. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Theory of Cryptography Conference*, pages 635–658. Springer.

Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. 2019. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 267–284.

Wei-Ning Chen, Christopher A Choquette Choo, Peter Kairouz, and Ananda Theertha Suresh. 2022. The fundamental price of secure aggregation in differentially private federated learning. In *International Conference on Machine Learning*, pages 3056–3089. PMLR.

Yae Jee Cho, Pranay Sharma, Gauri Joshi, Zheng Xu, Satyen Kale, and Tong Zhang. 2023. On the convergence of federated averaging with cyclic client participation. *arXiv preprint arXiv:2302.03109*.

Christopher A Choquette-Choo, H Brendan McMahan, Keith Rush, and Abhradeep Thakurta. 2022. Multi-epoch matrix factorization mechanisms for private machine learning. *arXiv preprint arXiv:2211.06530*.

Soham De, Leonard Berrada, Jamie Hayes, Samuel L Smith, and Borja Balle. 2022. Unlocking high-accuracy differentially private image classification through scale. *arXiv preprint arXiv:2204.13650*.

DP Team. 2022. Google's differential privacy libraries. https://github.com/google/differential-privacy.

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

Robin C Geyer, Tassilo Klein, and Moin Nabi. 2017. Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557*.

Andrew Hard, Kanishka Rao, Rajiv Mathews, Swaroop Ramaswamy, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. 2018. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*.

Peter Kairouz, Ziyu Liu, and Thomas Steinke. 2021a. The distributed discrete gaussian mechanism for federated learning with secure aggregation. In *International Conference on Machine Learning*, pages 5201–5212. PMLR.

Peter Kairouz, Brendan Mcmahan, Shuang Song, Om Thakkar, Abhradeep Thakurta, and Zheng Xu. 2021b. Practical and private (deep) learning without sampling or shuffling. In *International Conference on Machine Learning (ICML)*, pages 5213–5225.

Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kaylee Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaïd Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konecný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Mariana Raykova, Hang Qi, Daniel Ramage, Ramesh Raskar, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. 2019. Advances and open problems in federated learning. *CoRR*, abs/1912.04977.

Xuechen Li, Florian Tramer, Percy Liang, and Tatsunori Hashimoto. 2021. Large language models can be strong differentially private learners. *arXiv preprint arXiv:2110.05679*.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017.

Communication-efficient learning of deep networks from decentralized data. In *AISTATS*, pages 1273–1282. PMLR.

Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. 2018. Learning differentially private recurrent language models. In *International Conference on Learning Representations (ICLR)*.

Brendan McMahan and Abhradeep Thakurta. 2022. Federated learning with formal differential privacy guarantees.

Natalia Ponomareva, Hussein Hazimeh, Alex Kurakin, Zheng Xu, Carson Denison, H. Brendan McMahan, Sergei Vassilvitskii, Steve Chien, and Abhradeep Thakurta. 2023. How to dp-fy ml: A practical guide to machine learning with differential privacy.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*.

Swaroop Ramaswamy, Om Thakkar, Rajiv Mathews, Galen Andrew, H Brendan McMahan, and Françoise Beaufays. 2020. Training production language models without memorizing user data. *arXiv preprint arXiv:2009.10031*.

Raphael Shu and Hideki Nakayama. 2017. Compressing word embeddings via deep compositional code learning. *arXiv preprint arXiv:1711.01068*.

TFF Authors. 2022a. TensorFlow Federated. `https://github.com/tensorflow/federated`.

TFF Authors. 2022b. TensorFlow Federated StackOverflow dataset. `https://www.tensorflow.org/federated/api_docs/python/tff/simulation/datasets/stackoverflow`.

TFP Authors. 2022. TensorFlow Privacy. `https://github.com/tensorflow/privacy`.

US Census Bureau. 2021. Disclosure avoidance for the 2020 census: An introduction.

Boxin Wang, Yibo Jacky Zhang, Yuan Cao, Bo Li, H Brendan McMahan, Sewoong Oh, Zheng Xu, and Manzil Zaheer. 2023. Can public large language models help private cross-device federated learning? *arXiv preprint arXiv:2305.12132*.

Jianyu Wang, Zachary Charles, Zheng Xu, Gauri Joshi, H Brendan McMahan, Blaise Aguera y Arcas, Maruan Al-Shedivat, Galen Andrew, Salman Avestimehr, Katharine Daly, et al. 2021. A field guide to federated optimization. *arXiv:2107.06917*.

Zheng Xu, Maxwell Collins, Yuxiao Wang, Liviu Panait, Sewoong Oh, Sean Augenstein, Ting Liu, Florian Schroff, and H Brendan McMahan. 2022. Learning to generate image embeddings with user-level differential privacy. *arXiv preprint arXiv:2211.10844*.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2020. mt5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*.

Timothy Yang, Galen Andrew, Hubert Eichner, Haicheng Sun, Wei Li, Nicholas Kong, Daniel Ramage, and Françoise Beaufays. 2018. Applied federated learning: Improving google keyboard query suggestions. *arXiv preprint arXiv:1812.02903*.

Chen Zhu, Zheng Xu, Mingqing Chen, Jakub Konečnỳ, Andrew Hard, and Tom Goldstein. 2022. Diurnal or nocturnal? federated learning of multi-branch networks from periodically shifting distributions. In *International Conference on Learning Representations*.

A Privacy accounting for adaptive clipping

Theorem 1 (Privacy Accounting for Adaptive Clipping (Andrew et al., 2021)). *One step of DP-FTRL with adaptive clipping using σ_b noise standard deviation on the clipped counts $\sum b_i^t$ and z_Δ noise multiplier on the vector sums $\sum \Delta_i^t$ is equivalent to one step of non-adaptive DP-FTRL with noise multiplier z if we set $z_\Delta = \left(z^{-2} - (2\sigma_b)^{-2}\right)^{-1/2}$.*

B Implementation.

We use the open source implementation of DP-FTRL in Tensorflow Privacy (TFP Authors, 2022) integrated with Tensorflow Federated (TFF Authors, 2022a) as a DP aggregator for federated learning. Conceptually, DP-FTRL adds noise to the summation of updates across rounds, i.e., *PrivateSum* in Alg. 1. Instead of tracking the noise and summation separately, PrivateSum is implemented to only track the noise and updates $\tilde{\theta}^{t-1}$ by adding the residual of noise between round t and round $t-1$. This design makes it easy to integrate with various optimizer choices, for example, momentum that is important for utility; and also allows ephemeral access of model deltas without directly storing unnoised states.

C Reporting privacy guarantees

This section clarifies the nuances of the reported DP guarantees following the guidelines outlined in (Ponomareva et al., 2023, Sec. 5.3)

1. **DP setting**. This a central DP guarantee where the service provider is trusted to correctly implement the mechanism.
2. **Instantiating the DP Definition**
   (a) *Data accesses covered*: The DP guarantee applies to all well-behaved clients[2] in a single training run. We do not account for hyperparameter tuning in our guarantees. Public multilingual C4 data (Raffel et al., 2019; Xue et al., 2020) is used for pre-training.
   (b) *Final mechanism output*: Only the final model checkpoint is released for production launches, however the mechanism's output is technically the full sequence of privatized gradients, and so the guarantee also applies at this level, and hence all intermediate models are protected (including those sent to devices participating in federated learning).
   (c) *Unit of privacy*. Device-level DP is considered, i.e., the notion of adjacency is with respect to arbitrary training datasets on each client device, and the device might have an arbitrarily large local dataset containing arbitrary training examples. For user's with a single device, this corresponds directly to user-level DP; for devices shared with multiple users, this provides a stronger notion of DP than user-level; for a user with multiple devices that happen to both participate in training the model, the notion is weaker, but group privacy can be used to obtain a user-level guarantee.
   (d) *Adjacency definition for "neigbouring" datasets*: We use the zero-out definition (Kairouz et al., 2021b). This is a a special form of the add-or-remove definition, where neighboring data sets differ by addition/removal of a single client. In the absence of a client at any training step, we assume that the client's model update gets replaced with the all zeros vector. This assumption enforces a subtle modification to the traditional definition of the add/remove notion of DP which allows neighboring data sets to have the same number of records.
3. **Privacy accounting details**
   (a) *Type of accounting used*: Both $\rho-$zCDP (Bun and Steinke, 2016) accounting, and PLD accounting (DP Team, 2022) for $(\epsilon, \delta)-$DP are used.
   (b) *Accounting assumptions* : Each client only participates limited times during the training, and there are at least a min-separation number of rounds between two consecutive participation of a

[2]Clients that faithfully follow the algorithm including participation limits. Due to the design of the algorithm, a mis-behaved client does not adversely affect the DP guarantee of any well-behaved clients.

client, i.e., MaxP and MinS as discussed in Sec. 2.2. Client participation is enforced by a timer on clients in the cross-device FL system.

(c) *The formal DP statement*: The launched Gboard LMs have $\rho-$zCDP range in (0.2, 2). We also transform zCDP to $(\epsilon, \delta)-$DP by PLD accounting (DP Team, 2022): given $\delta = 10^{-10}$, the smallest zCDP $\rho = 0.25$ corresponds to DP $\epsilon = 4.49$; the largest zCDP $\rho = 1.86$ corresponds to DP $\epsilon = 13.69$.

(d) *Transparency and verifiability*: We open sourced our core implementation code in TensorFlow Federated and Tensorflow Privacy. Key portions of the cross-device FL system are also open sourced.