Can Public Large Language Models Help Private Cross-device Federated Learning?

Boxin Wang[3*], **Yibo Jacky Zhang**[4], **Yuan Cao**[2], **Bo Li**[3], **H. Brendan McMahan**[1], **Sewoong Oh**[1], **Zheng Xu**[1], **Manzil Zaheer**[2]
[1] Google Research, [2]Google Deepmind, [3]UIUC, [4]Stanford

Abstract

We study (differentially) private federated learning (FL) of language models. The language models in cross-device FL are relatively small, which can be trained with meaningful formal user-level differential privacy (DP) guarantees when massive parallelism in training is enabled by the participation of a moderate size of users. Recently, public data has been used to improve privacy-utility trade-offs for both large and small language models. In this work, we provide a systematic study of using large-scale public data and LLMs to help differentially private training of on-device FL models, and further improve the privacy-utility tradeoff by techniques of distillation. Moreover, we propose a novel distribution matching algorithm with theoretical grounding to sample public data close to private data distribution, which significantly improves the sample efficiency of (pre-)training on public data. The proposed method is efficient and effective for training private models by taking advantage of public data, especially for customized on-device architectures that do not have ready-to-use pre-trained models.

1 Introduction

Federated Learning (FL) (McMahan et al., 2017, 2018; Kairouz et al., 2019) is designed to collaboratively train a global model on decentralized data across user clients while protecting data privacy. FL emerged as an effective privacy-preserving solution of training (language) models, as rich text data are generated by users, which may contain sensitive and personal information. After McMahan et al. (2017) proposed to train on-device recurrent neural networks, FL has been widely used in various natural language processing applications and products, including next-word prediction (Hard et al., 2018), keyword spotting (Hard et al., 2020), and out-of-vocabulary word discovery (Chen et al., 2019).

To further protect user privacy, Differential Privacy (DP) (Dwork et al., 2006; Dwork, 2011; Dwork and Roth, 2014; McMahan et al., 2018) is introduced to provide formal privacy guarantees of models trained by federated learning. DP for deep learning explicitly adds random noise with bounded sensitivity to a training process (*e.g.*, DP-SGD (Abadi et al., 2016)), ensuring a quantifiable similarity in output model distributions when the training dataset changes. When combining DP with FL, a variant of DP-SGD called DP-FedAvg (McMahan et al., 2018)) is applied to guarantee user-level DP (Dwork, 2010). Current research primarily focuses on applying user-level DP to small on-device models with fewer than 10 million parameters (McMahan et al., 2018; Kairouz et al., 2021; Ramaswamy et al., 2020). The model size is limited due to challenges such as significant DP noise required to preserve privacy (Li et al., 2021) and the communication costs in cross-device FL.

Recent advances in large language models (LLMs) (Thoppilan et al., 2022; Radford et al., 2019; Brown et al., 2020; Devlin et al., 2019; Raffel et al., 2020) have revolutionized natural language processing (NLP) and achieved unprecedented performance on various tasks such as text generation, machine translation, and sentiment analysis. However, their success comes at a cost of requiring massive amounts of computational resources, making them difficult to deploy on resource-constrained devices such as smartphones, tablets, or other edge devices. Additionally, there are concerns regarding the user privacy in various aspects such as memorizing personal information in training, and exposing private query in inference.

Recent work explore incorporating public information to improve privacy-utility trade-off in applying DP for (large) LMs (Yu et al., 2022; Li et al.,

[*] Part of the work was done while Boxin Wang was an intern at Google. Correspondence to: Boxin Wang boxinw2@illinois.edu and Zheng Xu xuzheng@google.com.

Findings of the Association for Computational Linguistics: NAACL 2024, pages 934–949
June 16-21, 2024 ©2024 Association for Computational Linguistics

2021). Public data (Amid et al., 2021) or other side information (Li et al., 2022) are also studied for (DP) FL. In non-DP FL settings, Nguyen et al. (2022) studies the effect of initializing from a pre-trained model. However, it is an open question on *how to leverage the power of pre-trained LLMs to facilitate private FL for on-device LMs*.

In this work, we answer the question through systematic study aimed at enhancing private federated learning for on-device LMs with public pre-trained LMs. Specifically, Our approach involves leveraging both public data and pre-trained LLMs to improve differentially private federated learning for on-device models by techniques of public pre-training and distillation. Additionally, we propose a novel distribution matching algorithm, which is backed by theoretical analysis, to sample public data closely resembling the private data distribution, which significantly increases sample efficiency in public training. Moreover, our extensive empirical results align with our theoretical predictions, further substantiating our approach. Our work complements existing research by utilizing LLMs to improve public training through knowledge distillation for private cross-device federated learning, and achieve a strong privacy-utility trade-off with substantial improvements on sampling efficiency for public data. Our method points to a novel direction of efficiently enhancing private FL with public pretraining data and LLMs.

We summarize our **contributions** as follows:

- We focus on improving private federated learning for language modeling tasks and explore ways to leverage public data and pre-trained LLMs for tokenizers, training protocols, and data (sub)sampling.

- We conduct comprehensive studies and compare the use of Sentence Piece tokenizers from public LLM and unigram tokenizers from private corpus. We find that adopting public tokenizers from LLMs can not only prevent the potential privacy leakage from the private tokenizer vocabulary, but also lead to better learning utility with DP guarantees.

- For training protocol, we propose to leverage public LLM to teach private on-device LMs by knowledge distillation. We demonstrate that distilling public LLM to pre-train on-device LM can lead to more than 7% accuracy improvement with tight privacy bound ($\varepsilon = 1.77$). Moreover, it can achieve high data efficiency of using only 1% of the public data compared to that in public pre-training without LLM, and attain better accuracy.

- We further propose a novel distribution matching method that leverages both private on-device LMs and public LLMs to select public records close to private data distribution. We show that using 0.08% of carefully sampled public data to train on-device LM can lead to comparable performance as public pre-training on-device LMs with the whole pre-training corpus. Moreover, it reduces the public training time from more than one week to a few hours. Our method is grounded in theoretical analysis, which is corroborated by our extensive empirical results.

2 Differentially Private Federated Learning for On-device LMs

In this section, we walk through the preliminaries of differentially private federated learning of language models following the cross-device federated learning literature (McMahan et al., 2018; Kairouz et al., 2019, 2021). We also introduce the experimental setup used throughout this paper.

Cross-device Federated Learning. McMahan et al. (2017) introduce federated learning to collaboratively train LMs for next-word prediction from decentralized user data on a large number of mobile devices without directly sharing the private data. A common training algorithm of federated learning is `FedAvg` (McMahan et al., 2017), where each client downloads the current model from the centralized server, computes an update by performing local computation on their dataset (*e.g.*, running SGD) and sends the update back to the server. The server aggregates the updates across clients to update the global model and send the updated model back to local clients to achieve the goal of collaborative learning without directly accessing the training data on each user's mobile device.

In our experiments, we follow previous work (Kairouz et al., 2021; Amid et al., 2021; Wu et al., 2022) and sample 100 clients in each training round. Each client uses a batch size of 16 for local training. We set the training rounds $T = 1600$ in total.

User-level Differential Privacy. To further protect user privacy, Differential Privacy (DP) (Dwork et al., 2006; Dwork, 2011; Dwork and Roth, 2014) was introduced to provide a formal privacy guarantee for federated learning.

Definition 2.1 ((ε, δ)-Differential Privacy)**.** A randomized algorithm $\mathcal{M}$ with domain $\mathbb{N}^{|\mathcal{X}|}$ is (ε, δ)-differentially private if for all $\mathcal{S} \subseteq \mathrm{Range}(\mathcal{M})$ and for any adjacent datasets D and D':

$$\Pr[\mathcal{M}(D) \in \mathcal{S}] \leq \exp(\varepsilon) \Pr[\mathcal{M}(D') \in \mathcal{S}] + \delta.$$

Definition 2.1 provides a formal definition of (ε, δ)-DP by bounding the change in output distribution caused by a small input difference (or, adjacent datasets) for a randomized algorithm. In the FL setting, it is preferable to bound the output distribution caused by different users in order to protect the privacy of each client's whole dataset. Specifically, adjacent datasets of D and D' for user-level differential privacy (Dwork, 2010) are defined as: D can be obtained from D' by adding or subtracting all the records of a single user/client, which determines the unit of privacy guarantees.

In our experiments, we use DP-FTRL (Kairouz et al., 2021) for privacy accounting and private federated training, which can achieve strong privacy guarantee in practical FL scenarios (Xu et al., 2023). We use $\delta = 10^{-6}$ and consider two ε bounds: a tight privacy bound with $\varepsilon = 1.77$ by using a large noise multiplier $m = 8.83$, and a slightly loose privacy bound with $\varepsilon = 18.71$ and noise multiplier $m = 1.13$. We present more hyperparameter tuning details in Appendix §C.

On-device LMs. Due to the limited memory constraints of mobile devices, on-device LMs are relatively small (usually less than 10M parameters). In our work, we focus on two types of on-device auto-regressive LMs: LSTM (Hochreiter and Schmidhuber, 1997) and transformers (Vaswani et al., 2017). More model details can be found in Appendix §B.2.

Pre-trained LLMs. In addition to the on-device LMs trained on private datasets, this work also assumes that we have access to LLMs pre-trained on a large public corpus to aid private learning. Specifically, we use LaMDA (Thoppilan et al., 2022) 2B throughout this work as an example, and conduct a systematic study of leveraging LLMs to help private training of on-device LMs.

Datasets. We focus on next word prediction task on the StackOverflow benchmark dataset (2019) for private federated learning. Since StackOverflow is naturally keyed by users, each client in FL is a user in the Stack Overflow online forum. The examples of a client are sentences of questions and answers posted by a specific user. We follow (Reddi et al., 2021; Kairouz et al., 2021) to construct a validation set of 10K samples, and a test set of 16.5M samples. Our evaluation metric is in-vocabulary next word (token) prediction accuracy, which is computed as the ratio of accurately predicted in-vocabulary words to the total number of words in the sequence (excluding OOV tokens).

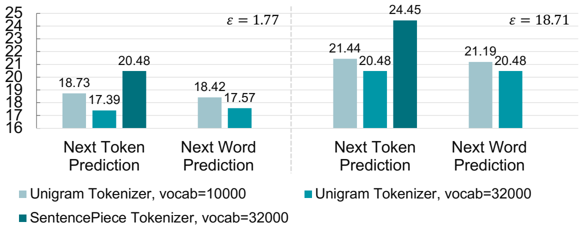


Figure 1: Next word (token) prediction accuracy for on-device LSTM with different tokenizers in the private FL.

In addition to StackOverflow as the (private) dataset, we use the realnews variant `c4/realnewslike` of C4 dataset (Raffel et al., 2020), as the public dataset. We analyzed the sources of the public C4 dataset and the Stackoverflow dataset for private training, and verified that there is no explicit overlap between public C4 dataset and the private StackOverflow dataset. More details can be found in Appendix §B.1.

3 Inspiration from LLMs

The success of publicly pre-trained LLMs motivate us to have retrospective views on further improving private on-device LMs. In this section, we explore inpiration from LLMs: the use of subword tokenizers and a large public corpus for pre-training. We apply them to on-device LMs, and observe that both techniques bring significant performance improvement for private FL.

3.1 Using Public Tokenizer from LLMs

Tokenizer is an important module of LMs, which transforms natural languages into a sequence of predefined symbol sets (vocabulary). Prior work in the literature of private FL of LMs (McMahan et al., 2018; Kairouz et al., 2021; Amid et al., 2021) use word-level unigram tokenizers potentially directly built from user data, which may need additional privacy budget (Ponomareva et al., 2022; Bagdasaryan et al., 2022).

Recent LLMs adopt sub-word tokenizers (Kudo and Richardson, 2018; Sennrich et al., 2016; Schuster and Nakajima, 2012), which mitigate most out-of-vocabulary (OOV) problems and yield state-of-the-art performance across different downstream tasks. This motivate us to replace the prior word-level unigram tokenizers with public sub-word tokenizers. Specifically, we use SentencePiece tokenizer (Kudo and Richardson, 2018) from LaMDA.

To conduct comparison between unigram tokenizers and subword tokenizers for next word (token) prediction task, we convert the next word prediction accuracy into next token prediction accuracy. This conversion is achieved through splitting each word using the SentencePiece tokenizer. We consider all tokens within a word as accurate if the predicted word is correct. We compare standard SentencePiece models (vocabulary size = $32K$) with unigram tokenizers that selects the top-k frequent words from user data with $k = 10K$ or $32K$ as vocabulary.

We present the private FL accuracy on the StackOverflow dataset in Figure 1. For the unigram tokenizer, using a larger vocabulary size in the DP setting can result in a slight performance drop, which can be different from the observation in non-DP settings (Charles et al., 2022; Xu et al., 2022a). It is possible that the parameter increase of the embedding layer enlarges the effect of DP noise and hurts the final accuracy. However, for next token prediction accuracy, although the public SentencePiece tokenizer from LaMDA also consists of $32K$ tokens, it can significantly improve the private FL accuracy upon the unigram tokenizers, especially with smaller DP noise and $\varepsilon = 18.71$. We also observe that SentencePiece tokenizer finds no OOV tokens in the StackOverflow dataset, thus yielding the same high prediction accuracy with or without the OOV token. Therefore, we use SentencePiece tokenizer in the rest of this paper.

3.2 Publicly pre-training for On-device LMs

In addition to the use of subword tokenizers, LLMs benefit from pre-training on a large public corpus (Li et al., 2022; Yu et al., 2022). In this section, we explore pre-training on-device LMs on public corpus to improve private federated learning.

Pre-training Details. We use the standard autoregressive language modeling loss $\mathcal{L}_{LM}$ to pre-train on-device LMs on the public C4 dataset, which takes around $1,400K$ steps (over a week of single GPU time) to process the entire dataset with the batch size of 512. We then use the publicly pre-trained checkpoint as the start point for private federated learning. We leave more details in §B.2.

Results. We present the next token prediction accuracy on the private StackOverflow dev set in Table 1. We observe that the accuracy on the private dataset significantly improves after pre-training for different different privacy budgets, shedding light on an effective way to boost private FL performance. We also observe that after pre-training, it gives reasonable zero-shot accuracy on the private dataset even without private training (round=0).

	w/o pre-training		w/ pre-training	
Rounds	0	1600	0	1600
$\varepsilon = 1.77$	0.00	20.48	16.94	27.27
$\varepsilon = 18.71$		24.45		30.13

Table 1: Next Token Prediction Accuracy on the private StackOverflow dev set with or without public pre-training.

4 Distillation from Public LLM

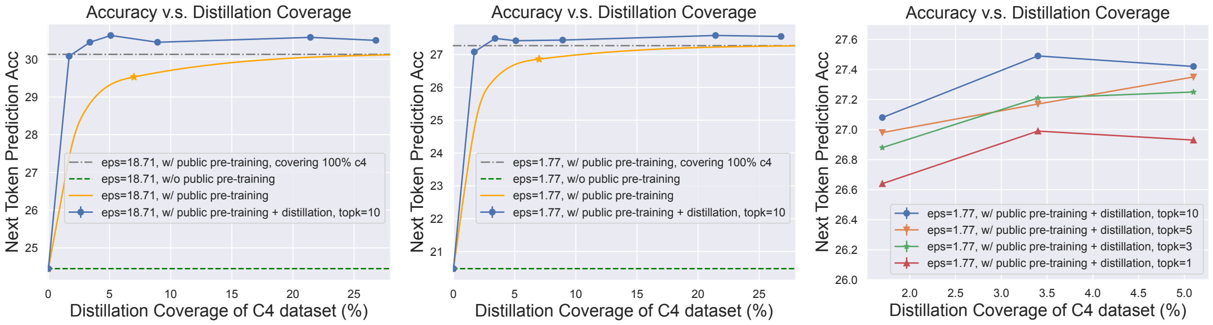
On one hand, the cost of public pre-training for on-device LMs is still expensive on a large public corpus (around a week of GPU time). On the other hand, existing LLMs are well pre-trained and demonstrate promising performance across a variety of downstream tasks. This motivates us to explore on whether we can leverage existing LLMs to improve the sample efficiency of pre-training on-device LMs. In this section, we answer the question above with systematic studies and show that we can improve the sample efficiency by using only 1% of pre-training data and distillation from LLMs, achieving similar or even better performance than using 100% of pretrianing data without distillation.

4.1 Distillation Design

Inspired by the literature of model compression (Sun et al., 2020; Jiao et al., 2019), we use knowledge distillation to transfer the knowledge from trained LLMs into on-device LMs during pre-training. The distillation pipeline contains the following two steps:

Building a distillation corpus. Given an input sequence from the public pre-training corpus, the LLM outputs the probability distribution over the vocabulary for next token prediction at each decoding step. To construct a distillation corpus, we save the top-k logits with k nonzero entries $\boldsymbol{z}_T$ from the teacher LLM as a silver-label dataset. In this way, the distillation corpus is model-agnostic, and thus can be applied to different variants of on-device LMs for pre-training. Moreover, selecting a reasonable top-k for the logits can both help compress the distillation corpus to a moderate size and filter out noisy signals from tokens with low output probabilities.

Public pre-training with distillation loss. Since we align the tokenizer of the on-device LM with the LLM to share the same vocabulary, we can



(a) Acc. v.s. distillation steps ($\varepsilon = 18.71$) (b) Acc. v.s. distillation steps ($\varepsilon = 1.77$) (c) Acc. v.s. top-k logits ($\varepsilon = 1.77$)

Figure 2: Ablation studies on how distillation steps and top-k logits in distillation impact next token prediction accuracy (Acc.) of on-device LSTM models on the dev set of the private StackOverflow dataset.

align the output distribution of on-device LMs and LLMs by the cross-entropy loss. Formally, for next token prediction task, given the output logits from student on-device LMs $\boldsymbol{z}_S$, the gold label from the pre-training corpus $\boldsymbol{y}$, and the logits from the distillation corpus of LLMs $\boldsymbol{z}_T$, we add an additional knowledge distillation loss $\mathcal{L}_{KD} = \mathrm{CE}(\boldsymbol{z}_S/t, \boldsymbol{z}_T/t)$ to the pre-training language modeling loss $\mathcal{L}_{LM} = \mathrm{CE}(\boldsymbol{z}_S, \boldsymbol{y})$ as our public pre-training loss $\mathcal{L}_{\text{pub}} = \mathcal{L}_{LM} + \beta\mathcal{L}_{KD}$ where t is the temperature. More distillation details are in §B.3.

4.2 Experimental Results

After public pre-training with knowledge distillation, We use the checkpoints at different pre-training steps as the start point for private federated learning. Our main results can be found in Table 2. We show that by using 1% C4 dataset for pre-training with knowlegde distillation, we can significantly improve the sample efficiency without hurting but even improving the private FL accuracy for both LSTM and transformers, when compared with public pre-training on the whole C4 dataset. The sample efficiency improvement thus reduces the pre-training cost from one week to around one day, shedding light on a promising direction to improve the efficiency and utility of private FL.

Ablation studies on distillation steps. To understand whether distillation for more epochs can help with private FL, we conduct a set of ablation studies on distillation steps given different privacy budgets as shown in Figure 2b and 2a. Specifically, we use the checkpoints at different distillation steps to initialize on-device LSTM and report the next word prediction accuracy after private FL at round 1600. We observe a consistent performance improvement when the distillation covers less than 5% of the C4 dataset. But when we pre-train the LM for more epochs, the improvement becomes marginal. This suggests that teaching on-device LMs via LLMs can converge quickly within a few iterations.

Abaltion studies on top-k logits. We take the top-k logits of the LLM to construct our distillation datasets and pre-train the on-device LMs. Here, we conduct an ablation study by pre-training different on-device LMs with different k and evaluate how top-k logits in distillation can impact the accuracy of private FL. We present our empirical results in Figure 2c and Appendix Figure 4. We observe that pre-training with a larger k is more helpful to achieve better downstream accuracy on private data. To have a reasonable trade-off between dataset size and pre-training performance, we use top-$k = 10$ in all the following experiments.

5 Distribution Matching

In the previous section, we achieve compelling performance by employing LLM distillation using only 1% of the *randomly sampled* pre-training corpus. Now we further investigate the possibility of improving sample efficiency by selectively identifying public samples that align with the distribution of private samples. To this end, we propose a novel distribution matching method to sample public records for pre-training with a novel theoretical analysis jointly considering public-private distribution shift and DP mechanism. We demonstrate that by carefully selected 0.08% of public samples, we can pre-train on-device LMs that perform as well as using 1% of public samples with distillation. This approach significantly improves sample efficiency, providing an additional knob of using public pre-training for private on-device models.

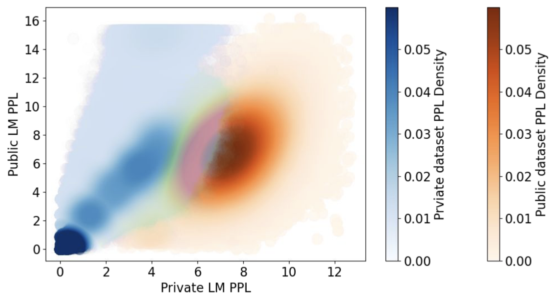


Figure 3: Visualization of PPL distribution of the private and public datasets evaluated by the private on-device LM and the public LLM. The private dataset exhibits a concentration of low PPL values, whereas the public corpus is dispersed across a broader range of PPL values, with a higher average PPL.

5.1 Algorithm

We hypothesize two principles to sample public records to match the private distribution: (i) the probability of the public sample x on the private data distribution $p_{\text{priv}}(x)$ is high, which can be approximated by the prediction of the on-device LMs trained on the private dataset; (ii) the probability of a public sample x on the public data distribution $p_{\text{pub}}(x)$ is also high, as we expect those samples are easy-to-learn (Swayamdipta et al., 2020) and of high data quality in the public corpus. The probability $p_{\text{pub}}(x)$ can be approximated by the public pre-trained LLMs.

To verify our hypothesis, we visualize the perplexity (PPL) distribution of public samples and private samples evaluated by both a privately fine-tuned on-device LM and a public pre-trained LLM in Figure 3. To have an "oracle" on-device LM that well captures the private data distribution, we fine-tune it on the private data without DP noise to overfit the private data distribution. We randomly sample 10k records from the public dataset and private dataset, respectively. We observe that the private dataset mostly concentrates on the regime with low PPL evaluated by the public and private LMs, whereas the public dataset is more diverse and distributed across a broader range of PPL values. The distribution visualization confirms our hypothesis to select public samples from the lower left corner, which correspond to samples with high probabilities $p_{\text{pub}}(x)$ and $p_{\text{priv}}(x)$ on public and private data distribution (*i.e.*, low perplexity evaluated by public and pirvate LMs).

In practice, we do not have an "oracle" on-device LM trained on private data for distribution match. Instead, we propose to fine-tune an on-device LM with DP for certain rounds $T' < T$ before consuming all the privacy budgets, and then use the checkpoint at round T' with DP guarantee to approximate $p_{\text{priv}}(x)$ and perform distribution matching to sample public records. This post-processing based on a DP checkpoint will not incur any additional privacy cost. Thereafter, we can use the sampled *public* records to further train the private checkpoint at round T', as a way for efficient public (pre-)training. Following the strategy in §4, we also employ the distillation loss to better train the on-device LM with carefully sampled public records to further enhance the sample efficiency. Lastly, we use the remaining privacy budgets to fine-tune the on-device LM until reaching round T, and evaluate its next token prediction accuracy at the dev and test sets. We term the paradigm of two-stage private learning combined with public training as "public mid-training". This approach differs from "public pre-training", which involves public pre-training prior to private FL. We present the distribution matching protocol in Algorithm 1.

Algorithm 1 Leveraging LLMs for distribution matching and public training in private federated learning.

Input: Public pre-training corpus D, private corpus D^*, sampling rate q, private fine-tuning rounds T, first-stage fine-tuning rounds $T' < T$ for distribution matching, a public pre-trained LLM
Output: Private on-device LM with DP guarantee

```
Randomly initialize an on-device LM;
// ① First-stage private federated learning
Use DP-FTRL to train the on-device LM for rounds $T'$;
for each $x \in D$ do
    // ② Probability evaluation
    Compute the average (token) log prob $\log p_{\text{priv}}(x)$ given the privately fine-tuned LM at round $T'$;
    Compute the average (token) log prob $\log p_{\text{pub}}(x)$ given a publicly pre-trained LLM ;
end for
// ③ Distribtion matching
Sort $D$ based on $\log p_{\text{priv}}(x) + \log p_{\text{pub}}(x)$
Sample a subset of $D$ as $D'$ with top $\log p_{\text{priv}}(x) + \log p_{\text{pub}}(x)$ values, such that $|D'| = q|D|$.
// ④ Public mid-training with LLM distillation
Train the on-device LM with the loss $\mathcal{L}_{\text{pub}}$ on $D'$
// ⑤ Second-stage private federated learning
Use DP-FTRL to train the on-device LM for the remaining rounds of $T - T'$
return On-device LM with DP guarantee
```

5.2 Theoretical Analysis

In this section, we provide the theoretical analysis of our distribution matching protocol to present the *intuition* behind our selection hypothesis. In essence, the goal of our distribution matching algorithm is to have a good estimator for the private distribution. However, characterizing the distribution shift in the context of differential privacy is a challenging problem, in that the private models are trained with DP noise, which can yield an inac-

	q (% of	LLM	Distribution	Accuracy (LSTM)		Accuracy (Transformer)	
	Public Data)	Distillation	Matching	ε=1.77	ε=18.71	ε=1.77	ε=18.71
No Public Training	0%			20.68$_{\pm 0.04}$	28.87$_{\pm 0.04}$	23.98$_{\pm 0.15}$	28.29$_{\pm 0.06}$
Pre-training w/ public data ($T' = 0$)	100%			28.01$_{\pm 0.26}$	30.70$_{\pm 0.01}$	**28.05**$_{\pm 0.02}$	30.10$_{\pm 0.00}$
· **LLM Distillation (100k steps)**	1%	✓		**28.68**$_{\pm 0.09}$	**31.13**$_{\pm 0.03}$	27.75$_{\pm 0.06}$	**30.19**$_{\pm 0.01}$
· **LLM Distillation (8k steps)**	0.08%	✓		26.18$_{\pm 0.04}$	29.53$_{\pm 0.10}$	25.31$_{\pm 0.08}$	29.36$_{\pm 0.12}$
Mid-training w/ public data ($T' = T/2$)	0.08%			26.67$_{\pm 0.06}$	29.76$_{\pm 0.03}$	25.83$_{\pm 0.03}$	29.15$_{\pm 0.01}$
· **LLM Distillation (8k steps)**	0.08%	✓		27.01$_{\pm 0.03}$	30.18$_{\pm 0.06}$	26.04$_{\pm 0.12}$	29.47$_{\pm 0.05}$
+ Distribution Matching	0.08%	✓	✓	**28.01**$_{\pm 0.08}$	**30.63**$_{\pm 0.02}$	**27.17**$_{\pm 0.03}$	**29.83**$_{\pm 0.01}$

Table 2: Summary of techniques to improve downstream stream next token **prediction accuracy** and **sample efficiency** for on-device LSTM and transformer model evaluated on the StackOverflow test set.

curate estimation of private data distribution, and thus add the complexity to our analysis.

Problem Setup. Define the text data domain as $\mathcal{X}$. Denote $\ell_{\text{pub}} : \mathcal{X} \to \mathbb{R}$ as the log-density function of the public data distribution (i.e., $\ell_{\text{pub}}(x) = \log p_{\text{pub}}(x)$ where $p_{\text{pub}}(x)$ is the public data density estimated by public LLMs), and ℓ_{priv} as the *accurate* log-density function of the private data distribution (i.e., $\ell_{\text{priv}}(x) = \log p_{\text{true priv}}(x)$ where $p_{\text{true priv}}(x)$ is the true private data density). However, due to limited private data sampled from the true private data distribution and DP noise injected in the private FL, we can only obtain an inaccurate estimation $\hat{\ell}_{\text{priv}} = \log p_{\text{priv}}(x)$ of the true private log-density ℓ_{priv}, where $p_{\text{priv}}(x)$ is the private data density estimated by private on-device LMs. Note that we use the hat notation $\hat{\ell}_{\text{priv}}$ to denote that it is an estimation of the true private log-density ℓ_{priv}.

We can view the estimation $\hat{\ell}_{\text{priv}}$ is a random variable where the randomness comes from: (i) that the private dataset we have is sampled from the private data distribution; and (ii) the randomness in the algorithm of obtaining $\hat{\ell}_{\text{priv}}$ based on the private dataset, e.g., differential privacy. Following previous work (Jiang et al., 2023), we make a standard assumption. We assume the estimated private data log-density function is an unbiased estimator, i.e., $\mathbb{E}[\hat{\ell}_{\text{priv}}] = \ell_{\text{priv}}$. Since ℓ_{pub} may not be ideal because of public-private domain shift, and $\hat{\ell}_{\text{priv}}$ may mot be ideal because of its DP noise, ℓ_{pub} and $\hat{\ell}_{\text{priv}}$ are neither good estimators for ℓ_{priv}. *Can we leverage both of the information and form a function $\hat{h} : \mathcal{X} \to \mathbb{R}$ that combines ℓ_{pub} and $\hat{\ell}_{priv}$ such that $\hat{h}$ is a good estimator for ℓ_{priv}?* In the following analysis, we choose $\hat{h} = \frac{1}{2}\ell_{\text{pub}} + \frac{1}{2}\hat{\ell}_{\text{priv}}$ and analyze when and why it can be a better estimator to the true private log-density ℓ_{priv} than ℓ_{pub} and $\hat{\ell}_{\text{priv}}$.

We need some mathematical tools to define what does it mean to be "better". Concretely, we need a metric to measure the distance between functions. This can be done by having an inner product $\langle \cdot, \cdot \rangle$ in the function space of $\mathcal{H} = \{f : \mathcal{X} \to \mathbb{R}\}$, and hence the norm in the function space $\mathcal{H}$ is $\|f\| = \sqrt{\langle f, f \rangle}$ for $\forall f \in \mathcal{H}$. Our analysis holds with *any* choice of the inner product as long as it does not make the log-densities norm infinite. We discuss a concrete choice of the inner product and its relation to the KL divergence in Appendix §D.

With the norm as a "ruler", we are able to define the following key quantities that formally characterize the setting.

1. **Public-Private Domain Distance.** Let $d_{\text{pub, priv}} = \|\ell_{\text{pub}} - \ell_{\text{priv}}\|$ denote the distance between the public data log-density ℓ_{pub} and the true private log-density ℓ_{priv}.
2. **Private Domain Randomness.** Let $\sigma^2_{\text{priv}} = \mathbb{E}[\|\hat{\ell}_{\text{priv}} - \ell_{\text{priv}}\|^2]$ denote the randomness of the estimated private log-density, i.e., the quality of the estimated private log-density $\hat{\ell}_{\text{priv}}$

The above definitions are important because the quality of a private log-density estimator would depend on the public-private domain shift and the private domain randomness as we show next.

Theorem 5.1. Let $\epsilon(\hat{f}) = \mathbb{E}[\|\hat{f} - \ell_{\text{priv}}\|^2]$ characterise how good $\hat{f}$ is as an estimator of the true private data log-density ℓ_{priv} for any random function $\hat{f} \in \mathcal{H}$. Consider the following three quantities:

1. $\epsilon(\ell_{\text{pub}})$ characterizing the error of the public log-density function ℓ_{pub} to approximate ℓ_{priv}
2. $\epsilon(\hat{\ell}_{\text{priv}})$ depicting the error of the noisy private log-density function $\hat{\ell}_{\text{priv}}$ to approximate ℓ_{priv}
3. $\epsilon(\hat{h})$ characterizing the error of $\hat{h} = \frac{1}{2}\ell_{\text{pub}} + \frac{1}{2}\hat{\ell}_{\text{priv}}$ to approximate ℓ_{priv}.

Then,

$$\epsilon(\ell_{\text{pub}}) = d^2_{\text{pub, priv}} \tag{1}$$

$$\epsilon(\hat{\ell}_{\text{priv}}) = \sigma^2_{\text{priv}} \tag{2}$$

$$\epsilon(\hat{h}) = \frac{1}{4} d^2_{\text{pub, priv}} + \frac{1}{4}\sigma^2_{\text{priv}} \tag{3}$$

Interpretation Theorem 5.1 implies that:

- $\epsilon(\hat{h}) \leq \frac{1}{2}\max\{\epsilon(\ell_{\text{pub}}), \epsilon(\hat{\ell}_{\text{priv}})\}$.

	LSTM		Transformer	
	ε=1.77	ε=18.71	ε=1.77	ε=18.71
w/ $p_{\text{pub}}(x)$	**28.01**$\pm$0.08	**30.63**$\pm$0.02	**27.17**$\pm$0.03	29.83$\pm$0.01
w/o $p_{\text{pub}}(x)$	27.77$\pm$0.05	30.56$\pm$0.06	26.70$\pm$0.04	**30.18**$\pm$0.05

Table 3: Ablation studies on the use of public LLM for distribution matching evaluated on the StackOverflow test set.

- $\epsilon(\hat{h}) \leq \min\{\epsilon(\ell_{\text{pub}}), \epsilon(\hat{\ell}_{\text{priv}})\}$ if $\frac{1}{3} \leq \frac{d^2_{\text{pub, priv}}}{\sigma^2_{\text{priv}}} \leq 3$.

Combining the above, we have the following conclusion: recall $\hat{h} = \frac{1}{2}\ell_{\text{pub}} + \frac{1}{2}\hat{\ell}_{\text{priv}} = \frac{1}{2}\log(p_{\text{pub}}(x)p_{\text{priv}}(x))$. We can expect that $\hat{h}$ is better than either ℓ_{pub} or $\hat{\ell}_{\text{priv}}$ for any settings. Moreover, we can expect $\hat{h}$ to be better than both ℓ_{pub} and $\hat{\ell}_{\text{priv}}$ if (i) there is a domain shift between the public-private domain; and (ii) our estimated private log-density $\hat{\ell}_{\text{priv}}$ is noisy in an extent comparable to the domain shift. We leave the full proof and additional discussion in Appendix D.

5.3 Experimental Results

Experimental Setup. We set $T' = T/2 = 800$ rounds for the first-stage private federated learning. We use $q = 0.08\%$ of the whole pre-training corpus for public training, which reduces the public training time from more than 1 weeks to a few hours with a single GPU. For the public mid-training setting, we also evaluate how LLM distillation and distribution matching can impact the private FL accuracy, respectively. We run all the experimental settings for three times and report the average and standard deviation of test accuracy on the private StackOverflow dataset.

We present the results of on-device LSTM and transformers in Table 2. In the pre-training setting ($T' = 0$), we show that we cannot further improve the sample efficiency from 1% to 0.08% with LLM distillation improves the sample efficiency, as the final accuracy after private FL significantly decreases. In comparison, in the mid-training setting ($T' = T/2$), using LLM distillation on the 0.08% of randomly sampled pre-training corpus already gives better performance than pre-training. Moreover, with distribution matching to carefully sample public data, we further improve the private FL accuracy, attaining comparable performance to the setting using the whole public corpus for pre-training.

Ablation studies on $p_{\textbf{pub}}(x)$. Our distribution matching algorithm leverages both on-device LM and LLM to sample data close to the private distribution. To understand how the use of LLM ($p_{\text{pub}}(x)$) impact the sampling quality, we conduct an ablation study to sample a subset of D' based on top $\log p_{\text{priv}}(x)$ values alone instead of $\log p_{\text{priv}}(x) + \log p_{\text{pub}}(x)$. We use the p_{priv}-sampled D' for public mid-training and report the test accuracy of three runs for both on-device LSTM and transformers given different privacy budgets in Table 3. The experimental findings corroborate our theoretical analysis. Specifically, when on-device language models (LMs) are trained with high noise levels ($\varepsilon = 1.77$), we find that a combined utilization of both on-device LMs and LLMs consistently yields superior performance. This is because the estimated private log-density $\hat{\ell}_{\text{priv}}$ is noisy to a degree comparable to the domain shift, making $\hat{h}$ a more reliable estimator than $\hat{\ell}_{\text{priv}}$. Conversely, when on-device LMs are trained with low noise ($\varepsilon = 18.71$), the performance difference between models with and without p_{pub} is negligible. This indicates that the noise introduced by differentially private (DP) training is not as significant as the distribution shift, allowing $\hat{\ell}_{\text{priv}}$ to serve as a good estimator.

T'	0	400	800	1200	1600
ε=1.77	25.41	27.08	**27.73**	26.40	18.40
ε=18.71	28.38	30.07	**30.37**	29.45	19.34

Table 4: Ablation studies on the timing (T') of distribution matching for mid-point public training on on-device LSTM evaluated the StackOverflow dev set.

Ablation studies on T'. T' separates two-stage private federated learning and determines the timing for distribution matching and public training. In this ablation study, we evaluate the dev set accuracy of on-device LSTM given different T' and privacy budgets, as shown in Table 4 and Appendix Table 5. From the table, we can see that the on-device LSTM achieves the best private FL accuracy given $T' = T/2 = 800$. We think the reasons are as follows: when $T' = 0$, we cannot perform distribution matching as the on-device LM is not trained on the private dataset yet, and thus we can only use the randomly sampled data for pre-training; when $T' = 400$, the on-device LM could not be well trained on the private data distribution, thus yielding worse distribution matching quality; when $T' = 1200$ and $T' = 1600$, the private on-device LM is biased towards the public data distribution due to public training, thus giving worse private FL accuracy. As a result, we use $T' = 800$ in our main experiments, as it balances the private federated training and public training to have satisfactory distribution matching capabilities without biasing too much towards the public data distribution.

6 Conclusion

In this work, we propose to improve private federated learning by using LLMs in public training. We leverage LLMs to aid public training of on-device LMs via distribution matching to sample public data close to private data distribution, which further improves the effectiveness and efficiency of public training, demonstrating strong private learning accuracy while minimizing the need for large amounts of public training data. Our work sheds light on a promising direction to improve private federated learning with public LLMs.

Acknowledgement

We gratefully thank the anonymous reviewers and meta-reviewers for their constructive feedback. The authors thank the early feedback from Yanxiang Zhang. BL acknowledges support from the National Science Foundation under grant No. 2046726, No. 2229876, DARPA GARD, the National Aeronautics and Space Administration (NASA) under grant No. 80NSSC20M0229, and Alfred P. Sloan Fellowship.

Limitations

This work has paved the way for enhancing the utility of differentially private on-device FL models, using large-scale public data and LLMs, but we also acknowledge the following limitations:

- **Data Distribution Matching**: The proposed distribution matching algorithm aims to sample public data close to the private data distribution. The choice of $\hat{h}$ can be data dependent and a weighted combination of ℓ_{pub} and $\hat{\ell}_{\text{priv}}$, *i.e.*, $\hat{h} = (1-\beta)\ell_{\text{pub}} + \beta\hat{\ell}_{\text{priv}}$ where $\beta \in [0,1]$, as mentioned in Appendix §D.3. In practice, the optimal β can be an important hyper-parameter to tune the distribution matching algorithm. Our work mainly leverages $\hat{h} = \frac{1}{2}\ell_{\text{pub}} + \frac{1}{2}\hat{\ell}_{\text{priv}}$ to analyze when and why a better estimator to the true private log-density ℓ_{priv} than ℓ_{pub} and $\hat{\ell}_{\text{priv}}$. We leave it as important future direction to get the optimal β theoretically and empirically.
- **Computational Resources**: The use of large-scale public data and LLMs can improve the privacy-utility trade-off in DP FL models, but this often comes at the cost of computational resources. Our work mainly focuses on LaMDA 2B as an example of LLM due to the lack of computational resources. While our main focus does not lie in the knowledge distillation, we leave it as future work to extend the size of LLMs in public pre-training.

References

Martín Abadi, Andy Chu, Ian J. Goodfellow, H. B. McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. *CCS*.

E. Amid, Arun Ganesh, Rajiv Mathews, Swaroop Indra Ramaswamy, Shuang Song, T. Steinke, V. Suriyakumar, Om Thakkar, and Abhradeep Thakurta. 2021. Public data-assisted mirror descent for private model training. *International Conference On Machine Learning*.

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

The TensorFlow Federated Authors. 2019. Tensorflow federated stack overflow dataset.

Eugene Bagdasaryan, Congzheng Song, Rogier van Dalen, Matt Seigel, and Áine Cahill. 2022. Training a tokenizer for free with private federated learning. *arXiv preprint arXiv: Arxiv-2203.09943*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Zhiqi Bu, Yu-Xiang Wang, Sheng Zha, and George Karypis. 2022. Differentially private bias-term only fine-tuning of foundation models. *arXiv preprint arXiv:2210.00036*.

Trevor Campbell and Tamara Broderick. 2019. Automated scalable bayesian inference via hilbert coresets. *The Journal of Machine Learning Research*, 20(1):551–588.

Zachary Charles, Kallista Bonawitz, Stanislav Chiknavaryan, Brendan McMahan, et al. 2022. Federated select: A primitive for communication-and memory-efficient federated learning. *arXiv preprint arXiv:2208.09432*.

Mingqing Chen, Rajiv Mathews, Tom Ouyang, and Françoise Beaufays. 2019. Federated learning of out-of-vocabulary words. *arXiv preprint arXiv: Arxiv-1903.10635*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language*

Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), pages 4171–4186. Association for Computational Linguistics.

Cynthia Dwork. 2010. Differential privacy in new settings. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '10, page 174–183, USA. Society for Industrial and Applied Mathematics.

Cynthia Dwork. 2011. A firm foundation for private data analysis. *Commun. ACM*, 54(1):86–95.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. *Calibrating Noise to Sensitivity in Private Data Analysis*, pages 265–284. Springer Berlin Heidelberg.

Cynthia Dwork and Aaron Roth. 2014. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3–4):211–407.

Maha Elbayad, Jiatao Gu, Edouard Grave, and Michael Auli. 2020. Depth-adaptive transformer. *ICLR*.

Arun Ganesh, Mahdi Haghifam, Milad Nasr, Sewoong Oh, Thomas Steinke, Om Thakkar, Abhradeep Thakurta, and Lun Wang. 2023. Why is public pretraining necessary for private model training? *ArXiv*, abs/2302.09483.

Mitchell Gordon, Kevin Duh, and Nicholas Andrews. 2020. Compressing BERT: Studying the effects of weight pruning on transfer learning. In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 143–155, Online. Association for Computational Linguistics.

Andrew Hard, Chloé M Kiddon, Daniel Ramage, Francoise Beaufays, Hubert Eichner, Kanishka Rao, Rajiv Mathews, and Sean Augenstein. 2018. Federated learning for mobile keyboard prediction.

Andrew Hard, Kurt Partridge, Cameron Nguyen, Niranjan Subrahmanya, Aishanee Shah, Pai Zhu, Ignacio Lopez Moreno, and Rajiv Mathews. 2020. Training keyword spotting models on non-iid data with federated learning. *arXiv preprint arXiv: Arxiv-2005.10406*.

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

Enyi Jiang, Yibo Jacky Zhang, and Oluwasanmi Koyejo. 2023. Federated domain adaptation via gradient projection. *arXiv preprint arXiv:2302.05049*.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, F. Wang, and Qun Liu. 2019. Tinybert: Distilling bert for natural language understanding. *Findings of EMNLP*.

P. Kairouz, B. McMahan, Shuang Song, Om Thakkar, Abhradeep Thakurta, and Zheng Xu. 2021. Practical and private (deep) learning without sampling or shuffling. *International Conference On Machine Learning*.

P. Kairouz, H. B. McMahan, Brendan Avent, A. Bellet, M. Bennis, A. Bhagoji, Keith Bonawitz, Zachary B. Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, S. Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, M. Gruteser, Z. Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, T. Javidi, Gauri Joshi, M. Khodak, Jakub Konecný, A. Korolova, F. Koushanfar, O. Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, M. Mohri, R. Nock, A. Özgür, R. Pagh, Mariana Raykova, Hang Qi, D. Ramage, R. Raskar, D. Song, Weikang Song, S. Stich, Ziteng Sun, A. 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. *Found. Trends Mach. Learn.*

Gavin Kerrigan, Dylan Slack, and Jens Tuyls. 2020. Differentially private language models benefit from public pre-training. In *Proceedings of the Second Workshop on Privacy in NLP*, pages 39–45, Online. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *Conference On Empirical Methods In Natural Language Processing*.

Tian Li, M. Zaheer, Sashank J. Reddi, and Virginia Smith. 2022. Private adaptive optimization with side information. *International Conference On Machine Learning*.

Xuechen Li, Florian Tramèr, Percy Liang, and Tatsunori B. Hashimoto. 2021. Large language models can be strong differentially private learners. *International Conference On Learning Representations*.

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

H. B. McMahan, Eider Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. *International Conference On Artificial Intelligence And Statistics*.

John Nguyen, Jianyu Wang, Kshitiz Malik, Maziar Sanjabi, and Michael Rabbat. 2022. Where to begin? on the impact of pre-training and initialization in federated learning. *arXiv preprint arXiv:2210.08090*.

Natalia Ponomareva, Jasmijn Bastings, and Sergei Vassilvitskii. 2022. Training text-to-text transformers with privacy guarantees. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2182–2193.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Ali Rahimi and Benjamin Recht. 2007. Random features for large-scale kernel machines. *Advances in neural information processing systems*, 20.

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: Arxiv-2009.10031*.

Sashank J. Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečný, Sanjiv Kumar, and Hugh Brendan McMahan. 2021. Adaptive federated optimization. In *International Conference on Learning Representations*.

Mike Schuster and Kaisuke Nakajima. 2012. Japanese and korean voice search. *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5149–5152.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. 2020. Mobilebert: a compact task-agnostic bert for resource-limited devices. *ACL*.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. 2020. Dataset cartography: Mapping and diagnosing datasets with training dynamics. *Conference On Empirical Methods In Natural Language Processing*.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, YaGuang Li, Hongrae Lee, Huaixiu Steven Zheng, Amin Ghafouri, Marcelo Menegali, Yanping Huang, Maxim Krikun, Dmitry Lepikhin, James Qin, Dehao Chen, Yuanzhong Xu, Zhifeng Chen, Adam Roberts, Maarten Bosma, Vincent Zhao, Yanqi Zhou, Chung-Ching Chang, Igor Krivokon, Will Rusch, Marc Pickett, Pranesh Srinivasan, Laichee Man, Kathleen Meier-Hellstern, Meredith Ringel Morris, Tulsee Doshi, Renelito Delos Santos, Toju Duke, Johnny Soraker, Ben Zevenbergen, Vinodkumar Prabhakaran, Mark Diaz, Ben Hutchinson, Kristen Olson, Alejandra Molina, Erin Hoffman-John, Josh Lee, Lora Aroyo, Ravi Rajakumar, Alena Butryna, Matthew Lamm, Viktoriya Kuzmina, Joe Fenton, Aaron Cohen, Rachel Bernstein, Ray Kurzweil, Blaise Aguera-Arcas, Claire Cui, Marian Croak, Ed Chi, and Quoc Le. 2022. Lamda: Language models for dialog applications. *arXiv preprint arXiv: Arxiv-2201.08239*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

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*.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in Neural Information Processing Systems*, 33:5776–5788.

Shanshan Wu, Tian Li, Zachary Charles, Yu Xiao, Ziyu Liu, Zheng Xu, and Virginia Smith. 2022. Motley: Benchmarking heterogeneity and personalization in federated learning. *arXiv preprint arXiv: Arxiv-2206.09262*.

Zhaozhuo Xu, Luyang Liu, Zheng Xu, and Anshumali Shrivastava. 2022a. Adaptive sparse federated learning in large output spaces via hashing. In *Workshop on Federated Learning: Recent Advances and New Challenges (in Conjunction with NeurIPS)*.

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

Zheng Xu, Yanxiang Zhang, Galen Andrew, Christopher Choquette, Peter Kairouz, Brendan McMahan, Jesse Rosenstock, and Yuanbo Zhang. 2023. Federated learning of gboard language models with differential privacy.

Da Yu, Saurabh Naik, Arturs Backurs, Sivakanth Gopi, Huseyin A. Inan, Gautam Kamath, Janardhan Kulkarni, Yin Tat Lee, Andre Manoel, Lukas Wutschitz, Sergey Yekhanin, and Huishuai Zhang. 2022. Differentially private fine-tuning of language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Jacky Zhang, Rajiv Khanna, Anastasios Kyrillidis, and Sanmi Koyejo. 2021. Bayesian coresets: Revisiting the nonconvex optimization perspective. In *International Conference on Artificial Intelligence and Statistics*, pages 2782–2790. PMLR.

A Additional Related Work

Private Federated Learning in On-device NLP Federated learning is designed to collaboratively training NLP models without sharing sensitive user data to protect user privacy. Given relatively small model sizes, state-of-the-art differentially private (DP) learning algorithms (McMahan et al., 2018; Kairouz et al., 2021) have enabled on-device LMs to achieve strong downstream task utility with reasonable user-level differentially privacy guarantee (Dwork, 2010). The success of private FL has also led to real-world applications such as GBoard, which uses on-device LMs for next word prediction (Hard et al., 2018; Ramaswamy et al., 2020). Recent advances in DP optimization (Kairouz et al., 2021) further improves upon the state-of-the-art DP-SGD algorithm (Abadi et al., 2016), providing a practical tool to analyze privacy bound for federated learning.

Privacy-preserving Large NLP Models Scaling up LMs with more data and parameters has significantly improved performance and achieved great success in a variety of NLP tasks. Moreover, recent studies show that LLM has great potential in private learning. For example, Kerrigan et al. (2020) show that public pre-training is helpful for downstream DP fine-tuning. Follow-up studies argue that large pre-trained LMs can be strong differentially private learners with parameter-efficient fine-tuning (Yu et al., 2022; Bu et al., 2022) or full model fine-tuning (Li et al., 2021), narrowing the gap between non-private training and private training. Ganesh et al. (2023) also provide theoretical groundings on the necessity of involving public training into private learning. Motivated by the recent success of LLMs, our work performs comprehensive studies on how to use public data and existing LLMs to help private training of cross-device FL models.

Model Compression for Pre-trained LMs One promising approach to address the resource limitations of LLMs is to compress them into smaller models through various techniques such as knowledge distillation (Jiao et al., 2019; Sun et al., 2020; Wang et al., 2020), or pruning (Elbayad et al., 2020; Gordon et al., 2020). While these techniques have demonstrated success in reducing the size of pre-trained LMs, most resulting models are still too large (with over 10 million parameters) to be effectively deployed on resource-constrained devices. In our work, we also explore the use of knowledge distillation in public training, but with a primary focus on leveraging LLMs to improve sample efficiency in pre-training on-device LMs. We aim to improve the private FL performance of on-device LMs while minimizing the need for large amounts of training data. We recognize that private federated learning can further benefit from advanced model compression techniques, and we leave this as a promising and orthogonal future direction for research in this area.

B Experimental Setup Details

B.1 Verification of Non-overlap between C4 and StackOverflow Datasets

StackOverflow contains 342K clients for training with 135.8M examples. In this section, we detail the method used to verify that there is no explicit overlap between the public C4 dataset and the private StackOverflow dataset utilized in our study.

We explored C4 which has multiple variants[1]: `c4/en, c4/realnewslike, and c4/webtextlike`.

To verify this hypothesis, we conducted a rigorous comparison of these two datasets and its variants. Specifically, we compared the unique identifiers (e.g., URL for webpages in the C4 dataset, and post ID for StackOverflow posts) between the two datasets.

No matching identifiers were found between the `c4/realnewslike` and the StackOverflow dataset. Thus we use the `c4/realnewslike` variant as our public pretraining corpus throughout the experiment.

Through this comprehensive comparison, we have confirmed that there is no explicit overlap between the public C4 dataset and the private StackOverflow dataset. This conclusion is critical to our study as it ensures that the integrity and privacy-preserving conditions of our experiment are maintained.

[1] https://www.tensorflow.org/datasets/catalog/c4

B.2 Pretraining Details

In this section, we outline the detailed procedures followed during the pretraining phase of our experiments. The pretraining phase consisted of the following steps:

1. **Data Preparation:** We tokenized both the C4 and StackOverflow datasets using the SentencePiece tokenizer, as described in the main text. The vocabulary size was set to $32K$ for both datasets.

2. **Model Architecture:** We follow previous work (Wang et al., 2021; Amid et al., 2021; Kairouz et al., 2021; Wu et al., 2022) and use one-layer LSTM and transformer. Both LSTM and transformer has a hidden size of 670 and embedding size of 96.

3. **Training Procedure:** We trained the model using a standard autoregressive LM loss for next token prediction.

4. **Training Hyperparameters:** We employed the Adam optimizer with a learning rate of 1e-3, a batch size of 512, and a maximum sequence length of 20 tokens. We also used gradient clipping to prevent exploding gradients. The model was pretrained for $1400K$ steps on the C4 dataset to cover the whole C4 pretraining corpus.

After pretraining, the model was then fine-tuned on the downstream task using federated learning with differential privacy. Further details regarding the fine-tuning process can be found in the relevant sections of the main text. We show that the pretraining procedure can significantly improve the model's robust performance in the downstream task performance.

B.3 Distillation Details

In this section, we delineate the specifics of our distillation process during the pretraining phase of our on-device LM. The pretraining procedure with distillation is mostly the same as details outlined in B.2 with slight hyper-parameter differences.

We set the temparature $t = 1$ and top-$k = 10$ to extract the logits $\boldsymbol{z}_T$ from teacher LLM. We use grid search to tune the best hyper-parameter $\beta \in \{1e-1, 1e-2, 1e-3\}$ and follow the same pre-training schedules as §3.2 but with a smaller batch size of 128 due to memory constraints.

C Additional Experimental Results

Hyper-parameter Tuning for Federated Learning Federated learning involves numerous hyperparameters, which is crucial for our experiment. Our hyper-parameter tuning strategy follows Xu et al. (2022b).

Throughout our experiments, we fix the number of total rounds $T = 1600$. In each round, we select 100 clients from the shuffled pool for DP-FTRL, ensuring that the clients are disjoint across rounds. Within each client, we fix the number of local epochs to one and set the batch size to 16. We also impose a constraint on the maximum number of samples on each client, limiting it to 256.

We tune the server learning rate, client learning rate and clip norm for a certain given a noise multiplier. Specifically, we use grid search and tune the server learning rate from $\{0.05, 0.1, 0.2, 0.5, 1, 2\}$, the client learning rate from $\{0.01, 0.02, 0.05, 0.1, 0.2, 0.5\}$. We use the adaptive clipping technique in (Andrew et al., 2021; Xu et al., 2023) to help determine the clip norm, which in most of our experiments falls into $\{0.1, 0.3, 0.4, 1\}$.

Abaltion studies on top-k logits We take the top-k logits of the LLM to construct our distillation datasets and pre-train the on-device LMs. Here, we conduct an ablation study by pre-training different on-device LMs with different k and evaluate how top-k logits in distillation can impact the accuracy of private FL. We present our empirical results in Figure 2c and Appendix Figure 4. We observe that pre-training with a larger k is more helpful to achieve better downstream accuracy on private data. To have a reasonable trade-off between dataset size and pre-training performance, we use top-$k = 10$ in all the following experiments.

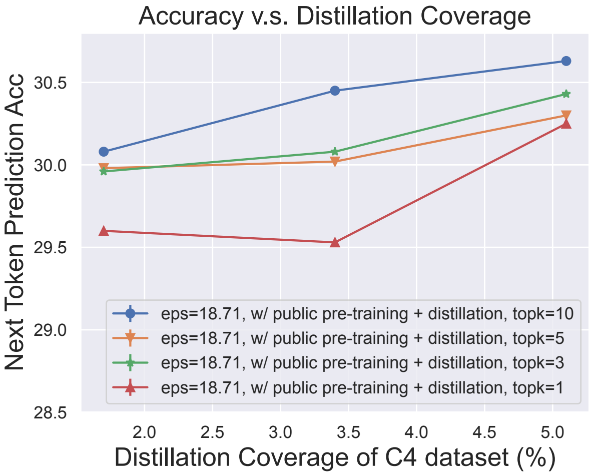


Figure 4: Ablation studies on how distillation steps and top-k logits in distillation impact next token prediction accuracy (Acc.) of on-device LSTM models on the private StackOverflow dataset.

Ablation studies on the timing T' for mid-training T' separates two-stage private federated learning and determines the timing for distribution matching and public training. In this ablation study, we evaluate the dev set accuracy of on-device LSTM given different T' and privacy budgets, as shown in Table 4 and Appendix Table 5. From the table, we can see that the on-device LSTM achieves the best private FL accuracy given $T' = T/2 = 800$. We think the reasons are as follows: when $T' = 0$, we cannot perform distribution matching as the on-device LM is not trained on the private dataset yet, and thus we can only use the randomly sampled data for pre-training; when $T' = 400$, the on-device LM could not be well trained on the private data distribution, thus yielding worse distribution matching quality; when $T' = 1200$ and $T' = 1600$, the private on-device LM is biased towards the public data distribution due to public training, thus giving worse private FL accuracy. As a result, we use $T' = 800$ in our main experiments, as it balances the private federated training and public training to have satisfactory distribution matching capabilities without biasing too much towards the public data distribution.

T'	0	400	800	1200
ε=1.77	25.41	26.43	**26.73**	25.20
ε=18.71	28.38	29.55	**29.70**	28.93

Table 5: Ablation studies on the timing (T') of mid-point public training for on-device LSTM w/o distribution matching.

D Detailed Theoretical Results

D.1 Discussion on the distance metrics of log-density functions

We need to define a meaningful distance metric in order to define the closeness of two log-density functions. To do this, we can choose any inner product $\langle \cdot, \cdot \rangle$ in the function space of $\mathcal{H} = \{f : \mathcal{X} \to \mathbb{R}\}$. Note that the log-density functions $\ell_{\text{pub}}, \ell_{\text{priv}}, \hat{\ell}_{\text{priv}} \in \mathcal{H}$. Accordingly, the norm in the function space $\mathcal{H}$ is denoted as $\| \cdot \|$ and by definition $\forall f \in \mathcal{H} : \|f\| = \sqrt{\langle f, f \rangle}$.

We note that our analysis works for **any** choice of the inner product as long as they don't make the log-densities norm infinite. For a concrete example, we discuss a generalization of the L^2 inner product, i.e., the L^π inner product where π is a distribution on $\mathcal{X}$.

Formally, for this example of $\mathcal{H} = L^\pi$ we define $\langle f, g \rangle_\pi = \mathbb{E}_{x\sim\pi}[f(x)g(x)]$ and $\|f\|_\pi = \sqrt{\mathbb{E}_{x\sim\pi}[f(x)^2]}$.

The L^π is a rather general definition that is common in the literature of Bayesian coresets (Zhang et al., 2021; Campbell and Broderick, 2019) and kernel machine (Rahimi and Recht, 2007). For example, it recovers L^2 if π is chosen to be the uniform distribution on $\mathcal{X}$.

Moreover, if we choose $\pi = p_{\text{priv}}$ as the private data density, we can show that for any probability density function p, the distance between $\log p$ and $\log p_{\text{priv}}$ measured by $L^{p_{\text{priv}}}$ norm upper bounds the KL divergence between p_{priv} and p:

$$\begin{aligned}
\|\log p - \log p_{\text{priv}}\|_\pi^2 &= \mathbb{E}_{x\sim p_{\text{priv}}}[(\log p(x) - \log p_{\text{priv}}(x))^2] = \mathbb{E}_{x\sim p_{\text{priv}}}\left(\log \frac{p(x)}{p_{\text{priv}}(x)}\right)^2 && (4)\\
&\geq \left(\mathbb{E}_{x\sim p_{\text{priv}}} \log \frac{p(x)}{p_{\text{priv}}(x)}\right)^2 && \text{(Jensen's Inequality)}\\
&= (\texttt{KL}(p_{\text{priv}}|p))^2 && (5)
\end{aligned}$$

In general, the distribution π characterize where in $\mathcal{X}$ we want to evaluate a function.

Above we discuss a concrete choice of the inner product and the accordingly the norm to measure the distance between log-density functions. Since our analysis will work with any choice of inner product, we return to using the notation of $\langle\cdot,\cdot\rangle$ and $\|\cdot\|$ to remain generality in our main result.

D.2 Proof

Theorem D.1 (Theorem 5.1 Restated)**.** Let $\epsilon(\hat{f}) = \mathbb{E}[\|\hat{f} - \ell_{\text{priv}}\|^2]$ characterise how good $\hat{f}$ is as an estimator of the true private data log-density ℓ_{priv} for any random function $\hat{f} \in \mathcal{H}$. Consider the following three quantities:

1. $\epsilon(\ell_{\text{pub}})$ that characterizes the error if we use the public log-density function ℓ_{pub} to approximate the ℓ_{priv}
2. $\epsilon(\hat{\ell}_{\text{priv}})$ that characterizes the error if we use the noisy private log-density function $\hat{\ell}_{\text{priv}}$ to approximate the ℓ_{priv}
3. $\epsilon(\hat{h})$ that characterizes the error if we use $\hat{h} = \frac{1}{2}\ell_{\text{pub}} + \frac{1}{2}\hat{\ell}_{\text{priv}}$ to approximate the ℓ_{priv}.

Then,

$$\epsilon(\ell_{\text{pub}}) = d^2_{\text{pub, priv}} \tag{6}$$
$$\epsilon(\hat{\ell}_{\text{priv}}) = \sigma^2_{\text{priv}} \tag{7}$$
$$\epsilon(\hat{h}) = \frac{1}{4}d^2_{\text{pub, priv}} + \frac{1}{4}\sigma^2_{\text{priv}} \tag{8}$$

Proof. We prove a general result which gives the theorem as special cases. For $\beta \in [0,1]$, define

$$\hat{f}_\beta = \beta\ell_{\text{pub}} + (1-\beta)\hat{\ell}_{\text{priv}}. \tag{9}$$

According to the definition of $\epsilon(\hat{f}_\beta) = \mathbb{E}[\|\hat{f}_\beta - \ell_{\text{priv}}\|^2]$, we have

$$\begin{aligned}
\epsilon(\hat{f}_\beta) &= \mathbb{E}[\|\hat{f}_\beta - \ell_{\text{priv}}\|^2] = \mathbb{E}[\|\beta\ell_{\text{pub}} + (1-\beta)\hat{\ell}_{\text{priv}} - \ell_{\text{priv}}\|^2] && (10)\\
&= \mathbb{E}[\|\beta(\ell_{\text{pub}} - \ell_{\text{priv}}) + (1-\beta)(\hat{\ell}_{\text{priv}} - \ell_{\text{priv}})\|^2] && (11)\\
&= \beta^2\|\ell_{\text{pub}} - \ell_{\text{priv}}\|^2 + (1-\beta)^2\mathbb{E}\left[\|\hat{\ell}_{\text{priv}} - \ell_{\text{priv}}\|^2\right] + 2\beta(1-\beta)\mathbb{E}\left[\langle\ell_{\text{pub}} - \ell_{\text{priv}}, \hat{\ell}_{\text{priv}} - \ell_{\text{priv}}\rangle\right] && (12)\\
&= \beta^2 d^2_{\text{pub, priv}} + (1-\beta)^2\sigma^2_{\text{priv}} + 2\beta(1-\beta)\langle\ell_{\text{pub}} - \ell_{\text{priv}}, \mathbb{E}[\hat{\ell}_{\text{priv}}] - \ell_{\text{priv}}\rangle && (13)\\
&= \beta^2 d^2_{\text{pub, priv}} + (1-\beta)^2\sigma^2_{\text{priv}} + 0 && (14)\\
&= \beta^2 d^2_{\text{pub, priv}} + (1-\beta)^2\sigma^2_{\text{priv}} && (15)
\end{aligned}$$

Therefore, we can see that the theorem stands as we substitute $\hat{f}_1 = \ell_{\text{pub}}$, $f_{\frac{1}{2}} = \hat{h}$, and $\hat{f}_0 = \hat{\ell}_{\text{priv}}$. □

D.3 Extended Analysis

Note that in the previous subsection the $\hat{f}_\beta$ is a weighted combination of ℓ_{pub} and $\hat{\ell}_{\text{priv}}$, *i.e.*, $\hat{f}_\beta = (1-\beta)\ell_{\text{pub}} + \beta\hat{\ell}_{\text{priv}}$ where $\beta \in [0,1]$. Therefore, one can show that with the optimal weight $\beta^\star$, it is guaranteed that $\epsilon(\hat{f}_{\beta^\star}) \leq \min\{\epsilon(\ell_{\text{pub}}), \epsilon(\hat{\ell}_{\text{priv}})\}$.

This framework of analysis is general (as it stands with any meaningful inner product and its norm), and it may inspire even better ways to design estimators mitigating the domain shift and private model noise.