arXiv:2305.13865v1 [cs.LG] 23 May 2023

Selective Pre-training for Private Fine-tuning

Da Yu[†] Sivakanth Gopi[‡*] Janardhan Kulkarni[‡*] Zinan Lin[‡*]
Saurabh Naik[§*] Tomasz Lukasz Religa[§*] Jian Yin[†*] Huishuai Zhang[‡*]

May 24, 2023

Abstract

Suppose we want to train text prediction models in email clients or word processors. The models must preserve the privacy of user data and adhere to a specific fixed size to meet memory and inference time requirements. We introduce a generic framework to solve this problem. Specifically, we are given a public dataset D_{pub} and a private dataset D_{priv} corresponding to a downstream task T. How should we pre-train a fixed-size model M on D_{pub} and fine-tune it on D_{priv} such that performance of M with respect to T is maximized and M satisfies differential privacy with respect to D_{priv}? We show that pre-training on a *subset* of dataset D_{pub} that brings the public distribution closer to the private distribution is a crucial ingredient to maximize the transfer learning abilities of M after pre-training, especially in the regimes where model sizes are relatively small. Besides performance improvements, our framework also shows that with careful pre-training and private fine-tuning, *smaller models* can match the performance of much larger models, highlighting the promise of differentially private training as a tool for model compression and efficiency.

1 Introduction

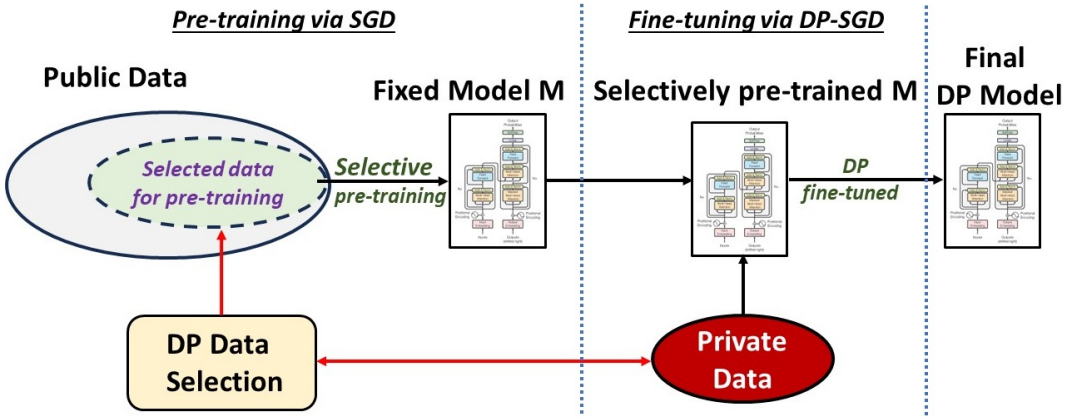

Figure 1: An illustration of our framework to train a given model of fixed size for a domain-specific task. We do selective pre-training of the model to maximize its transfer learning abilities followed by private fine-tuning. Our privacy guarantees take into account both data selection and fine-tuning.

Many papers have shown that deep learning models are vulnerable to attacks aimed at extracting information from the training data [66, 34, 15, 81, 17, 16, 55]. A provable path for mitigating such privacy attacks is to train the models with *differential privacy* (DP) [22], a mathematically rigorous notion for quantifying the privacy leakage

[†]Sun Yat-sen University. {yuda3@mail2,issjyin@mail}.sysu.edu.cn
[‡]Microsoft Research. {sigopi,jakul,zinanlin,huzhang}@microsoft.com
[§]Microsoft. {snaik,toreli}@microsoft.com
[*]Authors are listed in alphabetical order.

of a machine learning model. Over the past few years, there has been rapid progress in the understanding of deep learning with DP, both in terms of computational efficiency [35, 50, 12, 46, 70, 5] and privacy-utility trade-off [50, 80, 20, 56, 79, 85, 86, 27, 64, 13, 60, 54, 42]. One of the findings of these works has been that pre-training (or pre-trained models) is crucial for maximizing performance. There is some theoretical evidence of why and how pre-training helps private learning [49, 24].

Yet, most of these works assume pre-training as a black-box step, and do not tackle the question if there are pre-training strategies that are more friendly for private fine-tuning. Besides scientific curiosity, the question has real practical motivations that we discuss now. Much of the literature on DP fine-tuning, with the exception of Mireshghallah et al. [58] that we discuss in Section 1.3, focus on settings where *inference time* is not a bottleneck and one can deploy models of *any* size. In such a case, existing evidence is that larger models pre-trained on vast amounts of public data for a very long duration perform better with private fine-tuning [50, 80, 56]. However, there are plenty of applications where the size of the model is restricted by the inference time; think of a text prediction model of an email client running on a mobile device or a face identification model running in a security system. In such applications, if the inference time is not good then the *quality* of predictions becomes irrelevant. Further, note also that in both these applications the training data is quite sensitive, and the models should protect the privacy of users. In such a scenario, where one needs to train a model of a certain fixed size, how should we train the model so as to maximize the downstream performance? This constitutes the central question we investigate in this paper.

> *Suppose we want to train a language model of a fixed size for a domain-specific downstream task while respecting differential privacy. As such, how should one pre-train the model on public data and differentially privately fine-tune it on the private data to maximize performance?*

1.1 Our Contributions:

The main conceptual contribution of our work is to bring the focus on pre-training on the public data with the aim of maximizing the transfer learning capabilities of a model with respect to domain-specific private fine-tuning.

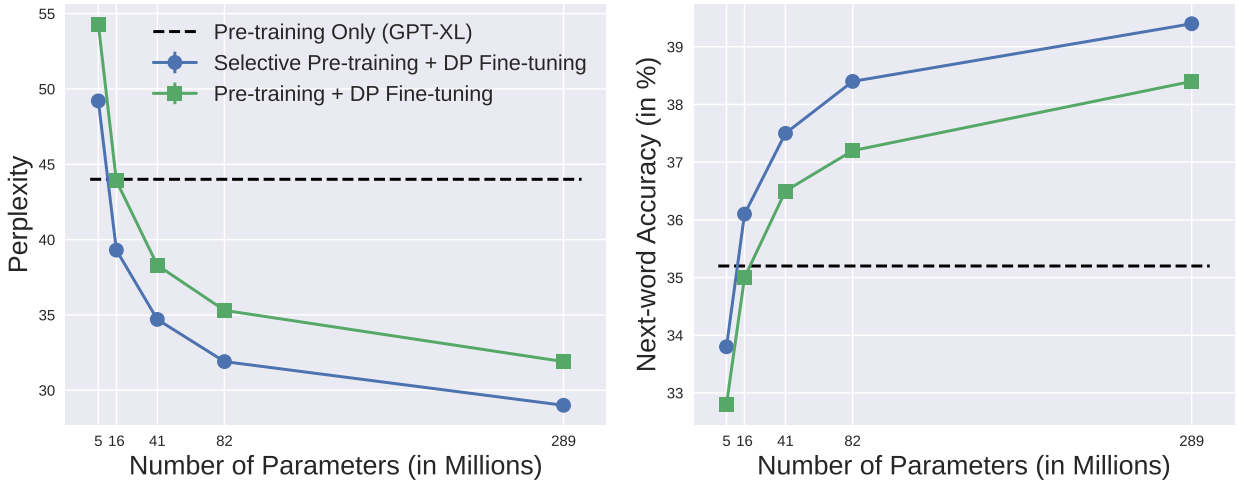

Figure 2: A representative result from our findings. We plot perplexity and top-1 next word accuracy of GPT models on the test set of the Enron email dataset [1]. Overall privacy budget is ($\varepsilon = 7.3, \delta = 1 \times 10^{-7}$). The dashed line shows the zero-shot performance of GPT2-XL with 1.5 billion parameters.

A Framework for Private Training To Maximize Downstream Performance We give a framework for training domain-specific language models of a given size with differential privacy (Figure 1). In our framework, a privacy-preserving algorithm is used to select a subset of public data for pre-training. We name it as *selective pre-training*. Our experiments demonstrate that selective pre-training enhances the transfer learning power of the model when privately fine-tuned with DP-SGD. Figure 2 is a representative summary of our findings. The main takeaway is the following:

> *Our new framework leads to clear improvements in the downstream task performance on standard NLP benchmarks, both in perplexity and prediction accuracy, for all model sizes we experimented with.*

Private Algorithm for Dataset Selection There is a growing interest in understanding the right data selection for pre-training both for general domain models (GPTs, PaLM, LLaMA) and specific models (Khanmigo, Codex, AlphaFold) in non-private deep learning literature; see [77, 31, 41] and references therein. It is reasonably well established that the pre-training dataset has a significant impact on the downstream performance of the model and its transfer learning abilities. As a byproduct of our framework, we introduce these problems to private learning literature, show a simple method for data selection that is more friendly for private learning. Our experiments in Section 4.1 and 4.2 bring to light the following phenomenon:

> *The benefits of selective pre-training are greater for private deep learning compared to non-private deep learning.*

Differential Privacy as a Tool For Model Compression Consider Figure 2 again. Observe that the performance of a tiny model with 16 million parameters can match the zero-shot performance of GPT2-XL *public model* with 1.5 billion parameters. Furthermore, smaller DP models with 41 and 82 million parameters that were trained using our new framework match or surpass the perplexity and top-1 accuracy of *larger DP models* with 82 and 289 million parameters respectively that were not carefully pre-trained. These observations have remarkable consequences for training domain-specific models in real-world applications.

> *In all the scenarios where differential privacy provides meaningful protection of data privacy, a small amount of private data along with our selective pre-training followed by private fine-tuning can lead to a substantial compression of the model sizes, thus improving the inference time and reduction in the inference cost.*

In other words, intuitively speaking, a model that has no access to high-quality data has to be larger to generalize, whereas smaller models that have access to high-quality data can outperform bigger models in domain-specific tasks. Thus, differentially private training (combined with our framework) can truly unlock the value of high-quality but sensitive data. We anticipate that this conceptual message of private learning as a tool for model efficiency, which to our knowledge has not been emphasized before in the literature, will find more applications in deep learning. On a scientific front, it brings the data quality aspect to focus in understanding deep learning.

Real-world Impact: Our framework was recently used in training an industry grade differentially private text prediction language model that now serves many applications in a big AI company. As text prediction models (on email clients/servers, word processors, etc.) serve billions of queries per hour, the inference cost savings due to the decrease in model size are significant. Further, due to better inference time, online performance metrics, such as the number of predictions accepted by the users, also improve. While we do not think these facts as a scientific contribution of our work, it highlights the efficacy of our framework for real-world datasets and applications, besides the standard benchmarks considered in this paper.

1.2 Discussion: On the Role of Pre-training in Private Learning

Our work shows that selective pre-training improves the transfer learning performance of a model. Does this imply that the power of pre-training is all due to more data that has a similar distribution as private data? This may not always be the case, particularly for transformer-based models as self-attention layers can learn complex set functions. Even when the public data distribution is quite different from the private distribution, pre-training models on a large corpus of public data can still benefit private learning. In the absence of selective pre-training, however, to see the benefits *one needs significantly larger and more powerful models trained on vast amounts of public data for a very long time*, which is consistent with what previous papers have shown [50, 80, 56, 20].

We demonstrate these intuitions by fine-tuning a 739M GPT model that is pre-trained from scratch on Spanish corpus [32] on the Enron email dataset. The privacy parameters are the same as those in Figure 2. The model achieves 37.7% accuracy on next-word prediction, which is comparable to the performance of an 82M GPT model that uses standard pre-training, as well as the performance of a 41M GPT model trained with selective pre-training. In contrast, the accuracy of a randomly initialized 739M GPT model when privately fine-tuned only achieves an accuracy of 17.2%. *These experiments show that pre-training is a crucial ingredient for private learning.* Finally, for extremely large models with 100s of billions of parameters, we believe selective pre-training is not for performance improvements but to reduce the pre-training cost and data cleaning.

1.3 Putting Our Framework in Context

For general literature on private deep learning and fine-tuning we refer the readers to [2, 35, 43, 50, 12, 46, 70, 5, 80, 20, 56, 79, 86, 64, 13, 60], and references there in. To the best of our knowledge, no prior work in DP literature has studied selective pre-training from scratch and its impact on the transfer learning abilities of a model. Our work is at the intersection of several related topics, and we give a brief overview of how our work fits into the broader literature on the topic.

Reducing Distribution Disparity Between Public and Private Data Public data has been widely used to improve private data analysis [62, 3, 8, 7, 42, 51, 85, 53, 4, 78, 48, 10]. To address the distribution shift between private and public data, a recent line of research explores domain adaption [74, 83] in the context of private learning [75, 84, 9]. However, these works are not very relevant to our setting. More closer to our work is the independent and concurrent works of Hou et al. [39] and Gu et al. [29] that study how to privately select an optimal public dataset from an explicitly given list of public datasets. For instance, suppose the private dataset is CIFAR-10, and available public datasets are MNIST, CIFAR100, and ImageNet. The goal is to design a private algorithm to find which of the three public datasets is better suited for private learning on CIFAR-10. In this paper, we explore how to select a subset of a single public dataset on a sample-by-sample basis, similar to what is studied in the non-private world [77]. Our algorithm does not require any explicit division of public data and runs efficiently on billions of tokens, making it well-suited for finding the right pre-training data for language models. Moreover, our emphasis is not just on performance, but how pre-training impacts accuracy-vs-model size trade-offs.

Non-Private Data Selection Automatic data selection and cleaning, along with how the pre-training data impacts the downstream task performance are important problems in deep learning. See [77, 31, 11, 18, 41, 36, 57, 47, 19] and references there in. Yet, the literature is scarce on the impact of selective pre-training on the model, except the recent concurrent work of [77]. Our work explores these questions in the context of private learning, with an emphasis on how the quality of data affects performance and model size. As a pilot study on designing privacy-preserving data selection algorithms, we use simple classification-based approaches that are easy to privatize and provide a clear illustration of the main messages of the paper. Exploring more sophisticated approaches [77] for private data selection is an interesting future direction.

DP Model Compression The setting considered in this paper, where we want to train a model of a certain size, is related to the model compression literature, which was recently studied by Mireshghallah et al. [58] in the private learning context. They use black-box compression techniques, such as distillation [37] or pruning [33], at the *fine-tuning* stage. The experiments in Section 4.2 show that using our framework alone can improve upon their results. Moreover, our framework is compatible with existing model compression techniques [37, 33]. How those techniques can be combined with selective pre-training to further improve the model performance and/or improve the compression ratio is an interesting research direction.

1.4 Preliminaries

We begin with the formal definition of differential privacy.

Definition 1 ((ε, δ)-Differential Privacy (DP) [22])**.** *A randomized algorithm $\mathcal{A}$ is (ε,δ)-differentially private if for any two neighboring datasets D and D', which differ in exactly one datapoint, and for every subset $\mathcal{S}$ of possible outputs:* $\Pr[\mathcal{A}(D) \in \mathcal{S}] \leq e^{\varepsilon} \Pr[\mathcal{A}(D') \in \mathcal{S}] + \delta$.

In the context of deep learning, DP guarantees that the trained model *weights* are private with respect to a training dataset, and hence can be released publicly. To train a deep learning model with privacy, the most popular method is to first release the gradients of an optimizer with differential privacy and then update the model with privatized gradients [67, 6, 2]. Using the privatized gradients does not increase the privacy cost because of the *post-processing* property of differential privacy [23]. We follow the approach in Abadi et al. [2] to make gradients differentially private. Abadi et al. [2] augment each minibatch of gradients with per-example gradient clipping and Gaussian noise addition steps. The clipping step ensures that no one user's sample significantly changes the weights of the model and the noise added guarantees that the contribution of a single example is masked. At a high level, the privacy analysis in Abadi et al. [2] proceeds by first showing that each release of the clipped and noisy gradient is differentially private for some privacy parameters, then applying the composition theorem of differential privacy across all the iterations. After obtaining the privatized gradients, we use Adam optimizer [44] to update the models.

2 Problem Statement and Our Framework

Input to our problem is a private dataset D_{priv} corresponding to a downstream task T, a model M of size p, privacy parameters $\varepsilon > 0$, $\delta > 0$, and a public dataset D_{pub}. Our goal is to train M on public and private datasets with the aim of maximizing the downstream performance on the task T. The entire process should be (ε, δ)-differentially private with respect to D_{priv}. The constraint on model size is important to compare various algorithms in our setting. In applications, the constraints on model size arise naturally as a consequence of memory and/or inference time requirements.

2.1 Our Framework

Our framework consists of the following steps, which is illustrated in Figure 1.

1. *Privacy Preserving Data Selection*: Given D_{priv}, invoke a privacy preserving data selection algorithm $\mathcal{A}_{\text{select}}$ to find a $D'_{\text{pub}} \subseteq D_{\text{pub}}$. The privacy budget for this step is $(\varepsilon_1, \delta_1)$.
2. *Non-Private Pre-training*: Pre-train the model M on D'_{pub} with a standard pre-training algorithm. This step does not consume any privacy budget.
3. *Private Fine-tuning*: Fine-tune M on D_{priv} with a differentially private algorithm $\mathcal{A}_{\text{finetune}}$. The privacy budget for this step is $(\varepsilon_2, \delta_2)$.

The non-private pre-training step can be viewed as a post-processing function to $\mathcal{A}_{\text{select}}$ and thus no privacy budget is consumed. The composition property of differential privacy, Theorem 2.1, ensures the entire framework is differentially private.

Theorem 2.1 (Advanced Composition of (ε, δ)-DP [69])**.** *Let $\mathcal{A}_{select} : \mathcal{D} \to \mathcal{D}'$ be the selection algorithm and D'_{pub} be its output. Further let $\mathcal{A}_{pre+fine} : \mathcal{D} \times \mathcal{D}' \to \Theta$ be the algorithm that implements non-private pre-training and private fine-tuning. The adaptive composition $\mathcal{A}(D_{priv}) = \big(\mathcal{A}_{select}(D_{priv}), \mathcal{A}_{pre+fine}(D_{priv}, D'_{pub})\big)$ is (ε, δ)-DP with respect to D_{priv} for any $\delta > \delta_1 + \delta_2$ with*

$$\varepsilon = \min\left\{\varepsilon_1 + \varepsilon_2, \frac{1}{2}(\varepsilon_1^2 + \varepsilon_2^2) + \sqrt{2\log(1/\delta')(\varepsilon_1^2 + \varepsilon_2^2)}\right\},$$

where $\delta' = \delta - (\delta_1 + \delta_2)$.

In our experiments, we use the Privacy Random Variable (PRV) Accountant [28, 25, 45], following the choice in [50, 80, 58]. The PRV accountant gives a tighter privacy bound through *numerical* composition. The rest of the paper is devoted to describing the first step of our framework, followed by experiments to verify the effectiveness of our methods on different datasets.

3 Privacy Preserving Data Selection

We describe our approach to implementing a privacy-preserving data selection algorithm. We provide a specific implementation of our framework and demonstrate its effectiveness, however, our approach is general and can be combined with other private data selection algorithms. Our experiments indicate that the data selected by our algorithm result in a distribution that is more similar to the target dataset than the original pre-training data.

3.1 Our Implementation of Data Selection

Our framework is loosely inspired by the data cleaning framework used in GPT3 and PaLM models [11, 18], although motivations are a bit different. The classifiers in Brown et al. [11], Chowdhery et al. [18] are trained to filter out noisy documents from datasets. In fact, the source datasets in our paper, i.e., OpenWebText and Wikipedia, are considered positive examples in Brown et al. [11]. Our classifier is trained to recognize examples that are similar to samples in the target data. We initialize the classifier with a pre-trained LM and fine-tune it with differential privacy to predict whether a sentence is sampled from the distribution of the target data. We use the classifier to predict all sentences in the source data and *rank* them according to confidence scores. Although deep neural networks could be overconfident and need

calibration in some applications [30, 82], not calibrating the outputs does not affect our algorithm because calibration does not change the relative ranking among sentences. We select the top sentences until we reach the target number of pre-training tokens. Figure 3 shows an overview of our implementation.

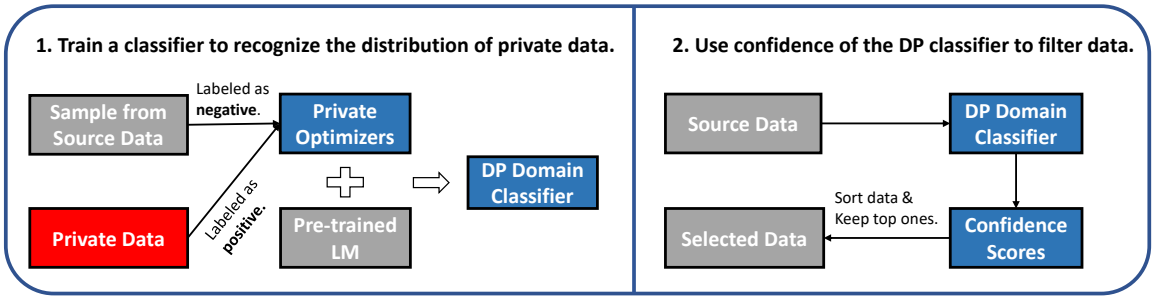

Figure 3: An illustration of the data selection process. We train a classifier to identify target examples and rank source examples based on the confidence scores of the classifier.

We create a training set to teach the classifier to recognize a target data distribution. Sentences in the target dataset are labelled as positive. Random samples from the source data are labelled as negative. It has been widely observed that a larger training set helps private learning [6, 71]. Therefore we set the number of negative examples as five times larger than the number of positive examples. The privacy cost of training the classifier is accounted in the overall privacy cost. We run experiments with the Enron Email dataset [1] as the target and the OpenWebText dataset [26] as the source. The classifier is initialized with an 82M GPT model pre-trained on OpenWebText. With a single Tesla A100 GPU, it takes approximately one hour for training the classifier. With eight Tesla V100 GPUs, it takes less than two hours for computing the confidence scores for all sequences in OpenWebText. The privacy guarantee is $(0.7, 1 \times 10^{-8})$-DP if we only consider the privacy cost of this step. More implementation details are in Section 4.1. The trained classifier achieves an F1-score of 98.5%. The classifier achieves an F1-score of 92.7% if it is not initialized with a pre-trained LM.

We use the trained classifier to select 10% of OpenWebText. We plot the word clouds of Enron email, OpenWebText, and the selected subset of OpenWebText (Figure 4), to visually illustrate the dataset selected by our algorithm. The word clouds only show the nouns to exclude common prepositions and verbs. There are 28 nouns which appear in both the top 100 nouns of the Enron email dataset and the top 100 nouns of OpenWebText. The number of overlaps increases to 39 when comparing Enron email with the selected subset of OpenWebText, suggesting the trained domain classifier is an effective tool for data selection. In Appendix A.1, we also present the results of using GLUE [73] tasks as the targets and the pre-training corpus of BERT [21] as the source.

(a) Enron Email (b) OpenWebText (c) Selected OpenWebText

Figure 4: The 100 most frequent nouns in Enron email, OpenWebText, or a selected subset of OpenWebText (10%). A larger font size indicates that the word appears more frequently. Green words are the 100 most frequent nouns in Enron Email. OpenWebText and selected OpenWebText have 28 and 39 words, respectively, that are among the 100 most frequent nouns in Enron Email.

4 Experimental Evaluation

We evaluate our proposed framework on language generation and understanding tasks, comparing on datasets that most previous works used [50, 80]. The goal here is to empirically verify the main points made in the introduction: *1) Our framework gives clear gains in the downstream task performance (Figure 2, 6, and 7) 2) Selective pre-training is more important in private learning (Figure 5 and 7). 3) Our framework can be used as an effective tool for model compression (Figure 2 and 7).*

4.1 Implementing the Framework on the Enron Email Dataset

Our first target task is causal language modeling on the Enron email dataset. The dataset contains approximately 0.5 million (M) emails written by employees of the Enron Corporation and is publicly available for research use [1]. We choose this dataset because its distribution closely resembles some private datasets in the real world for which it is hard to find off-the-shelf pre-training data.

4.1.1 Experiment Setup

We briefly describe important parameters of our experimental setup. More details are in Appendix B.

Target and Source Data We divide the text into sequences of length 256 and let each sequence be a datapoint, which constitutes the granularity of our privacy guarantees. There are $\sim$70K sequences in total. We use 80% of them for training and evenly split the rest 20% for validation and testing. The source data is OpenWebText [26] which contains $\sim$4 billion tokens.

Models Models in this section are from the GPT family [63]. We change the number of layers, hidden size, and intermediate size of the fully connected block to get five different model sizes (5M, 16M, 41M, 82M, and 289M). Following previous work [76], we do not include embedding matrices when computing the number of parameters. During inference, an embedding layer is a lookup table that is much more efficient than linear layers. Details of the models are in Table 2 in Appendix B.

Data Selection We use the algorithm in Section 3 to select 2M sequences from the source data for pre-training. The baseline in this section is selecting 2M random sequences. In Section 4.2, we experiment with GLUE tasks and *use pre-training with all of the source data* as another baseline. We use nodes with 8x Nvidia Tesla V100 GPUs to pre-train all models. Our compute budget allows us to train the largest model (289.3M) for $\sim$6 epochs on the selected data.

Private Learning The privacy budget is $(7.3, 1 \times 10^{-7})$-DP, similar to previous works on this topic [50, 80]. To reduce the privacy cost of hyperparameter tuning [52, 61, 59], we follow the findings in previous work to set most of the hyperparameters and only tune the learning rate to adapt to models of different sizes. The hyperparameters for private learning are listed in Table 4 in Appendix B.

4.1.2 Selective Pre-training Provides Clear Gains, Model Efficiency

Figure 2 (see Section 1.1) shows the perplexity and next-word prediction accuracy of different models on the test split of the Enron email dataset. We also present the next-word accuracy and its standard deviation across random seeds in Table 1 in Appendix A.2 as a complementary to Figure 2. It is clear from the figure that our framework improves performance compared to existing techniques.

More significantly, the 41M (82M) model using selective pre-training outperforms (matches) the 82M (289M) model using normal pre-training. *This demonstrates that the proposed framework can be used to improve the efficiency-utility trade-off of private learning.* We also include the zero-shot performance of the off-the-shelf GPT2-XL model (1.5 billion parameters) in Figure 2. The zero-shot performance of GPT2-XL is worse than the models that have access to private data and are of much smaller size. These findings highlight the importance of private data, which can be loosely treated as high quality data, as well as the importance of privacy-enhancing technologies that facilitate the trustworthy use of such data. Figure 10 in Appendix A.2 also presents the results under different privacy budgets (ε ranging from 2.3 to 10.9). We observe consistent gains when the selective pre-training framework is used.

4.1.3 Selective Pre-training in Non-private vs Private World.

We also fine-tune the models without differential privacy to see whether selective pre-training improves downstream performance in non-private learning. The results are in Figure 5. In this case, selective pre-training still improves the performance of all models. However, selective pre-training yields larger improvements in private learning. When

trained without differential privacy, the 82M (41M) model using selective pre-training is inferior to its larger counterpart. *This suggests that the benefits of selective pre-training are greater in private learning compared to non-private learning.*

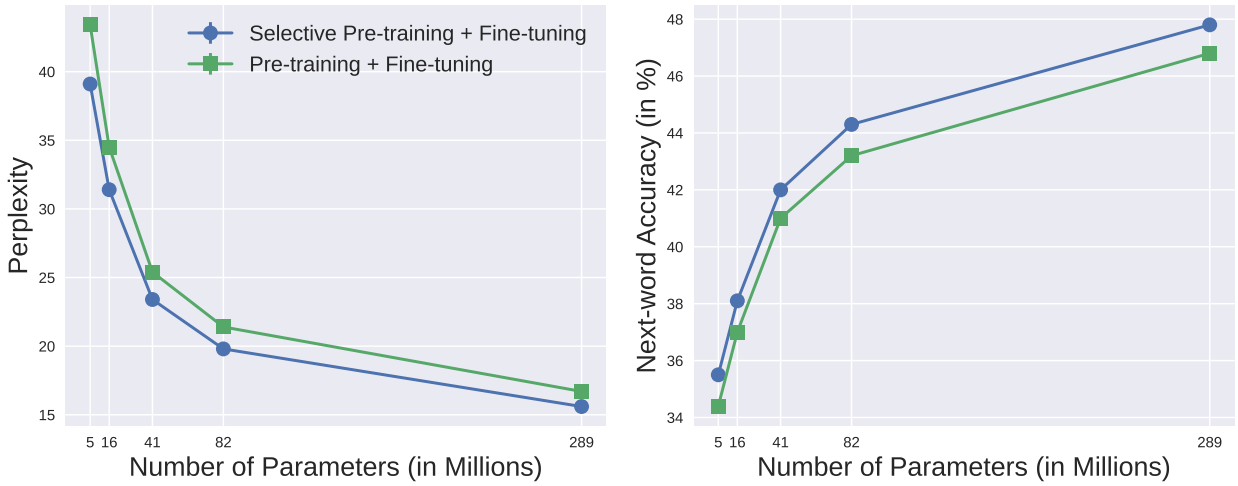

Figure 5: Perplexity and top-1 next word accuracy of GPT models on the test set of the Enron email dataset. The models are trained without DP. Selective pre-training still improves over standard pre-training, however, the improvements are smaller compared to private learning.

4.2 Experiments on GLUE

We conduct experiments on the General Language Understanding Evaluation (GLUE) benchmark [73]. It is a common benchmark for fine-tuning language models with DP [79, 50, 14]. Our results show that selective pre-training also improves DP fine-tuning for language understanding tasks.

4.2.1 Experiment Setup

Target and Source Data Our target tasks in this section are MNLI and SST-2, which have respectively the largest and smallest number of examples among the four tasks studied in previous work [79, 50, 14, 58]. The numbers of training examples (N) in MNLI and SST-2 are 393K and 67K. The source data for GLUE tasks is the pre-training corpus of BERT [21]; It consists of a subset of Wikipedia[1] and the entire Bookcorpus[2]. The source dataset has approximately 3.5 billion tokens.

Model Sizes We follow previous work to use models from the BERT family [21]. We consider four different model sizes (5M, 10M, 25M, and 44M). Details of the models are in Appendix B.

Data Selection For MNLI and SST-2, we experiment with selecting varying numbers of tokens from the source data. The target numbers of pre-training tokens are 20M, 40M, 200M, 400M, 800M, 1200M, and 2000M. More complete implementation details on data selection are in Appendix B.

Baselines The baselines include pre-training on randomly selected source data and pre-training on all source data. There are two additional baselines for the 44M model. The first is directly fine-tuning DistillBERT [65] with differential privacy. DistillBERT is distilled from BERT-base on the source data. The second is the best result in Mireshghallah et al. [58]. Mireshghallah et al. [58] compress a DP fine-tuned BERT-base model using differentially private distillation or pruning. The architecture of the compressed models in Mireshghallah et al. [58] and Sanh et al. [65] are of the same architecture as the 44M model. Although our framework is compatible with the techniques in Mireshghallah et al. [58] and Sanh et al. [65], we include the two additional baselines to demonstrate that the proposed framework alone is a competitive approach for model compression in private learning.

[1] https://huggingface.co/datasets/wikipedia/
[2] https://huggingface.co/datasets/bookcorpus

Private Learning We adopt the setup in Mireshghallah et al. [58]. The privacy budget is $(4, 1/10N)$-DP. The hyperparameters for private learning are also documented in Appendix B.

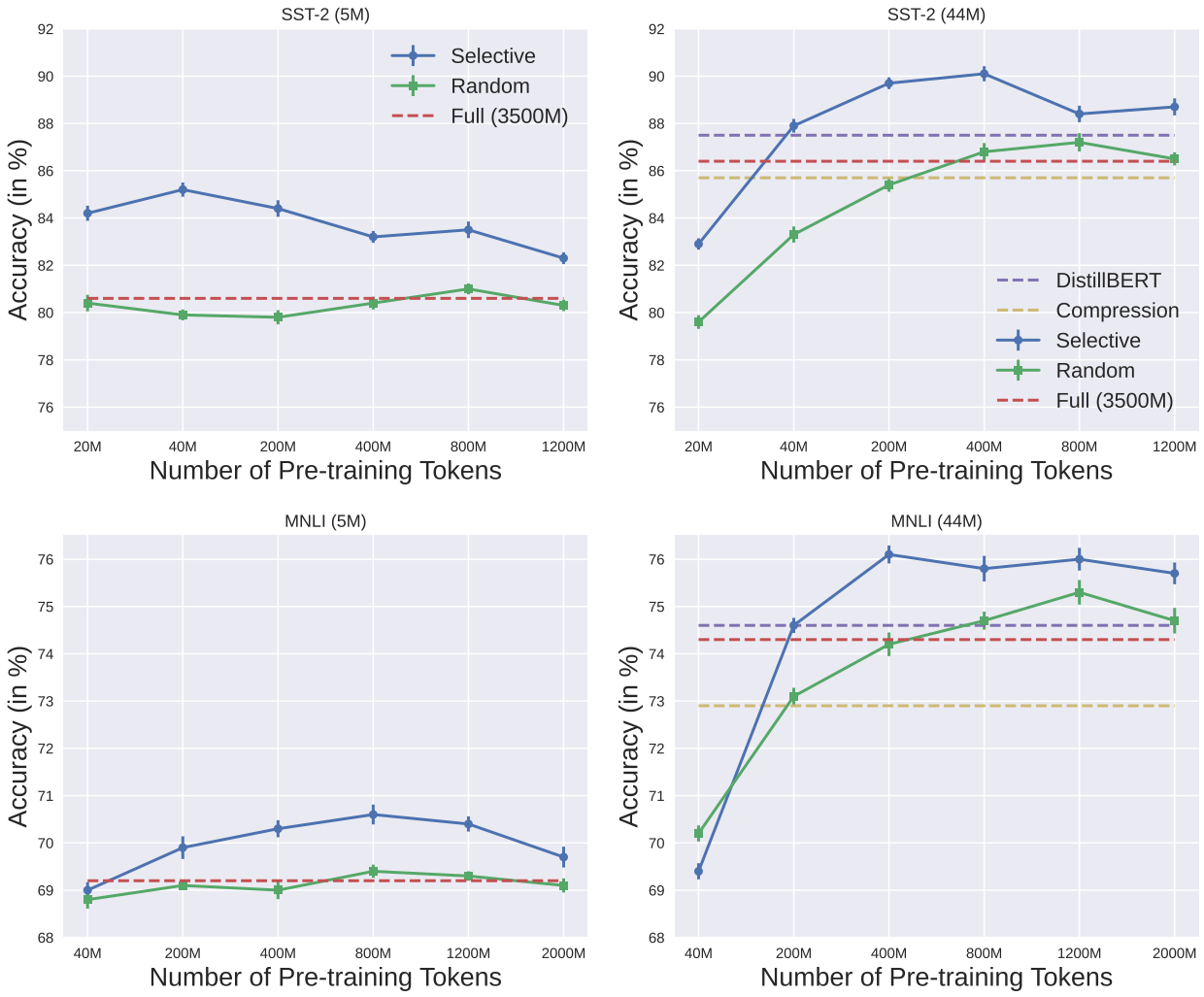

Figure 6: Results of pre-training with various numbers of tokens. The first column shows the results of 5M models and the second column shows the results of 44M models. Selective pre-training outperforms baseline algorithms in most of the cases.

4.2.2 Selective Pre-training Outperforms Baselines, Improves Model Efficiency

Figure 6 shows the test accuracy on MNLI and SST-2 after privately fine-tuning models pre-trained with varying numbers of tokens. Our first finding is that, for most of the settings, selective pre-training outperforms all the algorithms examined. On SST-2, selective pre-training achieves accuracy that is 4.6% and 3.7% higher than the accuracy of full pre-training for the 5M and 44M models, respectively. On MNLI, the accuracy improvements are 1.4% and 1.8%, respectively. Considering the simplicity of these tasks, these improvements are non-trivial. Our second finding is that, for a model of fixed size, *increasing the number of pre-training tokens does not necessarily lead to better downstream accuracy.* This suggests that there may be an optimal number of pre-training tokens for a given model size [38], further emphasizing the need to choose a task-specific subset from a large source data.

Figure 7 shows the test accuracy of models of different sizes. When trained with differential privacy, the 25M model with selective pre-training achieves comparable or better performance than the 44M baseline models, aligning with our observations on the Enron email dataset. The accuracy gains on SST-2 are greater than those achieved on MNLI,

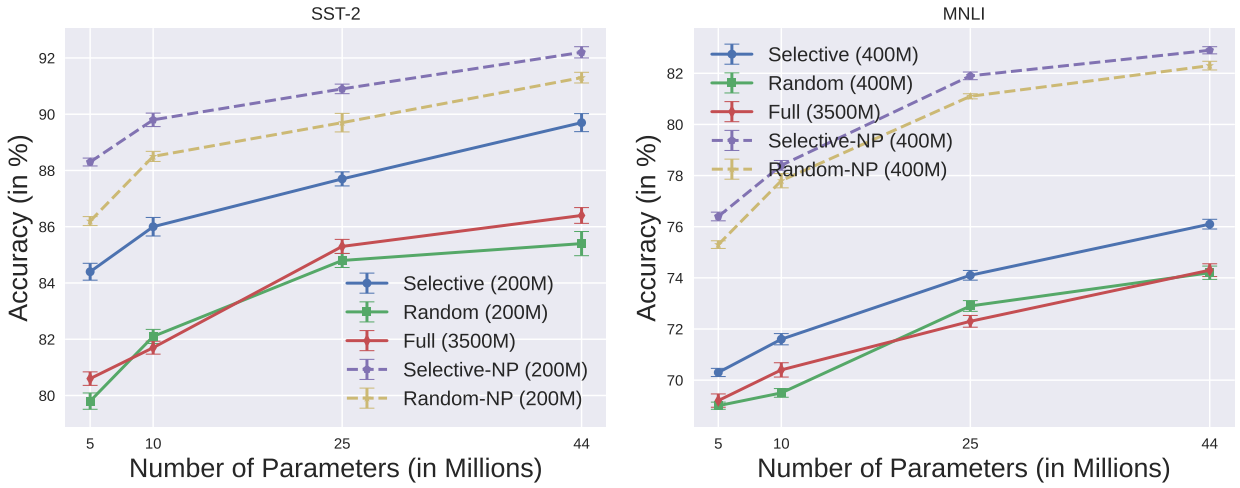

Figure 7: Results of pre-training with different model sizes. The numbers in the brackets are the numbers of tokens used for pre-training. 'NP' denotes that the models are fine-tuned without DP. Selective pre-training consistently improves performance across all settings. The improvements for models trained with DP are larger.

likely because MNLI data distribution is relatively closer to Wikipedia corpus; see Appendix A.1 for the word clouds comparison.

5 Conclusions, Open Questions, and Limitations

A limitation of our work is that we provide DP guarantees only with respect to private datasets and not public datasets. In building applications, it is important to take into account the privacy risks of public data [72]. Discussion in Section 1.2 and our selective pre-training can give some pointers on how to address these concerns. This work initiated the study of pre-training strategies that are friendly for the private fine-tuning of domain-specific models. Immediate follow-up questions are: are there better algorithms for data selection? How should one select the pre-training data for general purpose or multi-task models? Our work shows that training on the full public data may not be always optimal, and the number of pre-training tokens is a function of model size. These conclusions touch upon scaling laws [68, 38], and understanding the scaling behaviour of models in private deep learning is a fascinating research direction [64]. Finally, differential privacy as a tool for model efficiency can have a huge impact in real-world applications, and more research in this space can lead to a better understanding of its power.

References

[1] Enron email dataset. https://www.cs.cmu.edu/~enron/.

[2] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of ACM Conference on Computer and Communications Security*, 2016.

[3] Noga Alon, Raef Bassily, and Shay Moran. Limits of private learning with access to public data. *Advances in neural information processing systems*, 2019.

[4] Ehsan Amid, Arun Ganesh, Rajiv Mathews, Swaroop Ramaswamy, Shuang Song, Thomas Steinke, Vinith M Suriyakumar, Om Thakkar, and Abhradeep Thakurta. Public data-assisted mirror descent for private model training. In *International Conference on Machine Learning*, 2022.

[5] Rohan Anil, Badih Ghazi, Vineet Gupta, Ravi Kumar, and Pasin Manurangsi. Large-scale differentially private bert. *Findings of the Association for Computational Linguistics: EMNLP 2022*, 2022.

[6] Raef Bassily, Adam Smith, and Abhradeep Thakurta. Differentially private empirical risk minimization: Efficient algorithms and tight error bounds. *Annual Symposium on Foundations of Computer Science*, 2014.

[7] Raef Bassily, Albert Cheu, Shay Moran, Aleksandar Nikolov, Jonathan Ullman, and Steven Wu. Private query release assisted by public data. In *International Conference on Machine Learning*, 2020.

[8] Raef Bassily, Shay Moran, and Anupama Nandi. Learning from mixtures of private and public populations. *Advances in Neural Information Processing Systems*, 2020.

[9] Raef Bassily, Mehryar Mohri, and Ananda Theertha Suresh. Principled approaches for private adaptation from a public source. In *International Conference on Artificial Intelligence and Statistics*, 2023.

[10] Alex Bie, Gautam Kamath, and Vikrant Singhal. Private estimation with public data. *Advances in Neural Information Processing Systems*, 2022.

[11] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, 2020.

[12] Zhiqi Bu, Sivakanth Gopi, Janardhan Kulkarni, Yin Tat Lee, Hanwen Shen, and Uthaipon Tantipongpipat. Fast and memory efficient differentially private-sgd via jl projections. *Advances in Neural Information Processing Systems*, 2021.

[13] Zhiqi Bu, Jialin Mao, and Shiyun Xu. Scalable and efficient training of large convolutional neural networks with differential privacy. *Advances in Neural Information Processing Systems*, 2022.

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

[15] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. *USENIX Security Symposium*, 2021.

[16] Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. Quantifying memorization across neural language models. *International Conference on Learning Representations*, 2023.

[17] Christopher A Choquette-Choo, Florian Tramer, Nicholas Carlini, and Nicolas Papernot. Label-only membership inference attacks. In *International Conference on Machine Learning*, 2021.

[18] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

[19] Cody Coleman, Christopher Yeh, Stephen Mussmann, Baharan Mirzasoleiman, Peter Bailis, Percy Liang, Jure Leskovec, and Matei Zaharia. Selection via proxy: Efficient data selection for deep learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL `https://openreview.net/forum?id=HJg2b0VYDr`.

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

[21] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *Conference of the North American Chapter of the Association for Computational Linguistics*, 2019.

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

[23] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 2014.

[24] Arun Ganesh, Mahdi Haghifam, Milad Nasr, Sewoong Oh, Thomas Steinke, Om Thakkar, Abhradeep Thakurta, and Lun Wang. Why is public pretraining necessary for private model training? *arXiv preprint arXiv:2302.09483*, 2023.

[25] Badih Ghazi, Pritish Kamath, Ravi Kumar, and Pasin Manurangsi. Faster privacy accounting via evolving discretization. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato, editors, *International Conference on Machine Learning, ICML 2022*, 2022.

[26] Aaron Gokaslan and Vanya Cohen. Openwebtext corpus. `http://Skylion007.github.io/OpenWebTextCorpus`, 2019.

[27] Aditya Golatkar, Alessandro Achille, Yu-Xiang Wang, Aaron Roth, Michael Kearns, and Stefano Soatto. Mixed differential privacy in computer vision. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[28] Sivakanth Gopi, Yin Tat Lee, and Lukas Wutschitz. Numerical composition of differential privacy. *Advances in Neural Information Processing Systems*, 2021.

[29] Xin Gu, Gautam Kamath, and Zhiwei Steven Wu. Choosing public datasets for private machine learning via gradient subspace distance. *arXiv preprint arXiv:2303.01256*, 2023.

[30] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, 2017.

[31] Suchin Gururangan, Ana Marasovic, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don't stop pretraining: Adapt language models to domains and tasks. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational*, 2020.

[32] Asier Gutiérrez-Fandiño, Jordi Armengol-Estapé, Marc Pàmies, Joan Llop-Palao, Joaquin Silveira-Ocampo, Casimiro Pio Carrino, Aitor Gonzalez-Agirre, Carme Armentano-Oller, Carlos Rodriguez-Penagos, and Marta Villegas. Maria: Spanish language models. *arXiv preprint arXiv:2107.07253*, 2021.

[33] Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. *Advances in neural information processing systems*, 28, 2015.

[34] Jamie Hayes, Luca Melis, George Danezis, and Emiliano De Cristofaro. LOGAN: membership inference attacks against generative models. *Proceedings on Privacy Enhancing Technologies*, 2019.

[35] Jiyan He, Xuechen Li, Da Yu, Huishuai Zhang, Janardhan Kulkarni, Yin Tat Lee, Arturs Backurs, Nenghai Yu, and Jiang Bian. Exploring the limits of differentially private deep learning with group-wise clipping. *International Conference on Learning Representations*, 2023.

[36] Danny Hernandez, Tom B. Brown, Tom Conerly, Nova DasSarma, Dawn Drain, Sheer El Showk, Nelson Elhage, Zac Hatfield-Dodds, Tom Henighan, Tristan Hume, Scott Johnston, Benjamin Mann, Chris Olah, Catherine Olsson, Dario Amodei, Nicholas Joseph, Jared Kaplan, and Sam McCandlish. Scaling laws and interpretability of learning from repeated data. *CoRR*, abs/2205.10487, 2022. doi: 10.48550/arXiv.2205.10487. URL `https://doi.org/10.48550/arXiv.2205.10487`.

[37] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[38] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. An empirical analysis of compute-optimal large language model training. *Advances in Neural Information Processing Systems*, 2022.

[39] Charlie Hou, Hongyuan Zhan, Akshat Shrivastava, Sid Wang, Sasha Livshits, Giulia Fanti, and Daniel Lazar. Privately customizing prefinetuning to better match user data in federated learning. *ICLR Workshop on Pitfalls of Limited Data and Computation for Trustworthy ML*, 2023.

[40] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *International Conference on Learning Representations*, 2022.

[41] Saachi Jain, Hadi Salman, Alaa Khaddaj, Eric Wong, Sung Min Park, and Aleksander Madry. A data-based perspective on transfer learning. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[42] Peter Kairouz, Mónica Ribero, Keith Rush, and Abhradeep Thakurta. (nearly) dimension independent private ERM with adagrad rates via publicly estimated subspaces. In *Proceedings of the 34th Annual Conference on Learning Theory*, 2021.

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

[44] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 2015.

[45] Antti Koskela, Joonas Jälkö, and Antti Honkela. Computing tight differential privacy guarantees using FFT. In Silvia Chiappa and Roberto Calandra, editors, *International Conference on Artificial Intelligence and Statistics*, 2020.

[46] Jaewoo Lee and Daniel Kifer. Scaling up differentially private deep learning with fast per-example gradient clipping. *Privacy Enhancing Technologies*, 2021.

[47] Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. Deduplicating training data makes language models better. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational*

Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022, pages 8424–8445. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.acl-long.577. URL https://doi.org/10.18653/v1/2022.acl-long.577.

[48] Tian Li, Manzil Zaheer, Sashank Reddi, and Virginia Smith. Private adaptive optimization with side information. In *International Conference on Machine Learning*, 2022.

[49] Xuechen Li, Daogao Liu, Tatsunori B Hashimoto, Huseyin A Inan, Janardhan Kulkarni, Yin-Tat Lee, and Abhradeep Guha Thakurta. When does differentially private learning not suffer in high dimensions? *Advances in Neural Information Processing Systems*, 2022.

[50] Xuechen Li, Florian Tramer, Percy Liang, and Tatsunori Hashimoto. Large language models can be strong differentially private learners. *International Conference on Learning Representations*, 2022.

[51] Chong Liu, Yuqing Zhu, Kamalika Chaudhuri, and Yu-Xiang Wang. Revisiting model-agnostic private learning: Faster rates and active learning. *The Journal of Machine Learning Research*, 2021.

[52] Jingcheng Liu and Kunal Talwar. Private selection from private candidates. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, 2019.

[53] Terrance Liu, Giuseppe Vietri, Thomas Steinke, Jonathan Ullman, and Steven Wu. Leveraging public data for practical private query release. In *International Conference on Machine Learning*, 2021.

[54] Zelun Luo, Daniel J. Wu, Ehsan Adeli, and Li Fei-Fei. Scalable differential privacy with sparse network finetuning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 5059–5068. Computer Vision Foundation / IEEE, 2021. doi: 10.1109/CVPR46437.2021.00502. URL https://openaccess.thecvf.com/content/CVPR2021/html/Luo_Scalable_Differential_Privacy_With_Sparse_Network_Finetuning_CVPR_2021_paper.html.

[55] Tomoya Matsumoto, Takayuki Miura, and Naoto Yanai. Membership inference attacks against diffusion models. *arXiv preprint arXiv:2302.03262*, 2023.

[56] Harsh Mehta, Abhradeep Thakurta, Alexey Kurakin, and Ashok Cutkosky. Large scale transfer learning for differentially private image classification. *arXiv preprint arXiv:2205.02973*, 2022.

[57] Sören Mindermann, Jan Markus Brauner, Muhammed Razzak, Mrinank Sharma, Andreas Kirsch, Winnie Xu, Benedikt Höltgen, Aidan N. Gomez, Adrien Morisot, Sebastian Farquhar, and Yarin Gal. Prioritized training on points that are learnable, worth learning, and not yet learnt. *CoRR*, abs/2206.07137, 2022. doi: 10.48550/arXiv.2206.07137. URL https://doi.org/10.48550/arXiv.2206.07137.

[58] Fatemehsadat Mireshghallah, Arturs Backurs, Huseyin A Inan, Lukas Wutschitz, and Janardhan Kulkarni. Differentially private model compression. *Advances in Neural Information Processing Systems*, 2022.

[59] Shubhankar Mohapatra, Sajin Sasy, Xi He, Gautam Kamath, and Om Thakkar. The role of adaptive optimizers for honest private hyperparameter selection. In *Proceedings of the AAAI conference on artificial intelligence*, 2022.

[60] Ashwinee Panda, Xinyu Tang, Vikash Sehwag, Saeed Mahloujifar, and Prateek Mittal. Dp-raft: A differentially private recipe for accelerated fine-tuning. *arXiv preprint arXiv:2212.04486*, 2022.

[61] Nicolas Papernot and Thomas Steinke. Hyperparameter tuning with renyi differential privacy. *International Conference on Learning Representations*, 2022.

[62] Nicolas Papernot, Martín shokri, Ulfar Erlingsson, Ian Goodfellow, and Kunal Talwar. Semi-supervised knowledge transfer for deep learning from private training data. In *International Conference on Learning Representations*, 2017.

[63] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 2019.

[64] Tom Sander, Pierre Stock, and Alexandre Sablayrolles. Tan without a burn: Scaling laws of dp-sgd. *arXiv preprint arXiv:2210.03403*, 2022.

[65] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *NeurIPS Workshop on Energy Efficient Machine Learning and Cognitive Computing*, 2019.

[66] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *IEEE Symposium on Security and Privacy*, 2017.

[67] Shuang Song, Kamalika Chaudhuri, and Anand D Sarwate. Stochastic gradient descent with differentially private updates. In *Global Conference on Signal and Information Processing*, 2013.

[68] Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari Morcos. Beyond neural scaling laws: beating power law scaling via data pruning. *Advances in Neural Information Processing Systems*, 2022.

[69] Thomas Steinke. Composition of differential privacy & privacy amplification by subsampling. *arXiv preprint arXiv:2210.00597*, 2022.

[70] Pranav Subramani, Nicholas Vadivelu, and Gautam Kamath. Enabling fast differentially private sgd via just-in-time compilation and vectorization. *Advances in Neural Information Processing Systems*, 2021.

[71] Florian Tramèr and Dan Boneh. Differentially private learning needs better features (or much more data). *International Conference on Learning Representations*, 2021.

[72] Florian Tramèr, Gautam Kamath, and Nicholas Carlini. Considerations for differentially private learning with large-scale public pretraining. *CoRR*, abs/2212.06470, 2022. doi: 10.48550/arXiv.2212.06470. URL `https://doi.org/10.48550/arXiv.2212.06470`.

[73] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *EMNLP Workshop on BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, 2018.

[74] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. *International Conference on Learning Representations*, 2021.

[75] Qian Wang, Zixi Li, Qin Zou, Lingchen Zhao, and Song Wang. Deep domain adaptation with differential privacy. *IEEE Transactions on Information Forensics and Security*, 2020.

[76] Mengzhou Xia, Zexuan Zhong, and Danqi Chen. Structured pruning learns compact and accurate models. *Annual Meeting of the Association for Computational Linguistics*, 2022.

[77] Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy Liang. Data selection for language models via importance resampling. *arXiv preprint arXiv:2302.03169*, 2023.

[78] Jiaxi Yang and Xiang Cheng. Public data assisted differential private deep learning. In *International Joint Conference on Neural Networks*, 2022.

[79] Da Yu, Huishuai Zhang, Wei Chen, Jian Yin, and Tie-Yan Liu. Large scale private learning via low-rank reparametrization. In *International Conference on Machine Learning*, 2021.

[80] Da Yu, Saurabh Naik, Arturs Backurs, Sivakanth Gopi, Huseyin A Inan, Gautam Kamath, Janardhan Kulkarni, Yin Tat Lee, Andre Manoel, Lukas Wutschitz, et al. Differentially private fine-tuning of language models. *International Conference on Learning Representations*, 2022.

[81] Chiyuan Zhang, Daphne Ippolito, Katherine Lee, Matthew Jagielski, Florian Tramèr, and Nicholas Carlini. Counterfactual memorization in neural language models. *arXiv preprint arXiv:2112.12938*, 2021.

[82] Hanlin Zhang, Xuechen Li, Prithviraj Sen, Salim Roukos, and Tatsunori Hashimoto. A closer look at the calibration of differentially private learners. *arXiv preprint arXiv:2210.08248*, 2022.

[83] L Zhang and X Gao. Transfer adaptation learning: A decade survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[84] Maobo Zheng, Xiaojian Zhang, Xuebin Ma, et al. Unsupervised domain adaptation with differentially private gradient projection. *International Journal of Intelligent Systems*, 2023.

[85] Yingxue Zhou, Zhiwei Steven Wu, and Arindam Banerjee. Bypassing the ambient dimension: Private sgd with gradient subspace identification. *International Conference on Learning Representations*, 2021.

[86] Yuqing Zhu, Xiang Yu, Manmohan Chandraker, and Yu-Xiang Wang. Private-knn: Practical differential privacy for computer vision. *Conference on Computer Vision and Pattern Recognition*, 2020.

(a) SST-2 (b) Wikipedia+Bookcorpus (c) Selected Wikipedia+Bookcorpus

Figure 8: The 100 most frequent nouns in SST-2, the source data, and a selected subset of the source data. Green words are the 100 most frequent nouns in SST-2. The source data and the selected subset have 25 and 40 words, respectively, that are among the 100 most frequent nouns in SST-2.

(a) MNLI (b) Wikipedia+Bookcorpus (c) Selected Wikipedia+Bookcorpus

Figure 9: The 100 most frequent nouns in MNLI, the source data, and a selected subset of the source data. Green words are the 100 most frequent nouns in MNLI. The source data and the selected subset have 44 and 51 words, respectively, that are among the 100 most frequent nouns in MNLI.

A More Experiments

A.1 Results of Data Selection for GLUE Tasks

We plot the word clouds of SST-2/MNLI and (selected) source data to further demonstrate that the distribution of selected data is closer to the distribution of target data. The source data for SST-2 and MNLI is a subset of Wikipedia and the entire Bookcorpus.

The domain classifiers of SST-2 and MNLI are trained the same way as illustrated in Section 3. We select 400M tokens for SST-2 and MNLI, separately. The word clouds of the most frequent 100 nouns are in Figure 8 and 9. We exclude common prepositions and verbs in the word clouds. On SST-2, our selection algorithm improves the number of overlaps between the source data and the target data from 25 to 40. On MNLI, our algorithm improves the number of overlaps from 44 to 51. The results explain our findings in Section 4.2 that selective pre-training yields larger performance improvements on SST-2 than on MNLI.

A.2 More Experiments on the Enron Email Dataset

Table 1 shows the top-1 next word prediction accuracy on the test split of the Enron email dataset as well as the standard deviation over five random seeds. With selective pre-training, a 41M model achieves an accuracy of 37.5% which is 0.3% higher than the accuracy of an 82M model that is not carefully pre-trained.

We also test selective pre-training under different privacy budgets. Figure 10 presents perplexity and next-word prediction accuracy of 5M and 289M GPT models under a wide range of ε (ranging from 2.3 to 10.9). We fix the privacy parameter δ as $1 \times 10^{-7} < 1/10N$. We found that selective pre-training leads to similar improvements across all the choices of ε.

Table 1: Next word prediction accuracy (in %) of GPT models on the Enron email dataset. The overall privacy budget is $(7.3, 1 \times 10^{-7})$.

Parameters	5M	16M	41M	82M	289M
Random	$32.8_{\pm 0.02}$	$35.0_{\pm 0.01}$	$36.5_{\pm 0.01}$	$37.2_{\pm 0.02}$	$38.4_{\pm 0.02}$
Top	$33.8_{\pm 0.03}$	$36.1_{\pm 0.02}$	$37.5_{\pm 0.04}$	$38.4_{\pm 0.02}$	$39.4_{\pm 0.01}$

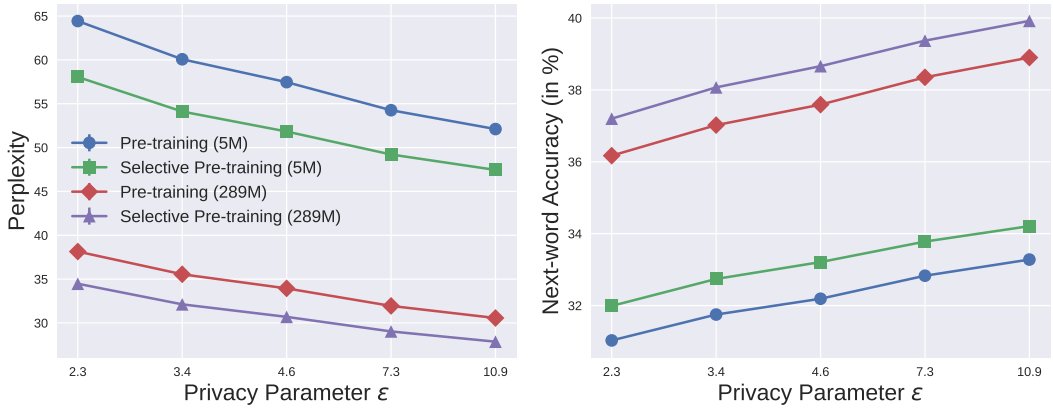

Figure 10: Perplexity and top-1 next-word accuracy on the Enron email dataset. We consider a wide range of ε (ranging from 2.3 to 10.9). The numbers in brackets are the number of model parameters. The privacy parameter δ is 1×10^{-7}. Selective pre-training yields consistent gains across all ε evaluated.

B Implementation Details

This section expands on the implementation details that are omitted from the main text due to space constraints. Our source code will be released at `https://github.com/dayu11/selective_pretraining_for_private_finetuning`.

Details of the Models Let L, d, and d_{FFN} be the number of layers, hidden size, and intermediate size of the fully connected block, respectively. We change L, d, d_{FFN} to get different model sizes. Other architecture hyperparameters are the same as those in Devlin et al. [21] and Radford et al. [63]. Table 2 and 3 show the model details for the Enron email dataset and GLUE tasks, respectively.

Data Selection for Enron Email The text in OpenWebText is also divided into sequences of length 256. To construct the training set of the domain classifier, we randomly sample $5N$ sequences from the source data as negative samples, where N is the number of examples in the target data. As a result, the training set of the domain classifier is 6 times larger than the target data. This significantly reduces the privacy cost of training the domain classifier because the probability of a target example being sampled becomes 6 times smaller. We initialize the domain classifier with an 82M GPT model pre-trained on OpenWebText and fine-tune it with DP-Adam on the constructed training set.

Param.	5M	16M	41M	82M	289M
L	4	4	6	12	24
d	312	576	768	768	1024
d_{FFN}	1248	2304	3072	3072	4096

Table 2: Architecture hyperparameters of the models for the Enron email dataset.

Param.	5M	10M	25M	44M
L	4	6	6	6
d	312	384	576	768
d_{FFN}	1200	1200	2304	3072

Table 3: Architecture hyperparameters of the models for GLUE tasks.

Table 4: Hyperparameters for private fine-tuning. We use N to denote the size of the target dataset.

Pre-training Method	Standard	Selective
Noise multiplier (Enron)	1.00	1.03
Noise multiplier (SST-2)	1.36	1.38
Noise multiplier (MNLI)	1.44	1.46
Epochs (Domain Classifier)	N/A	3
Epochs (Target Task)	30	
Clipping norm	1	
Learning rate	[1e-4, 5e-4, 1e-3, 3e-3]	
Weight decay	0	
Batchsize	$\lfloor 0.03N \rfloor$	
Privacy budget	$(7.3, 1 \times 10^{-7})$ for Enron; $(4, 1/10N)$ for GLUE	

Data Selection for GLUE Tasks Because the positive examples in SST-2 and MNLI are natural sentences instead of sequences of fixed length, we sample natural sentences in the source data as negative examples for training the domain classifier. The domain classifier is initialized with BERT-base. In MNLI, a single example contains two natural sentences, i.e., a premise and a hypothesis. In this case, only one of the two sentences is chosen randomly as a positive example. The number of negative examples is also $5N$.
The pre-training sequences in Devlin et al. [21] are of a fixed length. Each sequence may consist of several natural sentences. To get the ranking score of a sequence, we first break a fixed-length sequence into natural sentences and use the domain classifier to predict those sentences. The maximum confidence of the sentences is used as the ranking score for the sequence.

Hyperparameters For Pre-training The pre-training process uses common hyperparameters in the literature. For pre-training models from the BERT family, we follow the hyperparameters in Devlin et al. [21]. The hyperparameters for pre-training models from the GPT family are as follows. We use a dropout probability of 0.1 and a weight decay of 0.01. The β_1 and β_2 of Adam are 0.9 and 0.999, respectively. All models are pre-trained from scratch for 100K iterations with a batch size of 128. The initial learning rate is 5×10^{-4} and follows a linear decay schedule.

Hyperparameters For Private Fine-tuning We follow the findings in previous work to set most of the hyperparameters [50, 58]. We only tune the learning rate to adapt to the various model sizes we studied. Table 4 summarizes the hyperparameters for private learning. We use the parameter-efficient fine-tuning algorithm LoRA [40] to improve the efficiency of the DP fine-tuning of GPT models [80]. We do not use LoRA for the DP fine-tuning of BERT models to get a fair comparison to Mireshghallah et al. [58]. For a given set of hyperparameters, we use the PRV accountant to get the noise multiplier of DP-Adam. If we use selective pre-training, then the noise multiplier is slightly larger because we need to account for the privacy cost of training a domain classifier. We repeat each private fine-tuning experiment 5 and 3 times with different random seeds for the Enron email dataset and GLUE, respectively.