Journal of Privacy and Confidentiality
Vol. 14 (2) 2024

TPDP 2022

Submitted March 2023
Published May, 2024

DIFFERENTIALLY PRIVATE FINE-TUNING OF LANGUAGE MODELS

DA YU[1], SAURABH NAIK[2], ARTURS BACKURS[3], SIVAKANTH GOPI[3], HUSEYIN A. INAN[3], GAUTAM KAMATH[4], JANARDHAN KULKARNI[3], YIN TAT LEE[3], ANDRE MANOEL[3], LUKAS WUTSCHITZ[2], SERGEY YEKHANIN[3], AND HUISHUAI ZHANG[3]

[1] Sun Yat-sen University
[2] Microsoft
[3] Microsoft Research
[4] University of Waterloo

Abstract. We give simpler, sparser, and faster algorithms for differentially private fine-tuning of large-scale pre-trained language models that achieve the state-of-the-art privacy versus utility tradeoffs on many standard NLP tasks. We propose a meta-framework for this problem, inspired by the recent success of highly parameter-efficient methods for fine-tuning. Our experiments show that differentially private adaptations of these approaches outperform previous private algorithms in three important dimensions: utility, privacy, and the computational and memory cost of private training. On many commonly studied datasets, the utility of private models approaches that of non-private models. For example, on the MNLI dataset we achieve an accuracy of 87.8% using RoBERTa-Large and 83.5% using RoBERTa-Base with a privacy budget of $\epsilon = 6.7$. In comparison, absent privacy constraints, RoBERTa-Large achieves an accuracy of 90.2%. Our findings are similar for natural language generation tasks. Privately fine-tuning with DART, GPT-2-Small, GPT-2-Medium, GPT-2-Large, and GPT-2-XL achieve BLEU scores of 38.5, 42.0, 43.1, and 43.8 respectively (privacy budget of $\epsilon = 6.8, \delta =$ 1e-5), whereas the non-private baseline is 48.1. All our experiments suggest that larger models are better suited for private fine-tuning; while they are well known to achieve superior accuracy non-privately, we find that they also better maintain their accuracy when privacy is introduced.

Key words and phrases: differential privacy, large language models.

* Aside from the first and second authors, all other authors are listed in alphabetical order. Conference version of this paper appeared in the proceedings of the 10th International Conference on Learning Representations (ICLR 2022). E-mail addresses: yuda3@mail2.sysu.edu.cn, {snaik,lukas.wutschitz}@microsoft.com, {arturs.backurs,sigopi,huseyin.inan,jakul,yintatlee,amonteiroman,yekhanin,huzhang}@microsoft.com, g@csail.mit.edu.

[1] Work was done while an intern at Microsoft Research Asia.

[4] Work supported by an NSERC Discovery Grant.

www.journalprivacyconfidentiality.org
DOI:10.29012/jpc.v0.i0.880

1. Introduction

Deep learning models are well known to leak sensitive information about the dataset when trained using conventional methods (Shokri et al., 2017; Carlini et al., 2019, 2021). To combat this issue, models can instead be trained to guarantee differential privacy (DP) (Dwork et al., 2006b), a strong notion of data privacy which limits the influence of any individual training point on the final model. While DP is one of the few approaches capable of providing machine learning models with rigorous privacy guarantees, it generally comes at a cost in terms of test accuracy. One oft-cited explanation is that the constraint of DP necessitates much more training data (Tramèr and Boneh, 2021; Feldman, 2020; Brown et al., 2021). Unfortunately, more training data may be hard to acquire, particularly in settings where privacy is a concern.

Parallel to these developments, Transformer-based (Vaswani et al., 2017) large language models (LLMs), including the BERT (Devlin et al., 2019; Liu et al., 2019) and GPT (Radford et al., 2018, 2019; Brown et al., 2020) families, have enabled significant progress in natural language processing, achieving state-of-the-art accuracy in almost every task considered. These models are first pre-trained on an extremely large and diverse public dataset. The weights are then fine-tuned for each task of interest using a much smaller task-specific dataset. For example, a single pre-trained GPT-family model may be fine-tuned for various downstream tasks, such as email reply suggestion, sentence completion in text editors, language translation, and more. This two-stage paradigm can naturally be adapted to solve tasks in private learning, automatically addressing the aforementioned data shortage issue via the massive scale of the public pre-training dataset. One may pre-train the model on public data as usual,[1] but then fine-tune the model *privately*.

Despite the success of these models, task-specific fine-tuning introduces a number of technical challenges. In the non-private setting, the immense size of LLMs makes it impractical to fine-tune the full model and store a separate copy of the parameters for hundreds of downstream tasks. Things only get worse with privacy, which leads to overheads in terms of running time, memory usage, and most importantly, accuracy. The magnitude of noise introduced to a model due to DP grows as the model size increases (Bassily et al., 2014; Abadi et al., 2016; Bun et al., 2014), which can overwhelm any signal for larger models. A recent line of work in the non-private literature has proposed parameter-efficient methods to alleviate the issues of storage and computational cost for fine-tuning (Houlsby et al., 2019; Li and Liang, 2021; Aghajanyan et al., 2021; Hu et al., 2022; Mahabadi et al., 2021). The main focus of our work is to explore parameter-efficiency in the context of private learning.

1.1. Our Contributions.

Our primary contribution is to show that advanced parameter-efficient methods can lead to *simpler* and significantly improved algorithms for private fine-tuning. Our framework is illustrated in Figure 1. Our findings and contributions are summarized as follows:

- **State-of-the-art utility and privacy.** Empirical evaluation of our algorithms reveals that they achieve state-of-the-art accuracy versus privacy tradeoffs, improving upon the previous best (Yu et al., 2021b). More importantly, for many fine-tuning tasks, the utility

[1]Despite the fact that the pre-training data is public, there may nonetheless be privacy concerns related to personal or copyrighted data. However, since these pre-trained models have already been released, any associated privacy loss has already been incurred.

[2]https://gluebenchmark.com/leaderboard

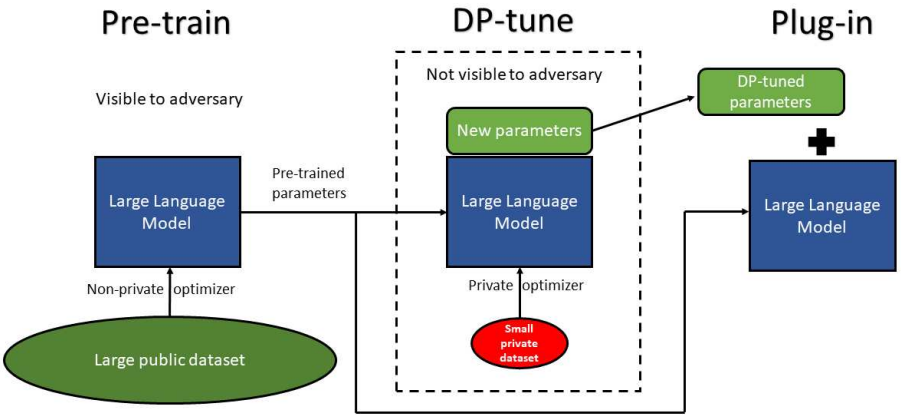

FIGURE 1. An illustration of our framework
First, the model is pre-trained on a large, public dataset. Next, new parameters are introduced and privately fine-tuned on a smaller, private task-specific dataset. The original parameters are frozen during this process. Finally, the fine-tuned new parameters may be released publicly and plugged-in to the model for downstream tasks, while still preserving privacy of the private dataset.

of models trained with DP approaches that of non-private models. For example, privately fine-tuning RoBERTa-Large on the MNLI data set (Williams et al., 2018), we achieve an accuracy of 87.8% with a privacy budget of ($\epsilon = 6.7, \delta =$ 1e-6). Without privacy guarantees, RoBERTa-Large achieves an accuracy of 90.2% (GPT-3 is known to achieve 91.7% (Hu et al., 2022)); see Table 1 for a summary. We also explore private natural language generation tasks, fine-tuning GPT-2 models on the E2E dataset (Novikova et al., 2017). Again, the utility approaches non-private levels; we achieve a ROUGE-L score of 67.8 with GPT-2-Large and ($\epsilon = 6.0, \delta =$ 1e-5), compared to 72.0 without privacy.

- **Larger models are better.** Prior work has consistently shown that larger language models achieve better accuracy for downstream tasks. Our results give evidence that this phenomenon extends to the private setting. For example, on the MNLI dataset, RoBERTa-Base achieves an accuracy of 83.5% (versus 87.6% non-privately, a drop of 4.1%) whereas RoBERTa-Large achieves an accuracy of 87.8% (versus 90.2% non-privately, a drop of 2.4%), both under a privacy budget of ($\epsilon = 6.7, \delta =$ 1e-6). Similarly, privately fine-tuning (using LoRA (Hu et al., 2022)) on DART ($\epsilon = 6.8, \delta =$ 1e-5), GPT-2-Medium

TABLE 1. Accuracy of fine-tuning for downstream tasks, RoBERTa-Large (in %).

Method	MNLI	SST-2	QQP	QNLI	Avg.	Trained params
Non-private SOTA[2]	92.3	97.5	90.9	96.7	94.4	N/A
Non-private fine-tuning	90.2	96.4	92.2	94.7	93.4	100%
Our results ($\epsilon = 6.7$)	87.8	95.3	87.4	90.8	90.3	0.94%

Our results achieve accuracy comparable to full fine-tuning non-privately, while simultaneously guaranteeing differential privacy and modifying less than 1% of the parameters. We choose $\delta = 1 \times 10^{-5}$ for SST-2 and QNLI and $\delta = 1 \times 10^{-6}$ for MNLI and QQP due to their dataset sizes. Implementation details are in Section 4.1. The non-private SOTA results are obtained from fine-tuning a model with 5.4B parameters.

TABLE 2. Fine-tuning GPT-2 models on the DART dataset.

Model	BLEU (DP)	BLEU (non-private)	Drop due to privacy
GPT-2-Medium	42.0	47.1	5.1
GPT-2-Large	43.1	47.5	4.4
GPT-2-XL	43.8	48.1	4.3

We observe that larger models have better utility, both in absolute numbers, and in terms of preserving non-private utility. DP parameters are ($\epsilon = 6.8, \delta =$ 1e-5).

achieves a BLEU score of 42.0 (versus 47.1 non-privately, a drop of 5.1) while GPT-2-XL achieves a BLEU score of 43.8 (versus 48.1 non-privately, a drop of 4.3), see Table 2. Observe that utility improves with model size in two ways: both in terms of absolute numbers, as well as the drop incurred due to privacy. While the power of large models has been established in the non-private setting, we find this phenomenon quite surprising under DP. There is often a tension when choosing private model architectures; larger models may have higher capacity but necessitate the introduction of more noise. Consequently, smaller and simpler private models achieve better accuracy in several settings (Papernot et al., 2019; Tramèr and Boneh, 2021). In contrast, our experiments show that fine-tuning the biggest models achieves the best accuracy,[3] which we consider to be one of our main findings.

- **Simpler, sparser, and faster.** Beyond accuracy concerns, DP requirements also lead to significant overheads in terms of computation and memory usage. The large number of parameters contributes to the high cost of training LLMs, and things get worse under privacy, which has been documented to increase training time by up to two orders of magnitude (Carlini et al., 2019; Subramani et al., 2021). The parameter-efficient approaches we employ partially offset the issue overheads present; as we only update a small fraction of the total number of parameters, training becomes considerably more computationally and memory efficient. Furthermore, as in the non-private setting, this framework leads to a modular design, where a single large pre-trained model can be augmented with lightweight modifications for each individual downstream task.

To the best of our knowledge, we are the first to fine-tune GPT-2-XL using differential privacy. GPT-2-XL is the largest model (with 1.5B parameters) trained thus far using DP. Given our state-of-the-art results for a variety of standard NLP tasks using advanced fine-tuning techniques, we believe that our paper will serve as a benchmark for further work in this direction. For example, the best average accuracy achieved by the prior work of Yu et al. (2021b) on four standard NLP tasks in Table 1 is 83.9% using $\epsilon = 8$ (and the same δ as in Table 1), whereas we can achieve an average accuracy of 90.3% using $\epsilon = 6.7$ by a combination of better algorithms, larger models, and new privacy accounting techniques.

Finally, though recently considered elsewhere (see Section 5), we put further focus on the framing of public pre-training and private fine-tuning as an important conceptual direction in DP deep learning.

[3] An alternative perspective is that what we currently think of as "large" language models are relatively small, and we are yet to reach the point where the benefits of model size on accuracy are outweighed by the drawbacks.

2. Preliminaries and Prior Algorithm Baselines

Recall the formal definition of differential privacy.

Definition 2.1 (Differential Privacy (DP) (Dwork et al., 2006b,a))**.** *A randomized algorithm $\mathcal{A}$ is (ϵ,δ)-differentially private if for any two neighboring datasets D and D', which differ in exactly the data pertaining to a single user, and for all sets $\mathcal{S}$ of possible outputs:* $\Pr[\mathcal{A}(D) \in \mathcal{S}] \leq e^{\epsilon} \Pr[\mathcal{A}(D') \in \mathcal{S}] + \delta$.

Two datasets D, D' are neighboring datasets if they differ in exactly one sample. In this paper, we use the add-remove neighboring relation, i.e., D can be transformed into D' by adding/removing one sample. We review prior techniques for private fine-tuning.

2.1. Full Fine-tuning via DPSGD.

To train a machine learning model with privacy, the most popular algorithm is the celebrated DP stochastic gradient descent (DPSGD) (Song et al., 2013; Bassily et al., 2014; Abadi et al., 2016)[4]. This optimization method serves as a drop-in replacement for SGD, augmenting it with the addition of per-example gradient clipping and Gaussian noise addition steps. These two steps serve to limit and mask the contribution of a single example. Two key points to note are that a) per-example gradient clipping incurs significant computational and memory overheads in most implementations, and b) noise introduced due to privacy grows as the square-root of the number of model parameters. With this tool in place, the most basic fine-tuning strategy is to train all parameters using DPSGD.

2.2. Reparametrized Gradient Perturbation.

To mitigate the limitations of DPSGD, a recent work of Yu et al. (2021b) introduced an elegant method called *reparametrized gradient perturbation* (RGP). RGP exploits the implicit low-rank structure in the gradient updates of SGD to substantially improve upon DPSGD. Specifically, they reparametrize each layer's weight matrix W into $LR + \tilde{W}$, where L and R are low-rank gradient-carrier matrices and $\tilde{W}$ is the residual weight. The authors show that one can obtain a low-dimensional projection of W's gradient by taking gradients only of the low-rank matrices L and R (and not the high-rank $\tilde{W}$). Privacy is introduced by clipping and noising these low-dimensional gradients of L and R. While this low-dimensional projection loses some of the signal in W's gradient, it turns out to contain enough to still achieve high accuracy. At the same time, the low-dimensional gradients alleviate the aforementioned issues related to privatization, significantly reducing the memory consumption and noise introduced. Although RGP uses a low-rank update at each step, we empirically verify that its accumulated update is not of low stable rank and hence can not be compressed into small plug-in modules. Possible reasons include: 1) the low-rank subspaces of RGP are different at different updates; 2) the accumulated update of RGP contains all the added noises, which are of high stable rank.

[4]Following Abadi et al. (2016), our implementation of DP-SGD uses shuffle data instead of Poisson sampling to enforce stochasticity. However, the privacy analysis in Abadi et al. (2016) uses Poisson sampling. Shuffle data is easier to implement but using it would create a mild discrepancy with the privacy analysis.

3. Our Approach

3.1. **A Meta-framework.** We introduce our approach as a meta-framework for private deep learning, which abstracts the key principles of recent fine-tuning methods.

Suppose $f(W_{\rm PT}; x)$ is a pre-trained model where $W_{\rm PT}$ are the pre-trained weights and x is any input. We create a new fine-tuning model

$$f_{\rm FT}(W_{\rm PT}, \theta; x) \tag{3.1}$$

which incorporates additional trainable parameters θ, where $\dim(\theta) \ll \dim(W_{\rm PT})$. That is, the number of new parameters in θ is a small fraction of the original number of parameters in the pre-trained weights $W_{\rm PT}$. Fine-tuning is done by running DPSGD on the additional parameters θ, while freezing the weights of pre-trained model $W_{\rm PT}$. The new parameters are initialized to θ_0 such that

$$f_{\rm FT}(W_{\rm PT}, \theta_0; x) = f(W_{\rm PT}; x). \tag{3.2}$$

The initialization condition (3.2) is very important, as it ensures that fine-tuning starts at the pre-trained model and improves it by modifying the parameters θ. Most fine-tuning methods are additive and have the following special form:

$$f_{\rm FT}(W_{\rm PT}, \theta; x) = f(W_{\rm PT} + \pi(\theta); x), \tag{3.3}$$

i.e., they modify the pre-trained weights by adding a correction term $\pi(\theta)$ parametrized by θ.

Recent work in the non-private literature has described concrete instantiations of this framework (Houlsby et al., 2019; Mahabadi et al., 2021; Hu et al., 2022), which (crucially) are effective when $\dim(\theta) \ll \dim(W_{\rm PT})$. In the non-private setting, such reparametrizations are useful for reducing the computation and memory required for fine-tuning, and enable lightweight and plug-in modifications to the base model for different downstream tasks. At the same time, they maintain (or sometimes surpass) the accuracy achieved by full fine-tuning.

We give some intuition as to why parameter-efficient methods could be more effective for private fine-tuning especially when private datasets are small. For simplicity, we assume that the fine-tuning method is additive as in (3.3), such that the fine-tuned weights $W_{\rm FT} = W_{\rm PT} + \pi(\theta)$. We can imagine that $W_{\rm FT}$ lies on a manifold passing through $W_{\rm PT}$ of very small dimension (equal to the dimension of θ) compared to the dimension of $W_{\rm PT}$. Even if the parameters θ are very noisy due to the noise added during DPSGD, we will always stay in this manifold. In particular, we are not disturbing the pre-trained weights in most directions (those orthogonal to the manifold near $W_{\rm PT}$). If we run DPSGD on all the weights instead, then we add noise in all directions, thus potentially unlearning the knowledge learned during pre-training, especially in low data regimes; see the discussion in Section 4.3 for more on this.

Besides substantial gains in the accuracy, the above method of reparametrization has several other advantages:

- A single pre-trained model such as BERT or GPT is generally applied to hundreds of downstream tasks via fine-tuning. Private fine-tuning using previous methods requires updating *all* parameters and storing a different copy of the fine-tuned model per task. This creates substantial overheads for storing and deploying, which can be very expensive in practice. On the other hand, the reparametrization (3.1) means that we only need to

store a single pre-trained model that can be shared across many downstream tasks. Each downstream task requires only a small number of new parameters that can be plugged in.
- Differentially private training requires computing and storing per-example gradients, which increases the memory footprint. In our approach, however, learning is done in a much lower dimension, hence saving on the memory cost as compared to prior works.
- Finally, we expect that (3.1) also gives a more communication-efficient method of fine-tuning in distributed settings such as federated learning, due to the significantly smaller number of parameters learned during fine-tuning.

3.2. **Instantiating the Meta-framework.** In this section, we discuss a few ways to instantiate our meta-framework. This list is non-exhaustive, but it covers the methods we employ in our experiments.

3.2.1. *Fine-tuning via Low-Rank Adaptation.* Low-Rank Adaptation (LoRA) (Hu et al., 2022) is an additive fine-tuning scheme as defined in (3.3). For each dense weight matrix W^i_{PT} of size $a \times b$ in the pre-trained network, we add a low-rank correction term $L^i R^i$, i.e.,

$$W^i = W^i_{\mathrm{PT}} + L^i R^i, \tag{3.4}$$

where $L^i \in \mathbb{R}^{a\times r}, R^i \in \mathbb{R}^{r\times b}$ are new trainable parameters. Hu et al. (2022) apply this reparameterization only to query and value weights in the Transformer attention blocks, and freeze all other weights. The rank r is typically chosen to be small, e.g., $r = 4, 16, 64$. Since most parameters in Transformer architectures are dense weight matrices, choosing a small r results in a nearly square-root reduction in the number of parameters.

3.2.2. *Fine-tuning via Adapters.* Houlsby et al. (2019) propose adapter-based fine-tuning, in which we modify the architecture of the pre-trained model by adding new "adapter" layers after each attention and feed-forward layer. Adapter layers are bottleneck layers with residual connections. Specifically, given an input x, an adapter layer A performs

$$A(x) = U(\tau(D(x))) + x, \tag{3.5}$$

where U is an up-projection affine linear map, D is a down-projection affine linear map, and τ is a non-linear activation function such as the Gaussian error Linear Unit (GeLU) (Hendrycks and Gimpel, 2016). If x has dimension d, then $U \in \mathbb{R}^{d\times r}, D \in \mathbb{R}^{r\times d}$ for some $r \ll d$. Thus, the number of introduced parameters is significantly less than the number of parameters in the pre-trained model. When fine-tuning, the parameters of the original model are frozen, and only parameters of the adapter layers (without bias terms), as well as layer normalizations, are modified. Note that fine-tuning with adapters is not an additive fine-tuning framework as in (3.3), but is captured by the broader framework in (3.1).

TABLE 3. Memory and speed comparison for RoBERTa-Large.

Method	Memory (GB)	Speed (seconds per epoch)
Full fine-tuning (DPSGD)	27.9	715
RGP	9.1	296
DP LoRA	6.1	271

The rank is chosen as $r = 16$ for RGP and LoRA. The speed is measured by the wall-clock time for training one epoch of the SST-2 dataset on a single Tesla V100 GPU with gradient accumulation for batch size 2000. The memory column shows the memory cost fo stroing both the model and gradients.

3.2.3. *Fine-tuning via Compacter.* The recent work of Mahabadi et al. (2021) introduces Compacters (Compact adapters), a method which further improves the parameter efficiency of adapters. This is done by replacing the dense matrices in the up-projection U and down-projection D by tensor products of smaller matrices, thus reducing the number of trainable parameters. Specifically, they replace the dense matrix M_ℓ in the adapter layer ℓ by a low-rank parameterized hypercomplex multiplication (LPHM) layer, i.e., each dense matrix $M_\ell \in \mathbb{R}^{a\times b}$ is expressed as

$$M_\ell = \sum_{i=1}^{n} A_i \otimes \left(S_i^\ell T_i^\ell\right) \tag{3.6}$$

where $A_i \in \mathbb{R}^{n\times n}, S_i^\ell \in \mathbb{R}^{a/n\times k}, T_i^\ell \in \mathbb{R}^{k\times b/n}$ and $\otimes$ is the matrix Kronecker product. Note the matrices A_i are not indexed by the layer ℓ because these matrices are shared among all the adapter layers. Since each adapter layers has two dense matrices (one for up-projection and one for down-projection), if there are L adapter layers, this reduces the number of parameters from $L(2ab)$ to $L(2(a+b)k) + n^3$. In practice, a and b are chosen to be either the model dimension d or the intermediate representation dimension r in the adapters, n is typically chosen to be a small constant such as $n = 2, 4, 8, 12$ and k is chosen to be 1.

3.2.4. *Why Does Parameter-Efficient Tuning Work?* Theoretical explanation of the success of parameter-efficient fine-tuning methods is an active area of research in deep learning. Indeed, since trends have consistently shown that model accuracy increases with size, how can one achieve competitive accuracy while fine-tuning less than 1% of the parameters? One popular hypothesis is *intrinsic dimensionality* (Li et al., 2018), which posits that the minimum number of parameters needed to train a machine learning model may be much less than the total number of model parameters. Aghajanyan et al. (2021) explore this hypothesis in the context of fine-tuning LLMs, showing that one can achieve most of their accuracy by training only a very small number of parameters (chosen via a random projection). Perhaps surprisingly, they find that as *the model size increases, intrinsic dimension decreases* in the limit exhibiting zero-shot learning. While we did not explore this hypothesis in the context of DP due to computational restrictions, we believe it may be an interesting lens through which one can understand the effectiveness of private parameter-efficient fine-tuning.

3.3. **Comparision with Baseline Algorithms.** We highlight some key algorithmic differences between our proposed methods and the baselines of full fine-tuning and RGP.

- DPSGD and RGP both require updating all parameters of the pre-trained model, whereas our proposed methods update only a tiny fraction (between 0.05% and 1%). The rightmost columns of Tables 4 and 5 list the number of parameters trained by these algorithms.
- RGP performs a low-rank decomposition of weight matrices which is very similar to LoRA, though there are subtle differences. Recall that in RGP, at the beginning of each iteration t, the historical weight matrix W_{t-1} is decomposed to find a low-rank product LR. The gradients computed on L and R are then projected back to the full parameter space to perform the descent step. Hence, RGP does not keep the pre-trained weights frozen during the learning process.

  LoRA can be viewed as a simplification of RGP. LoRA reparametrizes $W_{\text{FT}} := W_{\text{PT}} + LR$, where the pre-trained weight matrix W_{PT} is frozen during training. Hence, compared to RGP, LoRA eliminates the decomposition and the projection to the full parameter space at each iteration, simplifying the implementation and reducing the running time and memory cost. This is summarized in Table 3. We observe that DP LoRA reduces the memory cost by about 33% and the training speed by 8%. As we will see, this simplification also results in improved utility.
- Neither full fine-tuning nor RGP fall into our meta-framework described by (3.1). Thus, if a pre-trained model is to be applied to several downstream tasks, one must store a separate set of weights for each task, incurring a significant memory cost and losing the plug-in functionality. In contrast, our methods are much more lightweight.

4. Experiments

We experimentally evaluate our methods for DP fine-tuning to demonstrate their utility, privacy, and parameter-efficiency. We investigate both language understanding and text generation tasks, using RoBERTa and GPT-2 models, to establish that our techniques are applicable to a variety of tasks and model architectures.[5]

4.1. **Fine-Tuning for Language Understanding Tasks.** We first compare our methods with state-of-the-art fine-tuning algorithms using models from the BERT family, which was used in the prior work (Yu et al., 2021b). Specifically, we use RoBERTa models (Liu et al., 2019), which are pre-trained on public data collected from the web. RoBERTa-Base has 125M parameters and RoBERTa-Large has 355M parameters. We fine-tune the pre-trained models on four tasks: MNLI, QQP, QNLI and SST-2 from the GLUE benchmark (Wang et al., 2019), following Yu et al. (2021b).

Implementation Details: For fine-tuning with adapters, we may choose the intermediate representation dimension r, shared across all adapter layers. Similarly, for fine-tuning with Compacter, we can choose both the intermediate representation dimension r and the Kronecker product kernel dimension n in (3.6). For LoRA fine-tuning, we add bottleneck branches for both the attention layers and the feedforward layers, which differs slightly from the addition of bottleneck branches for only the W_q and W_v matrices of the attention layers as done by Hu et al. (2022). Given the same bottleneck representation dimension r in (3.4),

[5]Code for our experiments is available at https://github.com/huseyinatahaninan/Differentially-Private-Fine-tuning-of-Language-Models.

TABLE 4. Accuracy for fine-tuning downstream tasks with RoBERTa-Base (in %)

Method		MNLI	SST-2	QQP	QNLI	Avg.	Trained params
Full[6]	w/o DP	87.6	94.8	91.9	92.8	91.8	100%
	DP	53.1	82.6	74.4	63.9	68.5	
LoRA	w/o DP	87.5	95.1	90.8	93.3	91.7	0.24%
RGP[7]	DP	80.1	91.6	85.5	87.2	86.1	100%
Adapter	DP	83.4	**92.5**	85.6	**87.5**	**87.3**	1.40% ($r = 48$)
Compacter	DP	82.6	92.3	84.7	85.1	86.2	0.055% ($r = 96$, $n = 8$)
LoRA	DP	**83.5**	92.2	**85.7**	87.3	87.2	1.86% ($r = 16$)

The privacy parameters are $\epsilon = 6.7$, and $\delta = 1 \times 10^{-5}$ for SST-2 and QNLI and 1×10^{-6} for MNLI and QQP. Bold indicates the best accuracy with DP. Numbers for non-private fine-tuning are from Liu et al. (2019).

TABLE 5. Accuracy for fine-tuning downstream tasks with RoBERTa-Large (in %)

Method		MNLI	SST-2	QQP	QNLI	Avg.	Trained params
Full	w/o DP	90.2	96.4	92.2	94.7	93.4	100%
LoRA	w/o DP	90.6	96.2	91.6	94.9	93.3	0.23%
RGP	DP	86.1	93.0	86.7	90.0	88.9	100%
Adapter	DP	87.7	93.9	86.3	90.7	89.7	1.31% ($r = 48$)
Compacter	DP	87.5	94.2	86.2	90.2	89.5	0.053% ($r = 96$, $n = 8$)
LoRA	DP	**87.8**	**95.3**	**87.4**	**90.8**	**90.3**	1.74% ($r = 16$)

The privacy parameters are $\epsilon = 6.7$, and $\delta = 1 \times 10^{-5}$ for SST-2 and QNLI and $\delta = 1 \times 10^{-6}$ for MNLI and QQP. Bold indicates the best accuracy with DP. Numbers for non-private fine-tuning are from Liu et al. (2019).

our new implementation uses twice as many trainable parameters as the original paper, and achieves some improvements for learning with DP. We perform privacy accounting using the PRV Accountant from Gopi et al. (2021), which currently provides the tightest bounds.

Hyperparameter choice: Given the large number of hyperparameter choices, e.g., the intermediate representation dimension, learning rate, weight decay, privacy delta, and model size, an exhaustive grid search over all hyperparameters is expensive, due to the model sizes. Our hyperparameter choices are informed by prior work and are as follows. For privacy parameters, we use $\delta = 1 \times 10^{-5}$ for SST-2 and QNLI and $\delta = 1 \times 10^{-6}$ for MNLI and QQP due to their dataset sizes, and use noise multipliers $0.92, 0.83, 0.66$ and 0.65 for SST-2, QNLI, QQP and MNLI, respectively, which is the same as Yu et al. (2021b)[8]. The clipping threshold of per-example gradients is 10 for all methods. For adapters and Compacter, we follow suggestions in the original papers and choose r from a set $\{16, 48, 96\}$ and n from a set $\{4, 8, 12\}$. For LoRA, we choose the best-performing rank r from the set $\{4, 16, 48, 64\}$. The best performing hyperparameters are noted in Tables 4 and 5. We use batch size 2000

[6]A concurrent work by Li et al. (2022b) shows DP full fine-tuning can be significantly improved with carefully chosen hyperparameters. See Section 4.3 for a detailed discussion. We note that in this paper we use the same hyperparameters for all the algorithms.

[7]We report RoBERTa-Base numbers from `https://github.com/dayu11/Differentially-Private-Deep-Learning`, by the authors of Yu et al. (2021b). Though the paper itself only reports results on BERT-Base, we cite their paper to also reference the RoBERTa numbers.

[8]In Appendix A, we evaluate the proposed framework with various choices of privacy parameters.

and train with half-precision for 20 epochs. We use the optimizer AdamW (Loshchilov and Hutter, 2019) with weight decay 1e-2 and search over four learning rates {5e-4, 1e-3, 2e-3, 5e-3}. In Appendix B, we show the proposed algorithms perform well for a wide range of hyperparameters. Due to computational constraints, we cannot repeat all the experiments with different random seeds. To assess the influence of randomness, we fine-tuned the RoBERTa-Large model on the SST-2 dataset with five different random seeds. We use LoRA and choose the best performing hyperparameters. The test accuracy of five runs is 95.5%, 95.5%, 95.3%, 95.2%, and 94.9%. The average accuracy is 95.28% and the standard deviation of test accuracy is 0.22%. The variation in test accuracy is small and does not impact our main findings.

Results: We report the prediction accuracy on four tasks in Tables 4 and 5[9]. Our experiments using RoBERTa-Base serve as a direct comparison to the work of Yu et al. (2021b) who only trained the base model, whereas RoBERTa-Large experiments demonstrate the significance of using larger models. We could not report the numbers for full fine-tuning using DPSGD on RoBERTa-Large due to running time and memory costs; see the discussion at the end of this section. We summarize our key findings:

- On *all* datasets, our methods achieve the best accuracy while training a only tiny fraction of parameters; larger models give significant improvements.
- Noticeable improvements in the privacy parameter ϵ versus Yu et al. (2021b) are primarily due to new privacy accountants based on Fourier-based numerical composition (Koskela et al., 2020, 2021; Gopi et al., 2021); we use the PRV Accountant from Gopi et al. (2021) since it is the most efficient.
- Private adapters provide the best average performance for RoBERTa-Base, whereas LoRA outperforms all other methods for RoBERTa-Large.

4.2. **Fine-tuning for Natural Language Generation (NLG).** Next, we study private fine-tuning for text generation problems using the GPT-2 series of models on the End-2-End (E2E) NLG challenge (Novikova et al., 2017) and DART (Nan et al., 2021), two primary benchmarks used in recent works on non-private fine-tuning (Hu et al., 2022; Li and Liang, 2021). We use GPT-2-Small (117M parameters), GPT-2-Medium (345M parameters), GPT-2-Large (774M parameters), and GPT-2-XL (1.5B parameters).[10] To the best of our knowledge, we are the first to privately fine-tune for E2E-DART or fine-tune GPT-2-XL. The purpose of this section is not to evaluate various fine-tuning algorithms, but to show that private fine-tuning is competitive with non-private fine-tuning for text generation problems. Due to the high cost of training, we report experimental results only for fine-tuning (private and non-private) with LoRA. We think that all fine-tuning methods in this paper should achieve comparable accuracy.

E2E NLG challenge: The E2E dataset was introduced by Novikova et al. (2017), and contains template-like information in the restaurant domain to be mapped to natural language with end-to-end training. The dataset consists of 42K training samples, 4.6K validation samples, and 4.6K test samples.

[9]Since the original appearance of this paper, several subsequent works also conduct experiments on the GLUE benchmark (Bu et al., 2022, 2023; He et al., 2023). In Appendix D, we compare DP LoRA fine-tuning with the results in two representative subsequent works. The results in the subsequent works further confirm the main findings in this paper.

[10]`https://huggingface.co/transformers/model_doc/gpt2.html`.

TABLE 6. Metrics on the E2E NLG task. Non-DP results from Hu et al. (2022), except for GPT-2-XL, which was not reported in the paper. We ran GPT-2-XL with hyperparameters presented in Hu et al. (2022). Bold indicates the best accuracy with DP. DP parameters are ($\epsilon = 6.0, \delta =$ 1e-5). Val perp stands for validation perplexity.

Method	Val perp	BLEU	NIST	MET	ROUGE-L	CIDEr
GPT-2-Small + DP	4.51	63.8	7.19	39.5	67.5	1.87
GPT-2-Medium + DP	4.02	65.5	8.45	42.7	67.9	2.23
GPT-2-Large + DP	3.87	**66.7**	**8.63**	**44.0**	67.8	**2.33**
GPT-2-XL + DP	**3.79**	66.1	8.53	43.0	**68.1**	2.28
GPT-2-Medium	3.19	70.4	8.85	46.8	71.8	2.53
GPT-2-Large	3.06	70.4	8.89	46.8	72.0	2.47
GPT-2-XL	3.01	69.4	8.78	46.2	71.5	2.49

DART: DART was introduced as an open-domain data-to-text dataset by Nan et al. (2021). The dataset consists of 62K training samples, 6.9K validation samples, and 12K test samples. In comparison to E2E, the dataset is larger and the task is more challenging.

We use standard metrics such as BLEU, ROUGE-L, etc., used in (Hu et al., 2022) for measuring the quality of predictions.

Hyperparameter choice: For LoRA, we choose the bottleneck rank $r = 4$ in (3.4) and fine-tune W_q and W_v matrices of the attention layers as in the original paper. The fractions of trainable parameters are 0.12%, 0.11%, 0.09%, and 0.07% for GPT-2-Small, GPT-2-Medium, GPT-2-Large, and GPT-2-XL, respectively. We optimize using AdamW with learning rate 4e-4, weight decay 1e-2 and train our models for 20 epochs. We use batch size 128 for the experiments on E2E and batch size 256 for the experiments on DART. We take the gradient clipping parameter to be 1.0 and the noise multiplier to be 0.6 for the accountant in Gopi et al. (2021), achieving $\epsilon = 6.0, \delta =$1e-5 on E2E and $\epsilon = 6.8, \delta =$1e-5 on DART.

Results: The results of our experiments are summarized in the Table 6 and 7, which reiterate the main themes of our work: private fine-tuning with parameter-efficient approaches perform close to their non-private counterparts and show consistent improvement in the utility as model size increases. Note that on E2E dataset, although the validation perplexity improves as the model becomes larger, the metrics seem to saturate going from large to XL for both private and non-private cases. On the other hand, for DART dataset both validation perplexity and the metric improve as the model size increases.

4.3. **How Bad is DP Full Fine-tuning?** A concurrent work by Li et al. (2022b) shows that the performance of DP full fine-tuning is sensitive to hyperparameter choices, and that using a larger batch size and training with full-precision significantly improves the performance of full fine-tuning. We note that Li et al. (2022b) also propose *Ghost Clipping*, which is a novel clipping method that makes DP full fine-tuning more efficient. Without such techniques, it is impossible to do extensive hyperparameter tuning for DP full fine-tuning with limited compute. Due to compute limitations, we only did limited hyperparameter search and the hyperparameter sweep is the same for all the algorithms. This suggests that parameter-efficient methods may be more robust to the choice of hyperparameters. With

TABLE 7. Metrics on the DART dataset. Non-DP results from Hu et al. (2022), except for GPT-2-XL, which was not reported in the paper. We ran GPT-2-XL with hyperparameters presented in Hu et al. (2022). Bold indicates the best accuracy with DP. DP parameters are ($\epsilon = 6.8, \delta =$ 1e-5). Val perp stands for validation perplexity. Unlike all other metrics, lower measurements in the TER metric indicate better performance of the model.

Method	Val perp	BLEU	MET	TER
GPT-2-Small + DP	3.82	38.5	0.34	0.53
GPT-2-Medium + DP	3.30	42.0	0.36	0.51
GPT-2-Large + DP	3.10	43.1	0.36	**0.5**
GPT-2-XL + DP	**3.00**	**43.8**	**0.37**	**0.5**
GPT-2-Medium	2.67	47.1	0.39	0.46
GPT-2-Large	2.89	47.5	0.39	0.45
GPT-2-XL	2.83	48.1	0.39	0.46

the hyperparameters and setup of Li et al. (2022b) , we are able to reproduce their results. Moreover, we also get improvements around 1% for our algorithms. We report the new findings in Appendix C. In the new experiments, the gap between DP full fine-tuning and parameter-efficient methods on GLUE tasks is within 3% on average, which is much smaller than the gap in Table 4.

5. Related Work

5.1. **More on DP learning:** Some work studies private language models on more traditional architectures such as LSTMs (Hochreiter and Schmidhuber, 1997), either training with DPSGD (McMahan et al., 2018; Carlini et al., 2019) or related heuristics (Ramaswamy et al., 2020). Though pre-training on public data is suggested (McMahan et al., 2018), public data appears to only be used in one of these works for honest hyperparameter selection (Ramaswamy et al., 2020). A few more recent works consider training LLMs with DP. Anil et al. (2022) privately train BERT-Large from scratch, compared to our work which focuses on private fine-tuning. (Hoory et al., 2021; Basu et al., 2021) perform private full fine-tuning of BERT models. Hoory et al. (2021) achieve accuracy which is comparable to the non-private model, but additionally supplement the public pre-training data with additional domain-relevant material, while we use off-the-shelf pre-trained models. Basu et al. (2021) observe significant drops in utility, compared to our parameter-efficient methods which do not. While Kerrigan et al. (2020) consider public pre-training and private fine-tuning, their experiments are on much smaller architectures (i.e., feedforward networks with three hidden layers). A simultaneous work of Ginart et al. (2022) investigates private *prediction* (rather than learning) for next-token prediction. A subsequent work by Senge et al. (2021) also investigates the effect of private fine-tuning on various NLP tasks.

Our investigation fits more broadly into a line of work employing public data for private data analysis. Some works on image classification consider pre-training on a large public dataset and fine-tuning on a smaller private dataset (Abadi et al., 2016; Papernot et al., 2019; Tramèr and Boneh, 2021; Luo et al., 2021). In particular, Luo et al. (2021) investigate the role of parameter efficiency in private fine-tuning ResNet models, and propose strategies

to choose which parameters to fine-tune. One line of work uses unlabeled public data to train a student model (Papernot et al., 2017, 2018; Bassily et al., 2018), including one work simultaneous to our own for natural language generation Tian et al. (2022). Another recent idea uses a small amount of public data to identify a lower-dimensional subspace of the gradients in which to perform private descent (Zhou et al., 2021; Yu et al., 2021a; Kairouz et al., 2021). A simultaneous work of Amid et al. (2022) uses public data in the mirror map for a private mirror descent algorithm. Finally, other works (both theoretical and experimental) investigate the role of public data in private query release, synthetic data generation, and prediction (Ji and Elkan, 2013; Beimel et al., 2016; Alon et al., 2019; Nandi and Bassily, 2020; Bassily et al., 2020a,b; Liu et al., 2021).

Since the initial appearance of our paper, private fine-tuning (of language models and beyond) has become perhaps the standard paradigm for doing private machine learning in many settings (Bu et al., 2022; Wu et al., 2024; Du et al., 2023; Pelikan et al., 2023). He et al. (2023) explored larger-scale settings, including private fine-tuning of GPT-3. Other works have investigated private fine-tuning of language models with the addition of model compression (Mireshghallah et al., 2022), and for synthetic data generation with a focus on the quality of the downstream data (Yue et al., 2022; Kurakin et al., 2023). Turning to vision models, Golatkar et al. (2022); De et al. (2022); Mehta et al. (2023); Berrada et al. (2023) show that privately fine-tuning ResNets and Vision Transformers can provide high utility on benchmarks including CIFAR-10 and ImageNet, the latter of which was previously thought to be intractable under privacy constraints. Impressive qualitative improvements have also been shown for generative models (Ghalebikesabi et al., 2023; Harder et al., 2023; Wu et al., 2023). Some works (Li et al., 2022a; Ganesh et al., 2023) try to provide explanations as to why public pre-training is so valuable for private ML. See Cummings et al. (2024) (particularly Section 3.1) for more coverage of recent work using public data in private ML.

Some subsequent works have also been critical of directions we explore in this paper. Brown et al. (2021) highlight the difficulty of capturing the general concept of privacy via the semantics of differential privacy. Tramèr et al. (2022) argue that most private ML works which employ public pre-training are simplistic in their treatment of privacy semantics (which reductively treat data as either public or private) and limited in their choice of evaluation datasets.

5.2. **More on Fine-tuning:** There exist other parameter-efficient tuning methods which we did not evaluate in our work. Some of these include random subspace projection (exploiting intrinsic dimensionality (Li et al., 2018; Aghajanyan et al., 2021)), prefix and prompt tuning (Li and Liang, 2021; Lester et al., 2021), tuning only biases (Cai et al., 2020; Ben Zaken et al., 2022), and other architecture variants including Adapters (Pfeiffer et al., 2021; Rücklé et al., 2021). Other works investigate lightweight methods for adapting language models to different tasks (e.g., Dathathri et al. (2020)). An interesting direction for future work is to see whether parameter-efficient tuning approaches specifically designed for the private setting can achieve higher utility. We also mention zero-shot learning, in which no task-specific dataset is required and thus perfect privacy is achieved. Currently, zero-shot approaches achieve low utility compared to fine-tuning, though it is possible that future models may narrow this gap.

6. Conclusion

So far, DP deep learning has focused on training models from scratch. The spectacular success of transfer learning in real-world applications, however, shows that private fine-tuning is an equally pertinent problem to study and deserves more attention. We show that by combining recent advances in NLP, parameter-efficiency, privacy accounting, and using larger models, one can privately fine-tune models whose utility approaches that of non-private models. We hope our work inspires more study on the core problem of private fine-tuning, which we believe to be a central direction for research in private machine learning, leading to more interaction between the LLM and DP communities.

Acknowledgments

The authors would like to thank Rabeeh Karimi Mahabadi for sharing hyperparameters based on experiments in Mahabadi et al. (2021) and Xuechen Li for sharing the experimental setup and many suggestions about mixed-precision training. Janardhan Kulkarni would like to thank Edward Hu for sharing ideas on fine-tuning.

References

M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM Conference on Computer and Communications Security*, CCS '16, pages 308–318, New York, NY, USA, 2016. ACM. https://doi.org/10.1145/2976749.2978318.

A. Aghajanyan, L. Zettlemoyer, and S. Gupta. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, ACL/IJCNLP (1) '21, pages 7319–7328. Association for Computational Linguistics, 2021. https://doi.org/10.18653/v1/2021.acl-long.568.

N. Alon, R. Bassily, and S. Moran. Limits of private learning with access to public data. In *Advances in Neural Information Processing Systems 32*, NeurIPS '19, pages 10342–10352. Curran Associates, Inc., 2019. https://proceedings.neurips.cc/paper_files/paper/2019/file/9a6a1aaafe73c572b7374828b03a1881-Paper.pdf.

E. Amid, A. Ganesh, R. Mathews, S. Ramaswamy, S. Song, T. Steinke, V. M. Suriyakumar, O. Thakkar, and A. Thakurta. Public data-assisted mirror descent for private model training. In *Proceedings of the 39th International Conference on Machine Learning*, ICML '22. JMLR, Inc., 2022. https://proceedings.mlr.press/v162/amid22a.html.

R. Anil, B. Ghazi, V. Gupta, R. Kumar, and P. Manurangsi. Large-scale differentially private BERT. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, Findings of EMNLP '22, pages 6481–6491. Morgan Kaufmann Publishers Inc., 2022. https://doi.org/10.18653/v1/2022.findings-emnlp.484.

R. Bassily, A. Smith, and A. Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *Proceedings of the 55th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '14, pages 464–473, Washington, DC, USA, 2014. IEEE Computer Society. https://doi.org/10.1109/FOCS.2014.56.

R. Bassily, O. Thakkar, and A. Guha Thakurta. Model-agnostic private learning. In *Advances in Neural Information Processing Systems 31*, NeurIPS '18, pages 7102–7112. Curran Associates, Inc., 2018. https://papers.nips.cc/paper_files/paper/2018/hash/aa97d584861474f4097cf13ccb5325da-Abstract.html.

R. Bassily, A. Cheu, S. Moran, A. Nikolov, J. Ullman, and S. Wu. Private query release assisted by public data. In *Proceedings of the 37th International Conference on Machine Learning*, ICML '20, pages 695–703. JMLR, Inc., 2020a. https://proceedings.mlr.press/v119/bassily20a.html.

R. Bassily, S. Moran, and A. Nandi. Learning from mixtures of private and public populations. In *Advances in Neural Information Processing Systems 33*, NeurIPS '20, pages 2947–2957. Curran Associates, Inc., 2020b. https://proceedings.neurips.cc/paper/2020/hash/1ee942c6b182d0f041a2312947385b23-Abstract.html.

P. Basu, T. S. Roy, R. Naidu, Z. Muftuoglu, S. Singh, and F. Mireshghallah. Benchmarking differential privacy and federated learning for BERT models. *arXiv preprint arXiv:2106.13973*, 2021. https://doi.org/10.48550/arXiv.2106.13973.

A. Beimel, K. Nissim, and U. Stemmer. Private learning and sanitization: Pure vs. approximate differential privacy. *Theory of Computing*, 12(1):1–61, 2016. https://doi.org/10.1007/978-3-642-40328-6_26.

E. Ben Zaken, S. Ravfogel, and Y. Goldberg. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, ACL '22, pages 1–9. Association for Computational Linguistics, 2022. https://doi.org/10.18653/v1/2022.acl-short.1.

L. Berrada, S. De, J. H. Shen, J. Hayes, R. Stanforth, D. Stutz, P. Kohli, S. L. Smith, and B. Balle. Unlocking accuracy and fairness in differentially private image classification. *arXiv preprint arXiv:2308.10888*, 2023. https://doi.org/10.48550/arXiv.2308.10888.

G. Brown, M. Bun, V. Feldman, A. Smith, and K. Talwar. When is memorization of irrelevant training data necessary for high-accuracy learning? In *Proceedings of the 53nd Annual ACM Symposium on the Theory of Computing*, STOC '21, pages 123–132, New York, NY, USA, 2021. ACM. https://doi.org/10.1145/3406325.3451131.

T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33*, NeurIPS '20. Curran Associates, Inc., 2020. https://papers.nips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

Z. Bu, Y.-X. Wang, S. Zha, and G. Karypis. Differentially private bias-term only fine-tuning of foundation models. *arXiv preprint arXiv:2210.00036*, 2022. https://doi.org/10.48550/arXiv.2210.00036.

Z. Bu, Y.-X. Wang, S. Zha, and G. Karypis. Differentially private optimization on large model at small cost. In *International Conference on Machine Learning*, volume 202, pages 3192–3218. PMLR, 2023. https://proceedings.mlr.press/v202/bu23a.html.

M. Bun, J. Ullman, and S. Vadhan. Fingerprinting codes and the price of approximate differential privacy. In *Proceedings of the 46th Annual ACM Symposium on the Theory of Computing*, STOC '14, pages 1–10, New York, NY, USA, 2014. ACM. https://doi.org/10.1145/2591796.2591877.

H. Cai, C. Gan, L. Zhu, and S. Han. TinyTL: Reduce memory, not parameters for efficient on-device learning. In *Advances in Neural Information Processing Systems 33*, NeurIPS '20, pages 11285–11297. Curran Associates, Inc., 2020. https://proceedings.neurips.cc/paper/2020/hash/81f7acabd411274fcf65ce2070ed568a-Abstract.html.

N. Carlini, C. Liu, Ú. Erlingsson, J. Kos, and D. Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th USENIX Security Symposium*, USENIX Security '19, pages 267–284. USENIX Association, 2019. https://www.usenix.org/conference/usenixsecurity19/presentation/carlini.

N. Carlini, F. Tramèr, E. Wallace, M. Jagielski, A. Herbert-Voss, K. Lee, A. Roberts, T. Brown, D. Song, U. Erlingsson, A. Oprea, and C. Raffel. Extracting training data from large language models. In *30th USENIX Security Symposium*, USENIX Security '21, pages 2633–2650. USENIX Association, 2021. https://www.usenix.org/conference/usenixsecurity21/presentation/carlini-extracting.

R. Cummings, D. Desfontaines, D. Evans, R. Geambasu, M. Jagielski, Y. Huang, P. Kairouz, G. Kamath, S. Oh, O. Ohrimenko, N. Papernot, R. Rogers, M. Shen, S. Song, W. Su, A. Terzis, A. Thakurta, S. Vassilvitskii, Y.-X. Wang, L. Xiong, S. Yekhanin, D. Yu, H. Zhang, and W. Zhang. Advancing differential privacy: Where we are now and future directions for real-world deployment. *Harvard Data Science Review*, 2024. https://doi.org/10.1162/99608f92.d3197524.

S. Dathathri, A. Madotto, J. Lan, J. Hung, E. Frank, P. Molino, J. Yosinski, and R. Liu. Plug and play language models: A simple approach to controlled text generation. In *Proceedings of the 8th International Conference on Learning Representations*, ICLR '20, 2020. https://openreview.net/forum?id=H1edEyBKDS.

S. De, L. Berrada, J. Hayes, S. L. Smith, and B. Balle. Unlocking high-accuracy differentially private image classification through scale. *arXiv preprint arXiv:2204.13650*, 2022. https://doi.org/10.48550/arXiv.2204.13650.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. 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, Volume 1 (Long and Short Papers)*, NAACL-HLT '19, pages 4171–4186, 2019. https://doi.org/10.18653/v1/N19-1423.

M. Du, X. Yue, S. S. Chow, T. Wang, C. Huang, and H. Sun. Dp-forward: Fine-tuning and inference on language models with differential privacy in forward pass. *arXiv preprint arXiv:2309.06746*, 2023. https://doi.org/10.1145/3576915.3616592.

C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor. Our data, ourselves: Privacy via distributed noise generation. In *Proceedings of the 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, EUROCRYPT '06, pages 486–503, Berlin, Heidelberg, 2006a. Springer. https://doi.org/https://doi.org/10.1007/11761679_29.

C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the 3rd Conference on Theory of Cryptography*, TCC '06, pages 265–284, Berlin, Heidelberg, 2006b. Springer. https://doi.org/https://doi.org/10.1007/11681878_14.

V. Feldman. Does learning require memorization? a short tale about a long tail. In *Proceedings of the 52nd Annual ACM Symposium on the Theory of Computing*, STOC '20, pages 954–959, New York, NY, USA, 2020. ACM. https://doi.org/10.1145/3357713.3384290.

A. Ganesh, M. Haghifam, M. Nasr, S. Oh, T. Steinke, O. Thakkar, A. G. Thakurta, and L. Wang. Why is public pretraining necessary for private model training? In *Proceedings of the 40th International Conference on Machine Learning*, ICML '23, pages 10611–10627. JMLR, Inc., 2023. https://dl.acm.org/doi/10.5555/3618408.3618836.

S. Ghalebikesabi, L. Berrada, S. Gowal, I. Ktena, R. Stanforth, J. Hayes, S. De, S. L. Smith, O. Wiles, and B. Balle. Differentially private diffusion models generate useful synthetic images. *arXiv preprint arXiv:2302.13861*, 2023. https://doi.org/10.48550/arXiv.2302.13861.

A. Ginart, L. van der Maaten, J. Zou, and C. Guo. Submix: Practical private prediction for large-scale language models. *arXiv preprint arXiv:2201.00971*, 2022. https://openreview.net/forum?id=cKTBRHIVjy9.

A. Golatkar, A. Achille, Y.-X. Wang, A. Roth, M. Kearns, and S. Soatto. Mixed differential privacy in computer vision. In *Proceedings of the 2022 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, CVPR '22. IEEE Computer Society, 2022. https://openaccess.thecvf.com/content/CVPR2022/html/Golatkar_Mixed_Differential_Privacy_in_Computer_Vision_CVPR_2022_paper.html.

S. Gopi, Y. T. Lee, and L. Wutschitz. Numerical composition of differential privacy. In *Advances in Neural Information Processing Systems 34*, NeurIPS '21, pages 11631–11642. Curran Associates, Inc., 2021. https://proceedings.neurips.cc/paper/2021/hash/6097d8f3714205740f30debe1166744e-Abstract.html.

F. Harder, M. Jalali, D. J. Sutherland, and M. Park. Pre-trained perceptual features improve differentially private image generation. *Transactions on Machine Learning Research*, 2023. https://openreview.net/forum?id=R6W7zkMz0P.

J. He, X. Li, D. Yu, H. Zhang, J. Kulkarni, Y. T. Lee, A. Backurs, N. Yu, and J. Bian. Exploring the limits of differentially private deep learning with group-wise clipping. In *Proceedings of the 11th International Conference on Learning Representations*, ICLR '23, 2023. https://openreview.net/forum?id=oze0clVGPeX.

D. Hendrycks and K. Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. https://doi.org/10.48550/arXiv.1606.08415.

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8): 1735–1780, 1997. https://www.bioinf.jku.at/publications/older/2604.pdf.

S. Hoory, A. Feder, A. Tendler, A. Cohen, S. Erell, I. Laish, H. Nakhost, U. Stemmer, A. Benjamini, A. Hassidim, and Y. Matias. Learning and evaluating a differentially private pre-trained language model. In *Proceedings of the Third Workshop on Privacy in Natural Language Processing*, PrivateNLP '21, pages 21–29, 2021. https://doi.org/10.18653/v1/2021.findings-emnlp.102.

N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR, 2019. https://proceedings.mlr.press/v97/houlsby19a.html.

E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, and W. Chen. LoRA: Low-rank adaptation of large language models. In *Proceedings of the 10th International Conference on Learning Representations*, ICLR '22, 2022. https://openreview.net/forum?id=nZeVKeeFYf9.

Z. Ji and C. Elkan. Differential privacy based on importance weighting. *Machine Learning*, 93(1):163–183, 2013. https://doi.org/10.1007/s10994-013-5396-x.

P. Kairouz, M. Ribero, K. Rush, and A. Thakurta. (nearly) dimension independent private ERM with adagrad rates via publicly estimated subspaces. In *Proceedings of the 34th Annual Conference on Learning Theory*, COLT '21, pages 2717–2746, 2021. https://proceedings.mlr.press/v134/kairouz21a.html.

G. Kerrigan, D. Slack, and J. Tuyls. Differentially private language models benefit from public pre-training. *Proceedings of the Second Workshop on Privacy in NLP*, 2020. https://doi.org/10.18653/v1/2020.privatenlp-1.5.

A. Koskela, J. Jälkö, and A. Honkela. Computing tight differential privacy guarantees using fft. In *International Conference on Artificial Intelligence and Statistics*, pages 2560–2569. PMLR, 2020. https://proceedings.mlr.press/v108/koskela20b.html.

A. Koskela, J. Jälkö, L. Prediger, and A. Honkela. Tight differential privacy for discrete-valued mechanisms and for the subsampled gaussian mechanism using fft. In *International Conference on Artificial Intelligence and Statistics*, pages 3358–3366. PMLR, 2021. https://proceedings.mlr.press/v130/koskela21a.html.

A. Kurakin, N. Ponomareva, U. Syed, L. MacDermed, and A. Terzis. Harnessing large-language models to generate private synthetic text. *arXiv preprint arXiv:2306.01684*, 2023. https://doi.org/10.48550/arXiv.2306.01684.

B. Lester, R. Al-Rfou, and N. Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, EMNLP '21. Association for Computational Linguistics, 2021. https://doi.org/10.18653/v1/2021.emnlp-main.243.

C. Li, H. Farkhoor, R. Liu, and J. Yosinski. Measuring the intrinsic dimension of objective landscapes. In *Proceedings of the 6th International Conference on Learning Representations*, ICLR '18, 2018. https://openreview.net/forum?id=ryup8-WCW.

X. Li, D. Liu, T. B. Hashimoto, H. A. Inan, J. Kulkarni, Y.-T. Lee, and A. Guha Thakurta. When does differentially private learning not suffer in high dimensions? In *Advances in Neural Information Processing Systems 35*, NeurIPS '22, pages 28616–28630. Curran Associates, Inc., 2022a. https://proceedings.neurips.cc/paper_files/paper/2022/hash/b75ce884441c983f7357a312ffa02a3c-Abstract-Conference.html.

X. Li, F. Tramèr, P. Liang, and T. Hashimoto. Large language models can be strong differentially private learners. In *Proceedings of the 10th International Conference on Learning Representations*, ICLR '22, 2022b. https://openreview.net/forum?id=bVuP3ltATMz.

X. L. Li and P. Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, ACL-IJCNLP '21, pages 4582–4597, 2021. https://doi.org/10.18653/v1/2021.acl-long.353.

T. Liu, G. Vietri, T. Steinke, J. Ullman, and S. Wu. Leveraging public data for practical private query release. In *Proceedings of the 38th International Conference on Machine Learning*, ICML '21, pages 6968–6977. JMLR, Inc., 2021. https://proceedings.mlr.press/v139/liu21w.html.

Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. RoBERTa: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. https://openreview.net/forum?id=SyxS0T4tvS.

I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *Proceedings of the 7th International Conference on Learning Representations*, ICLR '19, 2019. https://openreview.net/forum?id=Bkg6RiCqY7.

Z. Luo, D. J. Wu, E. Adeli, and L. Fei-Fei. Scalable differential privacy with sparse network finetuning. In *Proceedings of the 2021 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, CVPR '21, pages 5059–5068, Washington, DC, USA, 2021. IEEE Computer Society. https://doi.org/10.1109/CVPR46437.2021.00502.

R. K. Mahabadi, J. Henderson, and S. Ruder. Compacter: Efficient low-rank hypercomplex adapter layers. In *Advances in Neural Information Processing Systems 34*, NeurIPS '21, pages 1022–1035. Curran Associates, Inc., 2021. https://proceedings.neurips.cc/paper/2021/hash/081be9fdff07f3bc808f935906ef70c0-Abstract.html.

H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang. Learning differentially private recurrent language models. In *Proceedings of the 6th International Conference on Learning Representations*, ICLR '18, 2018. https://openreview.net/forum?id=BJ0hF1Z0b.

H. Mehta, A. G. Thakurta, A. Kurakin, and A. Cutkosky. Towards large scale transfer learning for differentially private image classification. *Transactions on Machine Learning Research*, 2023. https://doi.org/10.48550/arXiv.2205.02973.

P. Micikevicius, S. Narang, J. Alben, G. Diamos, E. Elsen, D. Garcia, B. Ginsburg, M. Houston, O. Kuchaiev, G. Venkatesh, and H. Wu. Mixed precision training. In *Proceedings of the 6th International Conference on Learning Representations*, ICLR '18, 2018. https://openreview.net/forum?id=r1gs9JgRZ.

F. Mireshghallah, A. Backurs, H. A. Inan, L. Wutschitz, and J. Kulkarni. Differentially private model compression. In *Advances in Neural Information Processing Systems 35*, NeurIPS '22, pages 29468–29483. Curran Associates, Inc., 2022. https://proceedings.neurips.cc/paper_files/paper/2022/hash/bd6bb13e78da078d8adcabbe6d9ca737-Abstract-Conference.html.

L. Nan, D. Radev, R. Zhang, A. Rau, A. Sivaprasad, C. Hsieh, X. Tang, A. Vyas, N. Verma, P. Krishna, Y. Liu, N. Irwanto, J. Pan, F. Rahman, A. Zaidi, M. Mutuma, Y. Tarabar, A. Gupta, T. Yu, Y. C. Tan, X. V. Lin, C. Xiong, R. Socher, and N. F. Rajani. DART: open-domain structured data record to text generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL-HLT '21, pages 432–447. Association for Computational Linguistics, 2021. https://doi.org/10.18653/v1/2021.naacl-main.37.

A. Nandi and R. Bassily. Privately answering classification queries in the agnostic PAC model. In *Proceedings of the 31st International Conference on Algorithmic Learning Theory*, ALT '20, pages 687–703. JMLR, Inc., 2020. https://proceedings.mlr.press/v117/nandi20a.html.

J. Novikova, O. Dušek, and V. Rieser. The E2E dataset: New challenges for end-to-end generation. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, SIGDIAL '17, pages 201–206. Association for Computational Linguistics, 2017. https://doi.org/10.18653/v1/W17-5525.

M. Ott, S. Edunov, D. Grangier, and M. Auli. Scaling neural machine translation. In *Proceedings of the Third Conference on Machine Translation (WMT)*, 2018. https://doi.org/10.18653/v1/W18-6301.

N. Papernot, M. Abadi, U. Erlingsson, I. Goodfellow, and K. Talwar. Semi-supervised knowledge transfer for deep learning from private training data. In *Proceedings of the 5th International Conference on Learning Representations*, ICLR '17, 2017. https://openreview.net/forum?id=HkwoSDPgg.

N. Papernot, S. Song, I. Mironov, A. Raghunathan, K. Talwar, and Ú. Erlingsson. Scalable private learning with PATE. In *Proceedings of the 6th International Conference on Learning*

Representations, ICLR '18, 2018. https://openreview.net/forum?id=rkZB1XbRZ.

N. Papernot, S. Chien, S. Song, A. Thakurta, and U. Erlingsson. Making the shoe fit: Architectures, initializations, and tuning for learning with privacy. https://openreview.net/forum?id=rJg851rYwH, 2019.

M. Pelikan, S. S. Azam, V. Feldman, J. Silovsky, K. Talwar, T. Likhomanenko, et al. Federated learning with differential privacy for end-to-end speech recognition. *arXiv preprint arXiv:2310.00098*, 2023. https://doi.org/10.48550/arXiv.2310.00098.

J. Pfeiffer, A. Kamath, A. Rücklé, K. Cho, and I. Gurevych. Adapterfusion: Non-destructive task composition for transfer learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, EACL '21, pages 487–503. Association for Computational Linguistics, 2021. https://doi.org/10.18653/v1/2021.eacl-main.39.

A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. Improving language understanding by generative pre-training, 2018. https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf.

A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners, 2019. https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

S. Ramaswamy, O. Thakkar, R. Mathews, G. Andrew, H. B. McMahan, and F. Beaufays. Training production language models without memorizing user data. *arXiv preprint arXiv:2009.10031*, 2020. https://doi.org/10.48550/arXiv.2009.10031.

A. Rücklé, G. Geigle, M. Glockner, T. Beck, J. Pfeiffer, N. Reimers, and I. Gurevych. Adapterdrop: On the efficiency of adapters in transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, EMNLP '21, pages 7930–7946. Association for Computational Linguistics, 2021. https://doi.org/10.18653/v1/2021.emnlp-main.626.

M. Senge, T. Igamberdiev, and I. Habernal. One size does not fit all: Investigating strategies for differentially-private learning across nlp tasks. *arXiv preprint arXiv:2112.08159*, 2021. https://doi.org/10.18653/v1/2022.emnlp-main.496.

R. Shokri, M. Stronati, C. Song, and V. Shmatikov. Membership inference attacks against machine learning models. In *Proceedings of the 38th IEEE Symposium on Security and Privacy*, SP '17, pages 3–18, Washington, DC, USA, 2017. IEEE Computer Society. https://doi.org/10.1109/SP.2017.41.

S. Song, K. Chaudhuri, and A. D. Sarwate. Stochastic gradient descent with differentially private updates. In *Proceedings of the 2013 IEEE Global Conference on Signal and Information Processing*, GlobalSIP '13, pages 245–248, Washington, DC, USA, 2013. IEEE Computer Society. https://doi.org/10.1109/GlobalSIP.2013.6736861.

P. Subramani, N. Vadivelu, and G. Kamath. Enabling fast differentially private sgd via just-in-time compilation and vectorization. In *Advances in Neural Information Processing Systems 34*, NeurIPS '21. Curran Associates, Inc., 2021. https://proceedings.neurips.cc/paper/2021/hash/ddf9029977a61241841edeae15e9b53f-Abstract.html.

Z. Tian, Y. Zhao, Z. Huang, Y.-X. Wang, N. Zhang, and H. He. SeqPATE: Differentially private text generation via knowledge distillation, 2022. https://openreview.net/forum?id=5sP_PUUS78v.

F. Tramèr and D. Boneh. Differentially private learning needs better features (or much more data). In *Proceedings of the 9th International Conference on Learning Representations*,

ICLR '21, 2021. https://openreview.net/forum?id=YTWGvpFOQD-.

F. Tramèr, G. Kamath, and N. Carlini. Considerations for differentially private learning with large-scale public pretraining. *arXiv preprint arXiv:2212.06470*, 2022. https://doi.org/10.48550/arXiv.2212.06470.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, NIPS '17, pages 5998–6008. Curran Associates, Inc., 2017. https://papers.nips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*, 2019. https://openreview.net/forum?id=rJ4km2R5t7.

A. Williams, N. Nangia, and S. Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, NAACL-HLT '18, pages 1112–1122. Association for Computational Linguistics, 2018. https://doi.org/10.18653/v1/N18-1101.

F. Wu, H. A. Inan, A. Backurs, V. Chandrasekaran, J. Kulkarni, and R. Sim. Privately aligning language models with reinforcement learning. *International Conference on Learning Representations*, 2024. https://openreview.net/forum?id=3d0OmYTNui.

R. Wu, C. Guo, and K. Chaudhuri. Large-scale public data improves differentially private image generation quality. *arXiv preprint arXiv:2309.00008*, 2023. https://doi.org/10.48550/arXiv.2309.00008.

D. Yu, H. Zhang, W. Chen, and T.-Y. Liu. Do not let privacy overbill utility: Gradient embedding perturbation for private learning. In *Proceedings of the 9th International Conference on Learning Representations*, ICLR '21, 2021a. https://openreview.net/forum?id=7aogOj_VYO0.

D. Yu, H. Zhang, W. Chen, J. Yin, and T.-Y. Liu. Large scale private learning via low-rank reparametrization. In *Proceedings of the 38th International Conference on Machine Learning*, ICML '21. JMLR, Inc., 2021b. https://proceedings.mlr.press/v139/yu21f.html.

X. Yue, H. A. Inan, X. Li, G. Kumar, J. McAnallen, H. Shajari, H. Sun, D. Levitan, and R. Sim. Synthetic text generation with differential privacy: A simple and practical recipe. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL '23, pages 1321–1342. Association for Computational Linguistics, 2022. https://doi.org/10.18653/v1/2023.acl-long.74.

Y. Zhou, Z. S. Wu, and A. Banerjee. Bypassing the ambient dimension: Private SGD with gradient subspace identification. In *Proceedings of the 9th International Conference on Learning Representations*, ICLR '21, 2021. https://openreview.net/forum?id=7dpmlkBuJFC.

Appendix A. Experiments with Different Privacy Parameters

Now we test our framework under different privacy constraints. Specifically, we run LoRA on the language understanding tasks with various choices of privacy parameters ϵ and δ. We consider both RoBERTa-Base and RoBERTa-Large.

TABLE 8. Test accuracy for fine-tuning RoBERTa-Large with different privacy parameters. The number of training samples is denoted by n. The values of σ are noise multipliers. Numbers in the parentheses are the changes compared to the results in Table 5 ($\epsilon = 6.7$, $\delta = \Theta(1/n)$).

Taks	σ	$\delta = 1/n$	$\delta = 1/10n$	$\delta = 1/100n$	$\delta = 1/1000n$	Accuracy (in %)
MNLI	1.88	$\epsilon = 1$	$\epsilon = 1.35$	$\epsilon = 1.49$	$\epsilon = 1.61$	86.8 (-1.0%)
QQP	1.88	$\epsilon = 1$	$\epsilon = 1.40$	$\epsilon = 1.54$	$\epsilon = 1.67$	85.2 (-2.2%)
QNLI	3.01	$\epsilon = 1$	$\epsilon = 1.48$	$\epsilon = 1.64$	$\epsilon = 1.79$	88.0 (-2.8%)
SST-2	3.63	$\epsilon = 1$	$\epsilon = 1.47$	$\epsilon = 1.64$	$\epsilon = 1.80$	93.1 (-2.2%)
MNLI	0.91	$\epsilon = 3$	$\epsilon = 4.12$	$\epsilon = 4.51$	$\epsilon = 4.89$	87.4 (-0.4%)
QQP	0.93	$\epsilon = 3$	$\epsilon = 4.10$	$\epsilon = 4.49$	$\epsilon = 4.86$	86.8 (-0.6%)
QNLI	1.29	$\epsilon = 3$	$\epsilon = 4.45$	$\epsilon = 4.90$	$\epsilon = 5.33$	89.9 (-0.9%)
SST-2	1.52	$\epsilon = 3$	$\epsilon = 4.37$	$\epsilon = 4.83$	$\epsilon = 5.25$	94.1 (-1.2%)

For the RoBERTa-Large model, we set $\epsilon = 1$ and 3 with δ being the same as those in Section 4. We use the PRV accountant (Gopi et al., 2021). After getting the noise multipliers, we also reduce the value of δ and report the corresponding value of ϵ. The hyperparameters are the same as those in Section 4. We run experiments on all four tasks, i.e., MNLI ($n \sim$ 392k), QQP ($n \sim$ 364k), QNLI ($n \sim$ 104k), and SST-2 ($n \sim$ 67k). We use the official splits of validation and test sets. MNLI and QNLI are tasks that assess a model's capability to perform natural language inference. Given two sentences, the task in MNLI is to determine if the second sentence is an entailment, contradiction, or neutral with respect to the first sentence. Given one premise and one question, the task in QNLI is to determine whether the question can be answered by the premise. The task in QQP is to predict whether two questions are duplicated. The task in SST-2 is to perform sentiment analysis of sentences from movie reviews.

We report the results in Table 8. The performance of our framework is decent even with very tight privacy budgets. For instance, with $\epsilon < 2$ and $\delta = 1/1000n$, the accuracy gap between the non-private baseline is only 3.8 for MNLI and 2.1 for SST-2.

For the RoBERTa-Base model, we try various choices of ϵ. The values of ϵ are chosen from $[0.1, 0.5, 1, 3, 5, 8, 12]$. All other settings are the same as those in Section 4. We run experiments on the MNLI and SST-2 datasets. The results are presented in Figure 2. Our framework performs well for a wide range of ϵ. We note that our algorithm achieves meaningful accuracy even for very tight privacy parameters $\epsilon = 0.5$ and 1. Such values of ϵ are rarely explored when training deep models with differential privacy.

APPENDIX B. ON THE INFLUENCE OF HYPERPARAMETERS

Here we demonstrate that our algorithms perform well for a wide range of hyperparameters. We study two hyperparameters that are directly related to the variance of noise: clipping threshold and batchsize. The clipping threshold is chosen from $[0.1, 1.0, 3.0, 5.0, 10.0]$ and the batchsize is chosen from $[200, 500, 1000, 2000, 4000]$. We note that we keep the number of updates the same as that in Section 4 when the batchsize is changed. We fine-tune the

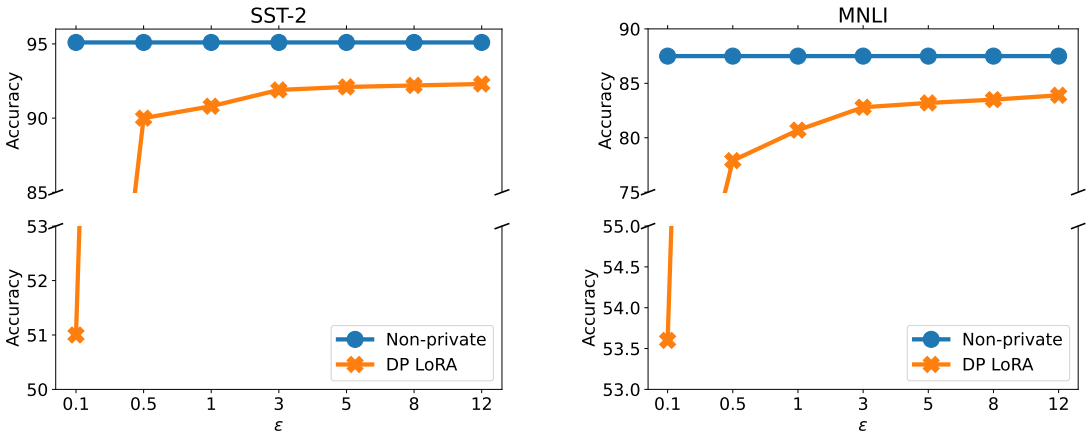

FIGURE 2. Test accuracy (in %) of fine-tuning the RoBERTa-Base model on MNLI and SST-2 with various choices of ϵ.

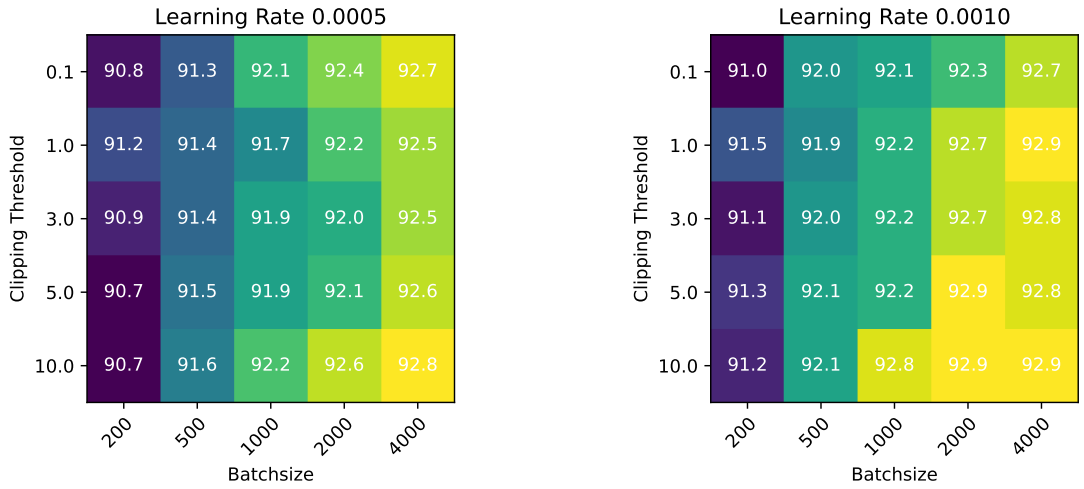

FIGURE 3. Test accuracy (in %) of fine-tuning RoBERTa-Base with differentially private LoRA on the SST-2 dataset. Our algorithm performs well on a wide range of hyperparameters.

RoBERTa-Base model with differentially private LoRA ($r = 16$) on the SST-2 dataset. The results are presented in Figure 3. DP LoRA performs well for all the hyperparameters considered. The gap between the best accuracy and the worst accuracy is only 2%.

APPENDIX C. FINE-TUNING FOR LANGUAGE UNDERSTANDING TASKS WITH LARGE BATCH SIZE AND FULL-PRECISION

Li et al. (2022b) show the performance of fine-tuning the full model is sensitive to the choice of hyperparameters. They give a configuration which can significantly improve the performance of full fine-tuning. In this section, we re-evaluate the tasks in Table 4 and 5 under the configuration in Li et al. (2022b).

TABLE 9. Accuracy for fine-tuning downstream tasks with RoBERTa-Base (in %). Experiments are run with full-precision. We also scale up the batch size according to the dataset size compared to SST-2. The privacy parameters are $\epsilon = 6.7$, and $\delta = 1 \times 10^{-5}$ for SST-2 and QNLI and 1×10^{-6} for MNLI and QQP.

Method		MNLI	SST-2	QQP	QNLI	Average Accuracy
Full	w/o DP	87.6	94.8	91.9	92.8	91.8
	DP	83.2	85.9	86.2	84.8	85.0
Adapter	DP	**84.6**	**92.9**	**87.4**	**89.2**	**88.5**
LoRA	DP	84.5	92.7	87.1	88.3	88.2

TABLE 10. Accuracy for fine-tuning downstream tasks with RoBERTa-Large (in %). Experiments are run with full-precision. We also scale up the batch size according to the dataset size compared to SST-2. The privacy parameters are $\epsilon = 6.7$, and $\delta = 1 \times 10^{-5}$ for SST-2 and QNLI and $\delta = 1 \times 10^{-6}$ for MNLI and QQP.

Method		MNLI	SST-2	QQP	QNLI	Average Accuracy
Full	w/o DP	90.2	96.4	92.2	94.7	93.4
	DP	86.4	90.9	87.5	89.4	88.6
Adapter	DP	88.6	94.5	87.8	91.6	90.6
LoRA	DP	**89.0**	**95.3**	**88.4**	**92.4**	**91.3**

The configuration in Li et al. (2022b) has two differences compared to that in Section 4. The first difference is Li et al. (2022b) run experiments with full-precision while the experiments in Section 4 use half-precision. Using half-precision is a common approach to speed up NLP experiments (Ott et al., 2018). However, half-precision may incur underflow issue which impacts the model performance (Micikevicius et al., 2018). The second difference is they use larger batch size for larger datasets. For example, the batch size for MNLI is roughly six times larger than the batch size for SST-2 in Li et al. (2022b). In Section 4, we use the same batch size for all datasets.

We follow the above setup and re-evaluate DP-LoRA and DP-Adapter. The results are in Table 9 and 10. The results of full fine-tuning with differential privacy are directly adopted from Li et al. (2022b). The configuration in Li et al. (2022b) further improves the strong results in Table 4 and 5. For example, we achieve 89.0% accuracy on the MNLI dataset, which is only 1.2% lower than the accuracy without DP constraint. Moreover, the benefit of the proposed framework over full fine-tuning is still clear. The average accuracy of the proposed algorithms is ∼3% higher than that of full fine-tuning.

APPENDIX D. COMPARISONS WITH THE RESULTS IN SUBSEQUENT WORKS

Since the original appearance of this paper, several subsequent works also conduct experiments on the GLUE benchmark (Bu et al., 2022, 2023; He et al., 2023). In this section, we compare our results with those in two representative works (Bu et al., 2022; He et al., 2023). Bu et al. (2022) explore the limit of parameter-efficient fine-tuning by fine-tuning only the bias-term parameters. Fine-tuning only the bias-term parameters does not require

TABLE 11. Accuracy (in %) comparisons with the results in subsequent works of the original appearance of this paper. The pretrained model is RoBERTa-Base. The privacy parameters are $\epsilon = 6.7$, and $\delta = 1 \times 10^{-5}$ for SST-2 and QNLI and $\delta = 1 \times 10^{-6}$ for MNLI and QQP.

Method	MNLI	SST-2	QQP	QNLI	Average Accuracy
LoRA (Non-private)	87.5	95.1	90.8	93.3	91.7
LoRA	83.5	92.2	85.7	87.3	87.2
Bu et al. (2022)	82.6	92.4	83.4	86.5	86.2
He et al. (2023)	83.8	92.4	86.2	87.1	87.4

TABLE 12. Accuracy (in %) comparisons with the results in subsequent works of the original appearance of this paper. The pretrained model is RoBERTa-Large. The privacy parameters are $\epsilon = 6.7$, and $\delta = 1 \times 10^{-5}$ for SST-2 and QNLI and $\delta = 1 \times 10^{-6}$ for MNLI and QQP.

Method	MNLI	SST-2	QQP	QNLI	Average Accuracy
LoRA (Non-private)	90.6	96.2	91.6	94.9	93.3
LoRA	87.8	95.3	87.4	90.8	90.3
Bu et al. (2022)	87.6	94.5	86.5	91.0	89.9
He et al. (2023)	87.6	94.0	87.2	90.8	89.9

caching the activations during forward, and hence is more memory-efficient than other parameter-efficient methods. He et al. (2023) employ an adaptive per-layer clipping method that significantly improves the efficiency of DP training.

Table 11 and 12 present the results of fine-tuning RoBERTa-base and RoBERTa-large, respectively. The results of LoRA fine-tuning are from Table 4 and 5. The new results in Bu et al. (2022) and He et al. (2023) further confirm the two findings in this paper 1) DP fine-tuning can achieve comparable accuracy as the non-private counterpart with a small computational overhead, 2) the gap between private fine-tuning and non-private fine-tuning diminishes as the pre-trained model becomes more powerful.