EFFICIENT LANGUAGE MODEL ARCHITECTURES FOR DIFFERENTIALLY PRIVATE FEDERATED LEARNING

Jae Hun Ro, Srinadh Bhojanapalli, Zheng Xu, Yanxiang Zhang, & Ananda Theertha Suresh
Google Research, New York
{jaero,bsrinadh,xuzheng,zhangyx,theertha}@google.com

ABSTRACT

Cross-device federated learning (FL) is a technique that trains a model on data distributed across typically millions of edge devices without data leaving the devices. SGD is the standard client optimizer for on device training in cross-device FL, favored for its memory and computational efficiency. However, in centralized training of neural language models, adaptive optimizers are preferred as they offer improved stability and performance. In light of this, we ask if language models can be modified such that they can be efficiently trained with SGD client optimizers and answer this affirmatively.

We propose a scale-invariant *Coupled Input Forget Gate* (SI CIFG) recurrent network by modifying the sigmoid and tanh activations in the recurrent cell and show that this new model converges faster and achieves better utility than the standard CIFG recurrent model in cross-device FL in large scale experiments. We further show that the proposed scale invariant modification also helps in federated learning of larger transformer models. Finally, we demonstrate the scale invariant modification is also compatible with other non-adaptive algorithms. Particularly, our results suggest an improved privacy utility trade-off in federated learning with differential privacy.

1 INTRODUCTION

Federated learning (FL) is a technique that trains a model on data distributed across devices without data leaving the device (Konecný et al., 2016; McMahan et al., 2017a). FL has been applied in a variety of diverse settings, including language-based applications (Hard et al., 2018b; Chen et al., 2019; Kairouz et al., 2019; Li et al., 2020a; Shah et al., 2020). Specifically, we examine cross-device FL (Kairouz et al., 2021b), where local clients are edge devices with limited resources and computing power, which can number in the millions. Previous works on language modeling in cross-device FL often use small recurrent-based models of less than 10M parameters (Hard et al., 2018b; Reddi et al., 2020; Xu et al., 2023), while more recent works leverage a variety of efficient techniques for training larger Transformer-based models (Hilmkil et al., 2021; Ro et al., 2022). In this work, we investigate modular strategies applicable to various model architectures for improving training of both small and large models in cross-device FL.

Existing works on improving FL usually focus on developing better optimizers FedAvg, FedProx, Mime, FedDyn etc (Li et al., 2020b; Reddi et al., 2021; Karimireddy et al., 2021; Acar et al., 2021). While advanced optimizers are typically used in the server, (e.g., in the optimizer FedAdam, Adam optimizer is used in the server), in practice, the preferred client optimizer is often SGD for its memory efficiency. Note that using an adaptive optimizer like Adam in clients requires storing first and second moments of gradients, which improves the memory requirement considerably. However, neural language models such as recurrent LSTMs (Yu et al., 2019) or Transformers (Vaswani et al., 2017), typically require more memory intensive adaptive optimizers, such as Adagrad or Adam that store both the first and second moment of gradients, and suffer in performance when trained with SGD (Zhang et al., 2020). Hence we ask the question: Can we achieve the best of both worlds and effectively train expressive architectures with memory efficient optimizers for language modeling in FL?

Li et al. (2022) studied this question in the context of training centralized Transformer encoder models, such as BERT, and proposed using Scale Invariant Transformers for improved optimization of Transformers using SGD. Using Scale Invariant Transformers, they were able to use SGD to obtain a similar performance to that of standard Transformers using the Adam optimizer. Naturally, this raises the question if there exists scale invariant version of other neural architectures, e.g. LSTMs that can be optimized well with simple SGD.

Federated learning can also be combined with other privacy techniques to provide strong privacy protection to various threat models (Zhang et al., 2023; Bonawitz et al., 2022). Differential privacy (DP) (Dwork et al., 2006) is a statistical framework that provides rigorous guarantees for privacy protection and is adopted in federated learning to prevent models from memorizing individual information (McMahan et al., 2017b; Ramaswamy et al., 2020; El Ouadrhiri & Abdelhadi, 2022; Wei et al., 2020; Girgis et al., 2021). More recently, by applying the family of DP-Follow The Regularized Leader (DP-FTRL) algorithms (Kairouz et al., 2021a; Choquette-Choo et al., 2023) that have strong privacy-utility trade-offs without relying on sampling assumptions, meaningful formal differential privacy guarantees have been achieved for production language models in practical cross-device systems (Xu et al., 2023).

2 Our contributions

Improving LSTM architectures for FL. Long Short-Term Memory (LSTM) (Hochreiter & Schmidhuber, 1997b) language models are often used in large scale FL studies due to their small size (McMahan et al., 2017a; Hard et al., 2018a;b; Kairouz et al., 2021a; Xu et al., 2023). In particular, Hard et al. (2018b) proposed to use Coupled Input Forget Gate (CIFG) LSTMs for federated learning for mobile keyboard predictions for its improved parameter and computational efficiency over the vanilla LSTM. Motivated by this, we develop a novel scale-invariant CIFG model (SI CIFG) with modified activation functions for FL.

Application to FL. In cross-device FL, each client typically runs multiple steps of local SGD on their local data to produce model parameter updates. These updates are then typically combined at the server with a federated optimizer such as FedAdam (Reddi et al., 2021). This raises an important question: does our SI CIFG offer any advantages in this setting where one of the optimizers is SGD and the other is an adaptive optimizer like Adam? We show that this is indeed the case and that both our proposed SI CIFG as well as the already existing scale-invariant Transformer (Li et al., 2022) (SI Transformer), using scale-invariant attentions, perform significantly better than their standard counterparts on a variety of experiments by improving convergence speeds in large scale FL experiments, while remaining robust to higher learning rates and heterogeneous networks.

Training with differential privacy. FL models are trained with differential privacy using the DP-FTRL algorithm (Kairouz et al., 2021a). In this scenario, while the local steps are still carried out via SGD, the model updates from clients are additionally clipped and aggregated with noise at the server. We show that scale invariant models also outperform their standard counterparts on experiments in a large-scale FL system with differential privacy.

3 Scale Invariant Architectures

3.1 Previous scale invariant architectures

In this section, we briefly review Scale Invariant Transformers (Li et al., 2022). Recall that a function f is scale invariant if $f(ax) = f(x)$ for any scalar $a > 0$. Let n be the input sequence length and d be the hidden dimension of the Transformer model. Recall that for a given input $\boldsymbol{X} \in \mathbb{R}^{d \times n}$, a Transformer computes self attention as follows:

$$\text{Attn}(\boldsymbol{X}) = \text{SoftMax}\left((\boldsymbol{W}_Q \boldsymbol{X})^\top . \boldsymbol{W}_K \boldsymbol{X}\right). \tag{1}$$

Here $\boldsymbol{W}_Q$ and $\boldsymbol{W}_K$ are the Query and Key projections, respectively. This operation is not scale invariant, as scaling the weights $(\boldsymbol{W}_Q, \boldsymbol{W}_K)$ changes the output attention probabilities. Li et al. (2022) proposed the following alternative attention computation:

$$\text{SI-Attn}(\boldsymbol{X}) = \text{N}\left(\text{ReLU}\left((\boldsymbol{W}_Q\boldsymbol{X})^\top.\boldsymbol{W}_K\boldsymbol{X}\right)\right). \tag{2}$$

Here, N is the row-wise normalization operator - $\text{N}(\boldsymbol{A})_{ij} = \frac{\boldsymbol{A}_{ij}}{\sum_j \boldsymbol{A}_{ij}}$. In particular, Li et al. (2022) replaced the softmax in attention computation, with the ReLU activation followed by row-wise normalization. This modifies the attention computation to be scale invariant. They further modify the Transformer to be a Pre-LN activation model and use ReLU activation instead of GeLU in the feedforward layers. We use the same architecture in our experiments.

However, Li et al. (2022) tested their method only on centralized encoder models (BERT). In this paper, we will extend the results to decoder-only Transformers trained using a language modeling objective in cross-device FL.

3.2 New scale invariant architectures

Inspired by the Scale Invariant Transformer, we now design a novel Scale Invariant version of the CIFG architecture we call SI CIFG. We note that the same changes from the Scale Invariant Transformer do not apply to the CIFG as due to architecture differences, scale sensitivity arises from different functions for CIFG models.

We focus on CIFG networks for their improved parameter and computational efficiency over the vanilla LSTMs. The CIFG network uses a single gate to control self-connections in both input and recurrent cells, which reduces the number of parameters per cell by 25% (Hochreiter & Schmidhuber, 1997a; Cho et al., 2014; Greff et al., 2017). The shared gates increase efficiency, with little to no impact on quality, which is critical in the typically resource constrained edge device environment of cross-device FL. Moreover, we expect that our proposed changes can also be directly applied to the LSTM model.

First, we review the basic CIFG before our proposed architecture changes. Recall that for a given time step t and input $x_t \in \mathbb{R}^d$, the CIFG forward pass can be written as follows:

$$\begin{aligned}
f_t &= \sigma(W_f x_t + U_f h_{t-1} + b_f) && \textit{forget gate} \\
i_t &= 1 - f_t && \textit{coupled input forget gate} \\
o_t &= \sigma(W_o x_t + U_o h_{t-1} + b_o) && \textit{output gate} \\
c_t &= f_t \odot c_{t-1} + i_t \odot \tanh(W_c x_t + U_c h_{t-1} + b_c) && \textit{cell state} \\
h_t &= o_t \odot \tanh(c_t)
\end{aligned}$$

where d and h are the input and hidden dimensions, respectively, and $W \in \mathbb{R}^{h\times d}$, $U \in \mathbb{R}^{h\times h}$, and $b \in \mathbb{R}^h$ are the cell's trainable weight and bias parameters. Here σ and $\tanh$ are Sigmoid and Tanh activation functions, respectively. This architecture is sensitive to input scale, mainly because of the non-linearities in the σ and $\tanh$ activations. We first propose modifying the activation functions to be scale invariant by replacing σ with Relu and $\tanh$ with linear activation. However, this no longer guarantees that intermediate outputs of different gates are normalized. To further ensure that the intermediate features are normalized we propose using a Max-Normalization - MaxN, which normalizes each entry of the feature vector using its max absolute value along the hidden dimension. Formally,

$$\text{MaxN}(x)_i = \frac{x_i}{\max_{j\in[d]} |x_j|}. \tag{3}$$

Based on this, we propose the following scale invariant replacement for σ activation.

$$\text{SI-}\sigma(x)_i = \text{MaxN}(\text{Relu}(x))_i = \frac{\text{Relu}(x)_i}{\max_{j\in[d]}(\text{Relu}(x)_j)}. \tag{4}$$

Similarly, we also propose a scale invariant version of $\tanh$.

$$\text{SI-}\tanh(x)_i = \text{MaxN}(x)_i = \frac{x_i}{\max_{j\in[d]}(|x_j|)}. \tag{5}$$

It is straightforward to see that both SI-σ and SI-$\tanh$ are scale invariant functions and we provide a short proof for completeness.

Proposition 3.1. *Both SI-σ and SI-$\tanh$ are scale invariant functions.*

Proof. Let $a > 0$. Then for any $i \in d$, $\text{Relu}(ax)_i = a\text{Relu}(x)_i$ and hence,

$$\text{SI-}\sigma(ax)_i = \frac{\text{Relu}(ax)_i}{\max_{j\in[d]}(\text{Relu}(ax)_j)} = \frac{a\text{Relu}(x)_i}{a\max_{j\in[d]}(\text{Relu}(x)_j)} = \frac{\text{Relu}(x)_i}{\max_{j\in[d]}(\text{Relu}(x)_j)} = \text{SI-}\sigma(x)_i.$$

The calculations for SI-$\tanh$ are are similar and omitted. □

4 Experiments with Federated Learning

We report results for experiments using scale invariant architectures in large scale FL in both simulation and live production experiments. For simulations, we train a language model on the English Stack Overflow federated dataset, containing questions and answers from the forum grouped by username, provided from TensorFlow Federated (TFF) (TFF, 2018). For live production experiments, we train an English language model on millions of virtual keyboard user devices and follow the same settings and FL requirements for client participation as Hard et al. (2018b). All experiments were implemented using the open-source FedJAX (Ro et al., 2021b) and TFF libraries.

4.1 Federated experiments on public datasets

For experiments on the Stack Overflow federated dataset, we compare the following models:

- CIFG 19M: Coupled Input Forget Gate variant of LSTM with 19M trainable parameters with 1 layer of size 2048, embedding size 1024, and tied input and output embeddings (Press & Wolf, 2017).
- SI CIFG 19M: Modified CIFG 19M using SI-σ and SI-$\tanh$ activations.
- Transformer 21M: Transformer with 21M trainable parameters with 6 layers, 8 attention heads, MLP size 2048, embedding size 512, and tied input and output embeddings.
- SI Transformer 21M: Modified Transformer 21M using SI-Attn.

We use WordPiece (Wu et al., 2016) for subword tokenization with a vocabulary size of $4K$ to avoid potential bottlenecks in embeddings for larger vocabularies. For FL training, we use FedAdam (Reddi et al., 2021) which uses Adam (Kingma & Ba, 2014) for the server optimizer and SGD for the client optimizer with the same settings used by Reddi et al. (2021), with the exception of learning rates. We then sweep over learning rates for each model with 5 different random seeds for client sampling with 500 clients per round for $3K$ communication rounds and maximum sequence length of 20. Details on speicifc hyperparameter settings and sweeps can be seen in Appendix A.

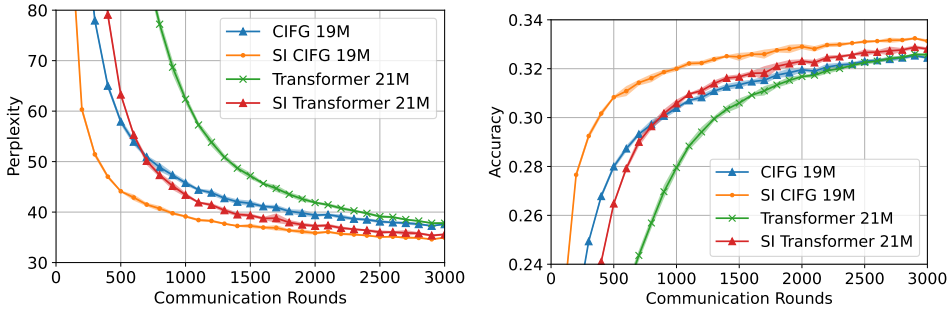

Figure 1: Perplexity and accuracy on the Stack Overflow test dataset with shading indicating standard deviation over 5 random seeds.

We report perplexity and accuracy, discounting end-of-sequence tokens, on the Stack Overflow test dataset over $3K$ communication rounds in Figure 1 with final values in Table 2 (Appendix A). We

observe that applying Scale Invariance significantly increases the rate of convergence for both the Transformer and CIFG, surpassing their respective base counterparts within 100 communication rounds. Our proposed SI CIFG yields the best final quality and has the fastest convergence speed by far. We next continue to live production experiments, where the network of clients is much larger and more heterogeneous than simulation.

4.2 LIVE PRODUCTION EXPERIMENTS

For live production experiments for cross-device FL on English virtual keyboard client devices, similar to Hard et al. (2018b), we compare the following models:

- CIFG 9M: CIFG with 9M trainable parameters with 1 layer of size 2048, embedding size 512, and tied input and output embeddings.
- SI CIFG 9M: Modified CIFG 9M using SI-σ and SI-$\tanh$ activations.
- Transformer 11M: Transformer with 11M trainable parameters with 3 layers, 8 attention heads, MLP size 2048, embedding size 512, and tied input and output embeddings.
- SI Transformer 11M: Modified Transformer 11M using SI-Attn.

We use smaller sizes here compared to our previous simulation experiments due to stricter resource constraints on client devices (Hard et al., 2018b; Ro et al., 2021a). Additionally, we also apply stochastic 8-bit uniform quantization (Alistarh et al., 2017; Suresh et al., 2017) on the upload of model updates from client to server due to tighter communication bottlenecks on mobile devices. We use Fast WordPiece (Song et al., 2021) for subword tokenization with a vocabulary size of $4K$ as it has been shown to be faster than WordPiece, allowing for more steps of training within the maximum time limit allocated for client devices. Again, we use the FedAdam algorithm with 500 clients per round for $3K$ communication rounds with maximum sequence length of 20. For more details on hyperparameters, refer to Appendix B.

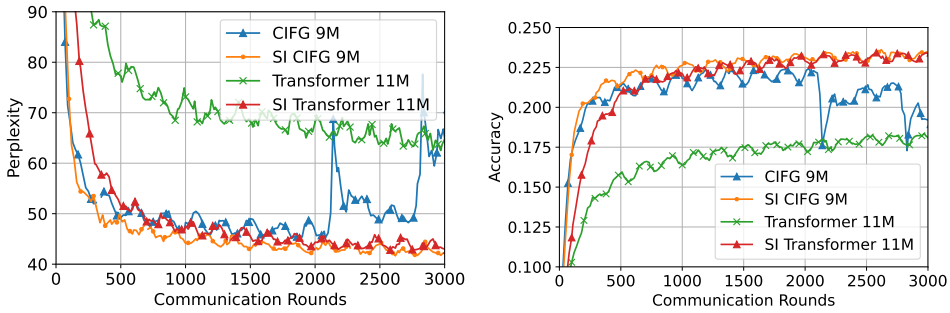

Figure 2: Perplexity and accuracy from live experiments on English virtual keyboard devices training from randomly initialized model weights.

We report perplexity and accuracy for training the models from randomly initialized parameters on the population of English virtual keyboard client devices over $3K$ communication rounds in Figure 2 with final values in Table 3 (Appendix B). The scale invariant architectures surpass their base counterparts within 100 communication rounds and converge to significantly higher qualities. While the base CIFG diverges in training at $2K$ rounds, which could be attributed to a number of potential issues (Pascanu et al., 2013) when training recurrent models with SGD on client devices, our proposed SI CIFG trains smoothly, significantly outperforms the other models within 200 rounds, and converges to the best final quality. This improved training stability could be due to robustness to out-sized client updates in the SI-σ and SI-$\tanh$ activations.

5 EXPERIMENTS WITH DIFFERENTIALLY PRIVATE FEDERATED LEARNING

In this section, we apply our proposed scale invariant architectures to differentially private (DP) FL. Specifically, we apply the DP variant of Follow-The-Regularized-Leader (DP-FTRL) *Online TreeAgg*

proposed by Kairouz et al. (2021a). For live production experiments, we train an English language model on millions of virtual keyboard user devices and mostly follow the same setup as Xu et al. (2023) for DP FL and compare the following models:

- CIFG 6M: CIFG with 6M trainable parameters with 1 layer of size 670, embedding size 96, and vocabulary size of $30K$.
- SI CIFG 6M: Modified CIFG 6M using SI-σ and SI-$\tanh$ activations.

For training, we use 6500 clients per round and the same noise multiplier of 7.0 for $3K$ communication rounds with maximum sequence length of 10 with word tokenization using a vocabulary size of $30K$. The client optimizer is SGD with learning rate of 0.5 and the server optimizer is SGD with momentum with learning rate 1.0 and momentum 0.9. We set the noise multiplier in the DP-FTRL algorithm to obtain a z-CDP privacy of 1.05. We refer readers to Bun & Steinke (2016) for the definition of z-CDP and Kairouz et al. (2021a) for the privacy guarantee calculations. For more details and hyperparameter configurations, refer to Appendix C. Before applying DP FL training, we first pre-train the models on the public English Colossal Clean Crawled Corpus (C4) (Raffel et al., 2019) dataset for $370K$ steps and start DP FL training from the pre-trained checkpoint. We report perplexity and in-vocab-accuracy, discounting out-of-vocabulary and end-of-sequence tokens, for DP FL training on the population of English virtual keyboard client devices over $3K$ communication rounds in Figure 3 with final values in Table 4 (Appendix C).

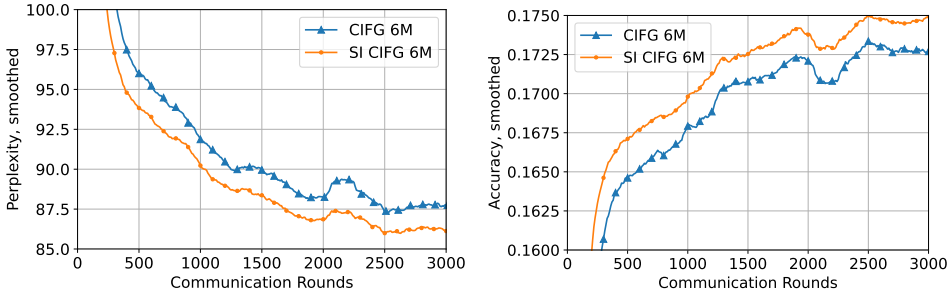

Figure 3: Smoothed perplexity and in-vocab-accuracy from DP live experiments on English virtual keyboard devices.

In the DP FL setting, our proposed SI CIFG consistently outperforms the base CIFG and under the same privacy budget, achieves better utility, measured by perplexity and accuracy.

6 Conclusion

We applied scale invariance to a variety of neural architectures and proposed a novel CIFG-LSTM architecture (SI CIFG) and evaluated their performance on a variety of cross-device and differentially private large scale FL experiments. We demonstrated that using scale invariant architectures in federated language modeling can significantly accelerate and improve model convergence, with our proposed SI CIFG consistently achieving the best performance and convergence speed. We hope that this study will motivate further studies into training larger models privately and effectively with federated learning.

References

Durmus Alp Emre Acar, Yue Zhao, Ramon Matas Navarro, Matthew Mattina, Paul N Whatmough, and Venkatesh Saligrama. Federated learning based on dynamic regularization. *arXiv preprint arXiv:2111.04263*, 2021.

Dan Alistarh, Demjan Grubic, Jerry Li, Ryota Tomioka, and Milan Vojnovic. Qsgd: Communication-efficient sgd via gradient quantization and encoding. In I. Guyon, U. Von

Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/6c340f25839e6acdc73414517203f5f0-Paper.pdf.

Kallista Bonawitz, Peter Kairouz, Brendan Mcmahan, and Daniel Ramage. Federated learning and privacy. *Communications of the ACM*, 65(4):90–97, 2022.

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

Mingqing Chen, Ananda Theertha Suresh, Rajiv Mathews, Adeline Wong, Cyril Allauzen, Françoise Beaufays, and Michael Riley. Federated learning of n-gram language models. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pp. 121–130, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/K19-1012. URL https://aclanthology.org/K19-1012.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1179. URL https://aclanthology.org/D14-1179.

Christopher A Choquette-Choo, Arun Ganesh, Ryan McKenna, H Brendan McMahan, Keith Rush, Abhradeep Guha Thakurta, and Zheng Xu. (amplified) banded matrix factorization: A unified approach to private training. *arXiv preprint arXiv:2306.08153*, 2023.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, 2006.

Hubert Eichner, Tomer Koren, Brendan McMahan, Nathan Srebro, and Kunal Talwar. Semi-cyclic stochastic gradient descent. In *International Conference on Machine Learning*, pp. 1764–1773. PMLR, 2019.

Ahmed El Ouadrhiri and Ahmed Abdelhadi. Differential privacy for deep and federated learning: A survey. *IEEE access*, 10:22359–22380, 2022.

Antonious Girgis, Deepesh Data, Suhas Diggavi, Peter Kairouz, and Ananda Theertha Suresh. Shuffled model of differential privacy in federated learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 2521–2529. PMLR, 2021.

Klaus Greff, Rupesh K. Srivastava, Jan Koutník, Bas R. Steunebrink, and Jürgen Schmidhuber. Lstm: A search space odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, 28(10): 2222–2232, 2017. doi: 10.1109/TNNLS.2016.2582924.

Andrew Hard, Chloé M Kiddon, Daniel Ramage, Francoise Beaufays, Hubert Eichner, Kanishka Rao, Rajiv Mathews, and Sean Augenstein. Federated learning for mobile keyboard prediction, 2018a. URL https://arxiv.org/abs/1811.03604.

Andrew Hard, Kanishka Rao, Rajiv Mathews, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. Federated learning for mobile keyboard prediction. *ArXiv*, abs/1811.03604, 2018b. URL https://api.semanticscholar.org/CorpusID:53207681.

Agrin Hilmkil, Sebastian Callh, Matteo Barbieri, Leon René Sütfeld, Edvin Listo Zec, and Olof Mogren. Scaling federated learning for fine-tuning of large language models. In *International Conference on Applications of Natural Language to Data Bases*, 2021.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997a.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997b.

Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, K. A. 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, Zaid Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konečný, 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. Advances and open problems in federated learning. 2019. URL https://arxiv.org/abs/1912.04977.

Peter Kairouz, Brendan Mcmahan, Shuang Song, Om Thakkar, Abhradeep Thakurta, and Zheng Xu. Practical and private (deep) learning without sampling or shuffling. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 5213–5225. PMLR, 18–24 Jul 2021a. URL https://proceedings.mlr.press/v139/kairouz21b.html.

Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends in Machine Learning*, 14 (1–2):1–210, 2021b.

Sai Praneeth Karimireddy, Martin Jaggi, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian U Stich, and Ananda Theertha Suresh. Breaking the centralized barrier for cross-device federated learning. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 28663–28676. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/f0e6be4ce76ccfa73c5a540d992d0756-Paper.pdf.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

Jakub Konecný, H. B. McMahan, Felix X. Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *ArXiv*, abs/1610.05492, 2016.

Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020a. doi: 10.1109/MSP.2020.2975749.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks, 2020b.

Zhiyuan Li, Srinadh Bhojanapalli, Manzil Zaheer, Sashank Reddi, and Sanjiv Kumar. Robust training of neural networks using scale invariant architectures. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 12656–12684. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/li22b.html.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. In Aarti Singh and Jerry Zhu (eds.), *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pp. 1273–1282. PMLR, 20–22 Apr 2017a. URL https://proceedings.mlr.press/v54/mcmahan17a.html.

H Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. *arXiv preprint arXiv:1710.06963*, 2017b.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In Sanjoy Dasgupta and David McAllester (eds.), *Proceedings of the 30th*

International Conference on Machine Learning, volume 28 of *Proceedings of Machine Learning Research*, pp. 1310–1318, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR. URL https://proceedings.mlr.press/v28/pascanu13.html.

Ofir Press and Lior Wolf. Using the output embedding to improve language models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pp. 157–163, Valencia, Spain, April 2017. Association for Computational Linguistics. URL https://aclanthology.org/E17-2025.

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

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

Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečnỳ, Sanjiv Kumar, and H Brendan McMahan. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*, 2020.

Sashank Reddi, Zachary Burr Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečný, Sanjiv Kumar, and Brendan McMahan (eds.). *Adaptive Federated Optimization*, 2021. URL https://openreview.net/forum?id=LkFG3lB13U5.

Jae Ro, Mingqing Chen, Rajiv Mathews, Mehryar Mohri, and Ananda Theertha Suresh. Communication-efficient agnostic federated averaging. In *Interspeech*, 2021a.

Jae Ro, Theresa Breiner, Lara McConnaughey, Mingqing Chen, Ananda Suresh, Shankar Kumar, and Rajiv Mathews. Scaling language model size in cross-device federated learning. In *Proceedings of the First Workshop on Federated Learning for Natural Language Processing (FL4NLP 2022)*, pp. 6–20, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.fl4nlp-1.2. URL https://aclanthology.org/2022.fl4nlp-1.2.

Jae Hun Ro, Ananda Theertha Suresh, and Ke Wu. FedJAX: Federated learning simulation with JAX. *arXiv preprint arXiv:2108.02117*, 2021b.

Aishanee Shah, Andrew Hard, Cameron Nguyen, Ignacio Lopez Moreno, Kurt Partridge, Niranjan Subrahmanya, Pai Zhu, and Rajiv Mathews. Training keyword spotting models on non-iid data with federated learning. In *Interspeech*, 2020.

Xinying Song, Alex Salcianu, Yang Song, Dave Dopson, and Denny Zhou. Fast WordPiece tokenization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 2089–2103, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.160. URL https://aclanthology.org/2021.emnlp-main.160.

Ananda Theertha Suresh, Felix X. Yu, Sanjiv Kumar, and H. Brendan McMahan. Distributed mean estimation with limited communication. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3329–3337. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/suresh17a.html.

TFF. Tensorflow federated, 2018. URL https://www.tensorflow.org/federated.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H Yang, Farhad Farokhi, Shi Jin, Tony QS Quek, and H Vincent Poor. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE transactions on information forensics and security*, 15:3454–3469, 2020.

Yonghui Wu, Mike Schuster, Z. Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason R. Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Gregory S. Corrado, Macduff Hughes, and Jeffrey Dean. Google's neural machine translation system: Bridging the gap between human and machine translation. *ArXiv*, abs/1609.08144, 2016.

Zheng Xu, Yanxiang Zhang, Galen Andrew, Christopher Choquette, Peter Kairouz, Brendan Mcmahan, Jesse Rosenstock, and Yuanbo Zhang. Federated learning of gboard language models with differential privacy. In Sunayana Sitaram, Beata Beigman Klebanov, and Jason D Williams (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track)*, pp. 629–639, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-industry.60. URL https://aclanthology.org/2023.acl-industry.60.

Yong Yu, Xiaosheng Si, Changhua Hu, and Jianxun Zhang. A review of recurrent neural networks: Lstm cells and network architectures. *Neural computation*, 31(7):1235–1270, 2019.

Jingzhao Zhang, Sai Praneeth Karimireddy, Andreas Veit, Seungyeon Kim, Sashank Reddi, Sanjiv Kumar, and Suvrit Sra. Why are adaptive methods good for attention models? *Advances in Neural Information Processing Systems*, 33:15383–15393, 2020.

Yuanbo Zhang, Daniel Ramage, Zheng Xu, Yanxiang Zhang, Shumin Zhai, and Peter Kairouz. Private federated learning in gboard. *arXiv preprint arXiv:2306.14793*, 2023.

A FEDERATED EXPERIMENTS ON PUBLIC DATASETS DETAILS

For all models and experiments with the Stack Overflow federated dataset, we used the followed fixed hyperparameters

- Number of clients per round = 500: Number of clients sampled per communication round of FL training.
- Client batch size = 10: Batch size used during local steps of training on client data.
- Number of client epochs = 1: Number of epochs of training on client data.
- Number of client batches = 120: Maximum number of client batches to train on until number of client epochs is reached.
- Maximum sequence length = 20: Maximum allowed sequence length. Shorter sequences are padded and longer sequences are truncated to this.
- Client optimizer = SGD
- Server optimizer = Adam with β_1 at 0.9, β_2 at 0.999, and epsilon at $1e^{-8}$.

Table 1 details the hyperparameter configurations swept over per model, where the selected hyperparameters were chosen based on the lowest loss on the heldout split of the Stack Overflow federated dataset after $3K$ rounds of training averaged over 5 random seeds. Table 2 reports the final evaluation results using these selected hyperparameters on the Stack Overflow test dataset.

B LIVE PRODUCTION EXPERIMENT DETAILS

For all models and experiments with the live English virtual keyboard user population, we used the followed fixed hyperparameters. We note that due to the nature of live production experiments and longer feedback times, we were not able to run any extensive hyperparameter sweeps and re-used

Table 1: Selected hyperparameters for each model. The values in [ ] are the possible hyperparameter values searched over.

Model	Client learning rate $[0.1, 0.5, 1.0, 2.0]$	Server learning rate $[0.001, 0.01]$
CIFG 19M	0.1	0.001
SI CIFG 19M	0.1	0.001
Transformer 21M	0.5	0.001
SI Transformer 21M	2.0	0.01

Table 2: Perplexity and accuracy on the Stack Overflow test dataset after $3K$ communication rounds.

Model	Perplexity	Accuracy%
CIFG 19M	35.5 ± 0.2	33.1 ± 0.1
SI CIFG 19M	$\mathbf{33.6 \pm 0.2}$	$\mathbf{33.6 \pm 0.1}$
Transformer 21M	34.6 ± 0.1	33.4 ± 0.0
SI Transformer 21M	33.7 ± 0.1	33.5 ± 0.0

many common settings used in previous experiments. Table 3 reports the final evaluation results using these hyperparameters averaged over the final 100 of $3K$ communication rounds to account for daytime variability (Eichner et al., 2019).

- Number of clients per round = 500: Number of clients sampled per communication round of FL training.
- Client batch size = 10: Batch size used during local steps of training on client data.
- Number of client epochs = 1: Number of epochs of training on client data.
- Number of client batches = 120: Maximum number of client batches to train on until number of client epochs is reached.
- Maximum sequence length = 20: Maximum allowed sequence length.
- Client optimizer = SGD with learning rate 0.7.
- Server optimizer = Adam with learning rate 0.02, β_1 at 0.9, β_2 at 0.999, and epsilon at $1e^{-8}$.

Table 3: Perplexity and accuracy from live experiments on English virtual keyboard devices averaged with standard deviations over the final 100 communication rounds. *For base CIFG, we use the last 100 rounds before divergence.

Model	Perplexity	Accuracy%
*CIFG 9M	47.5 ± 1.1	21.8 ± 0.2
SI CIFG 9M	$\mathbf{42.2 \pm 0.2}$	$\mathbf{23.4 \pm 0.1}$
Transformer 11M	63.6 ± 0.9	18.2 ± 0.1
SI Transformer 11M	44.3 ± 0.7	23.2 ± 0.2

C Experiments with differentially private federated learning details

For all models and DP experiments with the live English virtual keyboard user population, we used the followed fixed hyperparameters. Again, due to the nature of live production experiments and longer feedback times, we were not able to run any extensive hyperparameter sweeps and re-used many common settings used in previous experiments. Table 4 reports the final evaluation results using these hyperparameters averaged over the final 100 of $3K$ communication rounds to account for daytime variability.

- Number of clients per round = 6500: Number of clients sampled per communication round of FL training.
- Client batch size = 10: Batch size used during local steps of training on client data.
- Clipping norm = 5.0: Fixed L2 norm that client updates are clipped up to.
- Maximum sequence length = 10: Maximum allowed sequence length. Decreased here since word tokenization is used instead of the typically longer subword tokenization.
- Client optimizer = SGD with learning rate 0.5.
- Server optimizer = SGD with momentum with learning rate 1.0 and momentum 0.9.

Table 4: Perplexity and in-vocab-accuracy from DP live experiments on English virtual keyboard devices averaged with standard deviations over the final 100 communication rounds.

Model	Perplexity	Accuracy%
CIFG 6M	88.0 ± 0.8	17.3 ± 0.1
SI CIFG 6M	$\mathbf{86.1 \pm 0.7}$	$\mathbf{17.5 \pm 0.1}$