Population Expansion for Training Language Models with Private Federated Learning

Tatsuki Koga [1] **Congzheng Song** [2] **Martin Pelikan** [2] **Mona Chitnis** [2]

Abstract

Federated learning (FL) combined with differential privacy (DP) offers machine learning (ML) training with distributed devices and with a formal privacy guarantee. With a large population of devices, FL with DP produces a performant model in a timely manner. However, for applications with a smaller population, not only does the model utility degrade as the DP noise is inversely proportional to population, but also the training latency increases since waiting for enough clients to become available from a smaller pool is slower. In this work, we thus propose expanding the population based on domain adaptation techniques to speed up the training and improves the final model quality when training with small populations. We empirically demonstrate that our techniques can improve the utility by 13% to 30% on real-world language modeling datasets.

1. Introduction

Federated learning (FL) (McMahan et al., 2017) enables training machine learning (ML) models using on-device data and is widely used in our daily lives as usage of mobile devices, e.g., smartphones, smart watches, and smart speakers, increases. Although FL, by design, does not require raw data to be transmitted from devices, privacy breaches can happen by transmitting model gradients to the central server. Thus, FL algorithms are modified to satisfy differential privacy (DP) (McMahan et al., 2018) to provide a formal privacy guarantee. We refer this learning framework as private federated learning (PFL).

Successful ML models trained with PFL typically require the number of devices sampled at each round, *cohort size*, to be large enough to reduce the detrimental impact of DP noise on the model utility (Anil et al., 2022). The requirement of large cohort size, which is easily met with hundreds of millions of devices, can be hard to fulfill for applications with device-constrained populations. For a motivating example, to train a language model (LM) with PFL for automatic speech recognition (ASR) system in a virtual assistant, the on-device training data are transcribed speech. For popular languages, such as English or Chinese, there are ample devices with transcriptions. However, for less popular languages such as Romanian or Swahili, the population with data is orders of magnitude smaller due to the limited speaker base. In such small populations, as we will show in Section 3.1, the server needs to spend much longer waiting for a full cohort of devices to become available in each iteration, which is impractical for models that require thousands of iterations to converge. Thus, PFL has the tradeoff among privacy, utility, and *latency* for device-constrained applications.

Our contributions In this work, we develop approaches to expand the population size to address the latency bottleneck for PFL in the device-constrained scenarios. We propose to use data from different applications than the target application to augment the training data, e.g. there are more devices with typed text than those with audio transcriptions as the messaging application is used more frequently than a virtual assistant. Population expansion for PFL has three benefits: (1) training will be faster as there are more devices available, (2) DP noise scale will be smaller from amplification by subsampling (Wang et al., 2019) by making population size larger, and (3) sampling error will be smaller. We explore combinations of various domain adaptation techniques and show that they outperform naively augmenting the devices from other sources. We focus on training LMs and evaluate the proposed approaches on public benchmark datasets including Reddits Comments and Common Voice. We demonstrate our methods can expand the population size by 10 times, which significantly reduces the latency and achieves better model utility.

1.1. Related Work

Prior works on domain adaptation in the LM applications focuses on centralized training. Jiang & Zhai (2007) explored

[1]UC San Diego, work done while interning at Apple [2]Apple. Correspondence to: Congzheng Song <csong4@apple.com>.

Workshop of Federated Learning and Analytics in Practice, colocated with 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA.

instance weighting with importance sampling to reweight the training objective for domain adaptation. Moore & Lewis (2010) selected and used a portion of non-domain-specific language data for domain-specific LM training. Moriokal et al. (2018) extended LM neural networks (NNs) to have domain-specific and domain-shared representations so that those representations are learned separately. Gururangan et al. (2022) focused on transformer model and modify the model architecture to have domain-specific layers. More recently, Chronopoulou et al. (2022) adopted hierarchical network structures for training on data from a larger number of domains, where models are gradually trained along with the hierarchy in a top-down manner.

With regards to domain adaptation in the federated setting, prior works address the setting where the clients and the server own data from different domains (Peng et al., 2019; Yao et al., 2022). Shen et al. (2021) extended the adversarial domain adaptation technique to the federated setting, but their main focus is cross-silo FL, where the number of clients is much smaller. Peterson et al. (2019) also proposed a domain adaptation technique in cross-silo FL with differential privacy, which properly combines general and specific models.

2. Preliminaries

Federated Learning (FL) (McMahan et al., 2017) enables model training on multiple devices, each having a separate dataset, without sharing on-device dataset with a central server. In particular, we focus on *cross-device* FL where the number of clients is very large, as opposed to cross-silo FL where client population is small. The standard iterative procedure for training machine learning models executes at each iteration t: (1) the central server samples a set of clients $\mathcal{C}_t$ from the population, (2) each sampled client $i \in \mathcal{C}_t$ downloads the shared model parameter θ_t from the server and locally trains the model on its own data to produce a local model θ_i, (3) each sampled client i sends back the model difference $\Delta_{t,i} = \theta_i - \theta_t$ to the server, and (4) the server aggregates the model differences as a "pseudo-gradient" $\Delta_t = \frac{1}{|\mathcal{C}_t|}\Delta_{t,i}$ and uses it to update θ_t with any standard optimizer.

Differential Privacy (DP) provides strong privacy protections for sensitive data on device. DP is formally defined as follows:

Definition 2.1 ((ϵ, δ)-DP (Dwork et al., 2006))**.** A randomized algorithm M satisfies (ϵ, δ)-DP if for any neighboring datasets D, D' and for any $S \subseteq \text{range}(M)$,

$$\Pr[M(D) \in S] \leq \exp(\epsilon) \Pr[M(D') \in S] + \delta.$$

We say two datasets $D, D' \in \mathcal{X}$ are neighboring if they differ on at most an individual's participation. Two additional steps are added to the FL algorithm to ensure a DP guarantee: (1) each sampled client clips the model difference before sending it back to have a bounded norm, and (2) the server applies a DP building block, commonly the Gaussian mechanism (Dwork & Roth, 2014), when aggregating the model differences to get the *noisy* pseudo-gradient. We focus on using the Gaussian mechanism for aggregating the model differences in this work. The noise variance is then calibrated by the moment accountant (Abadi et al., 2016; Mironov, 2017; Mironov et al., 2019) with fixed sampling rate q (fraction of clients sampled in each iteration), number of training iterations T, and privacy budgets (ϵ, δ).

3. Expanding Population in PFL

3.1. Device Sampling Latency

We first formulate how population size N impacts latency in PFL. In each round of PFL, *cohort size* $C \approx Nq$ of devices are sampled to participate in training, where q is the device sampling probability to provide an amplification on privacy (Wang et al., 2019). Server tends to over-sample by using a slightly larger $q > \frac{C}{N}$ to improve the latency. In reality, only a proportion of devices satisfying certain conditions (e.g. locked, charging and on Wi-Fi) are eligible for training and devices might dropout or abort training (Bonawitz et al., 2019; Paulik et al., 2021), and we denote this ratio of eligible devices as p. Therefore, if C is larger than Npq, we need to wait until enough devices become available to participate before updating the model.

More formally, assume $Npq < C$, we model the latency to wait $C - Npq$ devices more to become available and be sampled as follows. Let $m = N - Np$ be the number of current unavailable devices, $k = C - Npq$ be the number of devices needed for current PFL iteration.

Proposition 3.1. *Assume that the time for the i-th unavailable device becoming available and being sampled for training is $T_i \sim \text{Exponential}(\lambda)$. Let U_k be the random variable which describes the time when the first k devices become available and are sampled. Then*

$$\frac{1}{\lambda} \cdot \frac{C - Npq}{N(1-p)+1} \leq \mathbb{E}[U_k] \leq \frac{C}{\lambda(N-C)}. \quad (1)$$

We defer the proof to Appendix B. We use exponential time model since it is a common choice for modeling training time in the distributed scenario (Lee et al., 2017; Tandon et al., 2017; Nguyen et al., 2022).

From the above proposition we see that the expected latency U_k is inversely proportional to the population size, i.e. the smaller the population size, the longer the server needs to wait for enough devices to become available in each iteration. Figure 1 illustrates the relationship between latency and population size.

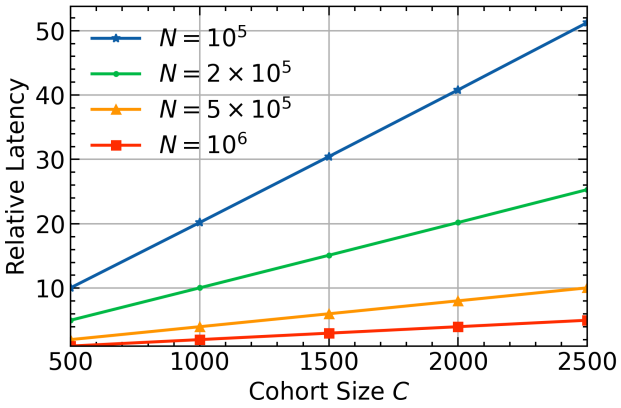

Figure 1. Relative latency estimated with Equation 1 for different cohort sizes C and population sizes N.

3.2. Domain Adaptation for Expanding Population

The small population situation happens often when building task-specific LMs, where potential data sources are scarce, e.g. training a LM on Swahili spoken texts as a part of virtual assistant system. It is a challenging task since only a small number of users are frequent users of a virtual assistant and have Swahili speech on their devices. Nonetheless, for such device-constrained locales, there could be other data sources, e.g., typed texts, with larger population. This motivates us to expand the population by exploiting another text source with a different distribution to train the LM for the target data source, which can be cast as a domain adaptation (DA) problem. Following DA convention, we denote data from other source applications with larger population as *source domain* $\mathcal{S}$, and data from target application with smaller population as *target domain* $\mathcal{T}$.

Goal We wish learn a global model that minimizes the objective $\mathbb{E}_{x\sim\mathcal{T}}[L(x)]$, where L is the loss function, with data from $\mathcal{S}\cup\mathcal{T}$ under a fixed privacy budget (ϵ,δ). The latency-utility trade-off should be much better than training in $\mathcal{T}$ alone.

Instance weighting (IW) Naively training with devices sampled from $\mathcal{S}\cup\mathcal{T}$ would bias towards $\mathcal{S}$ due to its larger population. To remedy this sampling bias, we apply instance weighting (Jiang & Zhai, 2007) on the training objective:

$$\mathbb{E}_{x\sim\mathcal{S}\cup\mathcal{T}}[w(x)L(x)], \tag{2}$$

where $w(x) = p_{\mathcal{T}}(x)/p_\pi(x)$ is the importance weight, $\pi\in\{\mathcal{S},\mathcal{T}\}$ denotes which domain x is from and $p_\pi(x)$ is the data density function for domain π. As $p_\pi(x)$ has to be estimated privately, we choose to approximate it with unigram likelihood $\hat{p}_\pi(x) = \prod_i \hat{u}_\pi(x_i)$ as unigram frequency $\hat{u}_\pi$ can be efficiently learned with a relative small privacy budget.

The product of unigrams in $\hat{p}_\pi(x)$ can lead to bipolarized density estimation, and thus unstable importance weights. We instead use relative importance weight (Yamada et al., 2013) to provide a more robust estimation:

$$w(x) = \frac{\hat{p}_{\mathcal{T}}(x)}{\alpha\hat{p}_{\mathcal{T}}(x) + (1-\alpha)\hat{p}_\pi(x)}, \tag{3}$$

where α is the proportion of the devices with data from $\mathcal{T}$.

The overall PFL training procedure with IW is: (1) learn the unigram frequency $\hat{u}_\pi$ for $\pi\in\{\mathcal{S},\mathcal{T}\}$ with privacy budget (ϵ_0,δ_0) which can be done with private federated statistics (McMillan et al., 2022), and (2) train model using objective weighted by Equation 3 with privacy budget $(\epsilon-\epsilon_0,\delta-\delta_0)$.

Pretrain in $\mathcal{S}$ and finetune in $\mathcal{T}$ (PT) Recent work (Ganesh et al., 2023) has shown that pretraining a model in a different domain to target domain with a large population reduces the amount of data required for private finetuning. We consider pretraining in $\mathcal{S}$ with a large cohort size C and finetune in $\mathcal{T}$ with a small cohort size αC so that the latency for finetuning stays roughly the same as pretraining. We enforce that the population of $\mathcal{S}$ and $\mathcal{T}$ to be disjoint so that both pretraining in $\mathcal{S}$ and finetuning in $\mathcal{T}$ can spend privacy budget of (ϵ,δ) with parallel composition (McSherry, 2009).

Instance weighted pretraining (IWPT) Domain adaptive pretraining (Gururangan et al., 2020) (DAPT) demonstrated the benefits of pretraining with in-domain data. However because the in-domain population is limited and it is inefficient to train with PFL, we consider instance weighted pretraining on $\mathcal{S}$ with objective weighted by Equation 3 as an approximation of DAPT.

4. Experiment

4.1. Datasets

To simulate a practical situation, we focus on using real-world datasets with user identifiers so that we can partition data naturally by users. In particular, we use two sources of data: (1) Reddits (Caldas et al., 2019) and (2) Common Voice (CV) (Ardila et al., 2020) to build two datasets for DA tasks. More data processing details are described in Appendix A.

SubReddits The first constructed DA dataset consists of only the Reddits dataset with different *SubReddit* topics. We treat a set of similar subreddits as a domain, where we choose stock-related subreddits {*Superstonk, amc-*

stock, wallstreetbets, GME, Wallstreetsilver} as $\mathcal{S}$ and news-related subreddits {*news, worldnews, politics*} as $\mathcal{T}$. As a result of the constrution, we have $117,708$ clients in total and $14,072$ clients (about 12%) have target domain data as well as source domain data.

CV&Reddits The other constructed DA dataset combines Reddit (typed texts) and CV (transcribed audios) which simulates the difference between spoken and typed texts domains. We treat texts from Reddits as $\mathcal{S}$ and texts from CV as $\mathcal{T}$. CV dataset has $68,312$ clients. We randomly select clients from Reddit dataset so that the total number of clients is 10 times more than the number of clients with Common Voice data.

4.2. Experiment Setup

Since there usually is a constraint on the client device storage and communication cost in real world applications, we consider a rather simple LSTM following (McMahan et al., 2018). We evaluate the performance of our approaches by the perplexity (PPL) in $\mathcal{T}$. We divide clients into training, validation, and test sets with the ratio of 6:2:2, where the hyper-parameters are tuned on validation set.

We consider two baselines with unweighted objective: (1) training with cohort sizes αC and C in $\mathcal{T}$ only where α is the proportion of the devices with data from $\mathcal{T}$, and (2) training with cohort size C in $\mathcal{S} \cup \mathcal{T}$. We also experiment the baseline (2) with domain adaptive layers proposed in domain-shared/domain-specific representations (DSDSR) (Moriokal et al., 2018) and DEMix (Gururangan et al., 2022).

To speed up the training process, we follow (McMahan et al., 2018) and set the cohort size C to be 5,000 for adjusting the magnitude of noise in the DP analysis and to be 400 for actual training. We set $\alpha = 0.1$ i.e. the ratio of population between $\mathcal{T}$ and $\mathcal{S}$. All experiments last for 2,000 server iterations and 1 client iteration. For fine-tuning experiments (PT and IWPT), we split the server iterations into 1,000 and 1,000 for pretraining and fine-tuning, respectively. We use FedAdam (Reddi et al., 2022) as the server optimizer with learning rate 0.1 and SGD as the client optimizer with learning rate 0.5.

We set the total privacy parameters to $(\epsilon, \delta) = (2, 10^{-6})$ throughout the experiments. The clipping bound of Gaussian mechanism in PFL is set to 0.5. For IW and IWPT, we allocate $(\epsilon_0, \delta_0) = (0.8, 0)$ for estimating unigrams with Geometric Mechanism (Ghosh et al., 2009), and $(\epsilon, \delta) = (1.2, 10^{-6})$ for model training. To bound the sensitivity for the unigram estimation, we use at most 5 sequences with each of which have a fixed length of 10 tokens.

Dataset	Approach	val PPL	test PPL
SubReddits	$\mathcal{T}$ w. αC	415.35	414.61
	$\mathcal{T}$ w. C	358.37	358.06
	$\mathcal{S} \cup \mathcal{T}$	398.82	400.90
	DSDSR	379.06	380.79
	DEMix	395.02	396.68
	IW	354.81	356.37
	PT	369.57	370.24
	IWPT	**346.85**	**347.78**
CV&Reddits	$\mathcal{T}$ w. αC	302.43	320.13
	$\mathcal{T}$ w. C	215.96	241.42
	$\mathcal{S} \cup \mathcal{T}$	275.85	302.64
	DSDSR	206.07	233.43
	DEMix	226.09	255.87
	IW	218.61	234.22
	PT	195.49	217.62
	IWPT	**180.98**	**203.14**

Table 1. Perplexity scores for baselines and different DA approaches. We set cohort size $C = 5,000$ and $\alpha = 0.1$.

4.3. Results

Table 1 summarizes the model performance of our algorithm and baseline approaches. First, we observe from results on both datasets that training models with a small cohort size αC in $\mathcal{T}$ only has the worst performance, which is because the DP noise dominates the model update in each iteration. Increasing the cohort size to C can greatly improve the utility for $\mathcal{T}$ only. However, according to the argument made in Section 3.1, we need to trade off a significant amount of training time for a larger C.

For the baseline trained with large population size in $\mathcal{S} \cup \mathcal{T}$ and large cohort size, simply treating source domain data as target domain data does not improve the performance much possibly because source domain data is from a different distribution and has larger volume which dominates the model update. DA specific architectures (DSDSR and DEMix) improved this baseline to some extent but can incur more communication cost due to larger model sizes.

On the other hand, both IW and PT approaches outperform the baseline methods, and are better than the DA specific architectures on SubReddits dataset. The combined IWPT approach achieves the best PPL, 13% and 30% lower than the baseline models on SubReddits and CV&Reddits, respectively.

5. Conclusion and Future Work

We demonstrate that the population size being small in PFL not only harms the model quality but also slows down the LM training. With our proposed domain adaptation algo-

rithm, which weights the source domain data appropriately, we show it is possible to have a larger population and train LMs with a better quality in a timely manner. Since instance weighting framework can be applied to other data domains than languages, extending the framework to other domains, e.g., images, is a direction for future work.

References

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pp. 308–318, 2016.

Anil, R., Ghazi, B., Gupta, V., Kumar, R., and Manurangsi, P. Large-scale differentially private BERT. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 6481–6491, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.findings-emnlp.484.

Ardila, R., Branson, M., Davis, K., Kohler, M., Meyer, J., Henretty, M., Morais, R., Saunders, L., Tyers, F., and Weber, G. Common Voice: A Massively-Multilingual Speech Corpus. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pp. 4218–4222, Marseille, France, May 2020. European Language Resources Association. ISBN 979-10-95546-34-4.

Bonawitz, K., Eichner, H., Grieskamp, W., Huba, D., Ingerman, A., Ivanov, V., Kiddon, C., Konečnỳ, J., Mazzocchi, S., McMahan, B., et al. Towards federated learning at scale: System design. *Proceedings of machine learning and systems*, 1:374–388, 2019.

Caldas, S., Duddu, S. M. K., Wu, P., Li, T., Konečný, J., McMahan, H. B., Smith, V., and Talwalkar, A. LEAF: A Benchmark for Federated Settings, December 2019.

Chronopoulou, A., Peters, M., and Dodge, J. Efficient Hierarchical Domain Adaptation for Pretrained Language Models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1336–1351, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.96.

Dwork, C. and Roth, A. The Algorithmic Foundations of Differential Privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, August 2014. ISSN 1551-305X. doi: 10.1561/0400000042.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. Calibrating Noise to Sensitivity in Private Data Analysis. In Halevi, S. and Rabin, T. (eds.), *Theory of Cryptography*, Lecture Notes in Computer Science, pp. 265–284, Berlin, Heidelberg, 2006. Springer. ISBN 978-3-540-32732-5. doi: 10.1007/11681878_14.

Ganesh, A., Haghifam, M., Nasr, M., Oh, S., Steinke, T., Thakkar, O., Thakurta, A., and Wang, L. Why is public pretraining necessary for private model training? *arXiv preprint arXiv:2302.09483*, 2023.

Ghosh, A., Roughgarden, T., and Sundararajan, M. Universally utility-maximizing privacy mechanisms. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pp. 351–360, 2009.

Gururangan, S., Marasović, A., Swayamdipta, S., Lo, K., Beltagy, I., Downey, D., and Smith, N. A. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8342–8360, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.740. URL https://aclanthology.org/2020.acl-main.740.

Gururangan, S., Lewis, M., Holtzman, A., Smith, N. A., and Zettlemoyer, L. Demix layers: Disentangling domains for modular language modeling. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 5557–5576, 2022.

Jiang, J. and Zhai, C. Instance weighting for domain adaptation in NLP. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pp. 264–271, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL https://aclanthology.org/P07-1034.

Lee, K., Lam, M., Pedarsani, R., Papailiopoulos, D., and Ramchandran, K. Speeding up distributed machine learning using codes. *IEEE Transactions on Information Theory*, 64(3):1514–1529, 2017.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pp. 1273–1282. PMLR, 2017.

McMahan, H. B., Ramage, D., Talwar, K., and Zhang, L. Learning differentially private recurrent language models. In *International Conference on Learning Representations*, 2018.

McMillan, A., Javidbakht, O., Talwar, K., Briggs, E., Chatzidakis, M., Chen, J., Duchi, J., Feldman, V., Goren, Y., Hesse, M., et al. Private federated statistics in an interactive setting. *arXiv preprint arXiv:2211.10082*, 2022.

McSherry, F. D. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, pp. 19–30, 2009.

Mironov, I. Rényi Differential Privacy. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, pp. 263–275, August 2017. doi: 10.1109/CSF.2017.11.

Mironov, I., Talwar, K., and Zhang, L. Rényi Differential Privacy of the Sampled Gaussian Mechanism, August 2019.

Moore, R. C. and Lewis, W. Intelligent Selection of Language Model Training Data. In *Proceedings of the ACL 2010 Conference Short Papers*, pp. 220–224, Uppsala, Sweden, July 2010. Association for Computational Linguistics.

Moriokal, T., Tawara, N., Ogawa, T., Ogawa, A., Iwata, T., and Kobayashi, T. Language Model Domain Adaptation Via Recurrent Neural Networks with Domain-Shared and Domain-Specific Representations. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6084–6088, April 2018. doi: 10.1109/ICASSP.2018.8462631.

Nguyen, J., Malik, K., Zhan, H., Yousefpour, A., Rabbat, M., Malek, M., and Huba, D. Federated learning with buffered asynchronous aggregation. In *International Conference on Artificial Intelligence and Statistics*, pp. 3581–3607. PMLR, 2022.

Parzen, E. *Modern probability theory and its applications*. Wiley, 1960.

Paulik, M., Seigel, M., Mason, H., Telaar, D., Kluivers, J., van Dalen, R., Lau, C. W., Carlson, L., Granqvist, F., Vandevelde, C., et al. Federated evaluation and tuning for on-device personalization: System design & applications. *arXiv preprint arXiv:2102.08503*, 2021.

Peng, X., Huang, Z., Zhu, Y., and Saenko, K. Federated Adversarial Domain Adaptation, December 2019.

Peterson, D., Kanani, P., and Marathe, V. J. Private Federated Learning with Domain Adaptation, December 2019.

Reddi, S. J., Charles, Z., Zaheer, M., Garrett, Z., Rush, K., Konečný, J., Kumar, S., and McMahan, H. B. Adaptive Federated Optimization. In *International Conference on Learning Representations*, March 2022.

Shen, Y., Du, J., Zhao, H., Zhang, B., Ji, Z., and Gao, M. FedMM: Saddle Point Optimization for Federated Adversarial Domain Adaptation, November 2021.

Tandon, R., Lei, Q., Dimakis, A. G., and Karampatziakis, N. Gradient coding: Avoiding stragglers in distributed learning. In *International Conference on Machine Learning*, pp. 3368–3376. PMLR, 2017.

Wang, Y.-X., Balle, B., and Kasiviswanathan, S. P. Subsampled rényi differential privacy and analytical moments accountant. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1226–1235. PMLR, 2019.

Yamada, M., Suzuki, T., Kanamori, T., Hachiya, H., and Sugiyama, M. Relative density-ratio estimation for robust distribution comparison. *Neural computation*, 25(5): 1324–1370, 2013.

Yao, C.-H., Gong, B., Qi, H., Cui, Y., Zhu, Y., and Yang, M.-H. Federated Multi-Target Domain Adaptation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1424–1433, 2022.

A. Dataset Preprocessing

The set of known vocabulary is built with target domain data in the training set by choosing top 10K frequent words and is assumed to be known in advance. Every word outside the vocabulary list is mapped as <UNK>. We append <BOS> to the beginning and <EOS> to the end of every sentence. Within each user, we limit the number of tokens (words) to 1,600 and cut the input sentences into sequences of length 10. When a sequence has length less than 10, we append <PAD> to make it have length 10.

B. Proof of Proposition 3.1

Here we restate and prove Proposition 3.1. Let $m = N - Np$ be the number of current unavailable devices, $k = C - Npq$ be the number of devices needed for current PFL iteration.

Proposition B.1. *Assume that the time for the i-th unavailable device becoming available and being sampled for training is $T_i \sim \text{Exponential}(\lambda)$. Let U_k be the random variable which describes the time when the first k devices become available and are sampled. Then*

$$\frac{1}{\lambda} \cdot \frac{C - Npq}{N(1-p)+1} \leq \mathbb{E}[U_k] \leq \frac{C}{\lambda(N-C)}.$$

Proof. We first state two properties about Exponential distribution:

1. The minimum of n exponential random variables is exponential: $\min\{T_1, \dots T_n\} \sim \text{Exponential}(n\lambda)$ (Parzen, 1960).
2. The exponential random variable T_i is a memoryless: $P(T_i > a + b | T_i > b) = P(T_i > a)$ (Parzen, 1960).

In our definition, $U_1 = \min\{T_1, \dots T_m\} \sim \text{Exponential}(m\lambda)$ from the first property.

WLOG, let $U_i = \min\{T_i, \dots, T_m\} | T_j > U_{i-1}$ where $j = \{i, \dots, m\}$ and $i > 1$, then with the second property we can derive:

$$\begin{aligned} P(U_i - U_{i-1} > a) &= P(U_i - U_{i-1} > a | U_i > U_{i-1}) \\ &= P(U_i > a + U_{i-1} | U_i > U_{i-1}) \\ &= P(U_i > a) \\ &= P(\min\{T_i, \dots, T_m\} > a). \end{aligned}$$

Thus, $U_i - U_{i-1} \sim \text{Exponential}((m-i+1)\lambda)$ from the first property.

Then we have:

$$\mathbb{E}[U_k] = \mathbb{E}[\sum_{i=2}^{k}(U_i - U_{i-1}) + U_1] = \sum_{i=2}^{k}\mathbb{E}[(U_i - U_{i-1})] + \mathbb{E}[U_1] = \frac{1}{\lambda}\sum_{x=m-k+1}^{m}\frac{1}{x}.$$

Since $\frac{1}{x}$ is convex, we know from the lower and upper Riemann sum that:

$$\int_{m-k+1}^{m+1}\frac{1}{x}dx \leq \sum_{x=m-k+1}^{m}\frac{1}{x} \leq \int_{m-k}^{m}\frac{1}{x}dx$$

Then the lower bound can be derived as follows:

$$\begin{aligned} \int_{m-k+1}^{m+1}\frac{1}{x}dx &= \ln\frac{m+1}{m-k+1} \\ &\geq 1 - \frac{m-k+1}{m+1} \\ &= \frac{k}{m+1} = \frac{C-Npq}{N(1-p)+1}, \end{aligned}$$

where the first inequality comes from the fact that $1 - \frac{1}{x} \leq \ln x \leq x - 1$.

Similarly, the upper bound can be derived as follows:

$$\begin{aligned}\int_{m-k}^{m} \frac{1}{x} dx &= \ln \frac{m}{m-k} \\ &\leq \frac{m}{m-k} - 1 \\ &= \frac{k}{m-k} \\ &= \frac{C - Npq}{N(1-p) - (C - Npq)} \\ &= \frac{C}{N\frac{(1-p)}{(1-\frac{Nq}{C}p)} - C} \\ &\leq \frac{C}{N-C},\end{aligned}$$

where the last inequality comes from the fact that server tends to oversample $q \geq \frac{C}{N}$. □