Differentially Private Parameter-Efficient Fine-tuning for Large ASR Models

Hongbin Liu[†‡*], Lun Wang[‡], Om Thakkar[‡], Abhradeep Thakurta[‡], Arun Narayanan[‡]
[†]Duke University [‡]Google
hongbin.liu@duke.edu, {lunwang, omthkkr, athakurta, arunnt}@google.com

***Abstract*—Large ASR models are often pre-trained on a large corpus of public data and then fine-tuned on potentially sensitive/proprietary downstream data. However, recent work shows that such ASR models may leak privacy of fine-tuning data. Differentially private SGD (DP-SGD) is the standard method for fine-tuning ASR models with formal DP guarantees. However, fine-tuning full ASR models with DP-SGD is computationally expensive and can hurt model's utility due to the *curse of dimensionality* for DP-SGD. Thus DP parameter-efficient fine-tuning (PEFT) is a competitive alternative. In this work, we conduct the first comprehensive evaluation of DP full fine-tuning and DP-PEFT methods for training ASR models. We also propose novel modifications to improve privacy-utility trade-offs, e.g., training on synthetic data between pre-training and fine-tuning. Our best method for DP-PEFT achieves state-of-the-art 8.0% WER on LibriSpeech test-other under a strong $(10, 3.52\mathrm{e}{-6})$-DP guarantee on a 600M Conformer.**

I. Introduction

The performance of Automatic Speech Recognition (ASR) models has significantly improved over the last several years. Large ASR models such as Conformer [1] and wav2vec [2] set new benchmarks on speech recognition tasks while also enabling strong performance on related tasks like speech translation [3], [4], and speaker verification [5], [6]. The current training paradigm of large ASR models usually pre-trains a foundation model on a large corpus of often publicly-available speech data to learn general representations of speech in a self-supervised or semi-supervised manner. The pre-trained models are subsequently adapted for downstream tasks by fine-tuning on potentially sensitive and/or proprietary domain-specific datasets.

As large ASR models become more capable, ensuring privacy of their training data is increasingly important. Recent work [7], [8], [9] has shown various privacy attacks on trained ASR models. Specifically, Wang *et al.* [8] demonstrate that large ASR models can *unintentionally memorize* rare/unique samples in their training data, highlighting the necessity to preserve the privacy of the data used in training such models. Differentially private stochastic gradient descent (DP-SGD) is the gold standard for training machine learning models with formal privacy guarantees. However, using DP-SGD to train large models usually hinders model performance [10], [11], [12], and incurs a high computational overhead compared to non-private training [13], [14]. This can be prohibitive since the (non-private) training cost for large models is usually exorbitant to begin with.

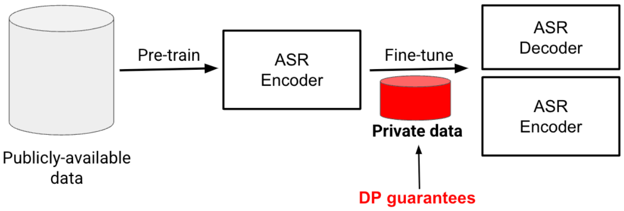


Fig. 1: Illustrating our setting of DP fine-tuning.

In this work, we consider the following set-up (also shown in Figure 1) where we first pre-train an ASR encoder on a large amount of publicly-available data, and then privately fine-tune the encoder and an attached decoder on a dataset with privacy requirements. Motivated by recent work in the DP language modeling domain [15], we extensively study parameter-efficient ASR fine-tuning with DP in order to achieve better privacy-utility-computation trade-offs. We conduct the first comprehensive evaluation of a variety of differentially private parameter-efficient fine-tuning (DP-PEFT) methods including Adapter [16], LoRA [17] and BitFit [18], on a large state-of-the-art 600M parameter Conformer [1] ASR model.

We also propose a new PEFT method, namely random projection, which frzes the downscale projection matrix in LoRA to halve the number of trainable parameters. We compare these methods to standard differentially private fine-tuning (DP-FT) in terms of word error rate (WER), number of trainable parameters, and compute efficiency. Our experiments show that DP-BitFit is the best performing and most parameter efficient DP-PEFT method. DP-BitFit achieves the state-of-the-art WER of 10.94% on LibriSpeech test-other and 7.79% WER on LibriSpeech test-clean at $(10, 3.52\mathrm{e}{-6})$-DP by fine-tuning only 2.9% of the parameters. Further, our work is the first to show that one can leverage random synthetic audio data (arguably having even lower privacy risks than publicly-available data) to improve the privacy-utility trade-offs for DP fine-tuning of large ASR models. Using synthetic data and other standard optimizations, we improve the WER using DP-BitFit to 8.0%. More generally, our results show that DP-PEFT is advantageous for practical DP fine-tuning of large ASR models.

To summarize, we make the following main contributions:

- We comprehensively evaluate the mainstream DP-PEFT methods on large ASR models and find that DP-BitFit

[*] Work done as a student researcher at Google.

provides the best privacy-utility trade-offs.

- We propose to use synthetic data to further improve the privacy-utility trade-offs for DP-BitFit.
- Using DP-BitFit along with our improvements using synthetic data, we achieve 8.0% WER on Librispeech test-other using a 600M Conformer model pre-trained on Librilight and fine-tuned on Librispeech with $(10, 3.52\text{e}{-6})$-DP, which can serve as a benchmark for future DP ASR fine-tuning.

II. Background and Related Work

A. Differential Privacy

Differential privacy [19] (DP) is a formal notion of privacy, at a high level based on comparing the information leakage of any algorithm on *adjacent* input datasets. Specifically, two input datasets are considered adjacent if one can be obtained from the other by adding or removing one example. Informally, a randomized algorithm satisfies DP if its output distributions on any possible pair of adjacent input datasets are statistically *close*.

Definition 1 (Differential privacy [19]). *A randomized function $\mathcal{F} : \mathcal{D} \to \mathcal{R}$ satisfies (ε, δ)-DP if, for any two adjacent datasets D, $D' \in \mathcal{D}$ and for any subset $S \subseteq \mathcal{R}$, it holds that:*

$$Pr[\mathcal{F}(D) \in S] \leq e^{\varepsilon} Pr[\mathcal{F}(D') \in S] + \delta. \quad (1)$$

In deep learning, $\mathcal{F}$ is the training algorithm. Smaller values of ε, δ correspond to a stronger privacy guarantee.

DP-SGD [20] is the workhorse for DP deep learning. In each training step, it randomly samples a mini-batch of examples, *clips* each example's gradient to a prefixed L2-bound, and then adds calibrated noise to ensure privacy for the gradient update. The DP guarantee for the algorithm is derived by *composition* across training steps, using the Moments Accountant[20], [21]. Though the foundational technique is DP-SGD, the noised mini-batch gradient can be passed to any deep learning optimizer that takes a mini-batch gradient as input, without affecting the privacy of the training method. We use Adam optimizer [22] in all experiments in this paper.

B. Parameter-Efficient Fine-Tuning

Parameter-efficient fine-tuning (PEFT) [16], [17], [18], [23], [24] is a class of fine-tuning techniques that update only a small number of either newly added, or existing parameters of a pre-trained large model. The first proposed PEFT is Adapter [16], which adds two low-rank projection matrices and one activation layer before the layer norm and after the feed-forward layer in each Transformer block [25], and only fine-tunes the added parameters. Inspired by the success of Adapters, various PEFT techniques have been proposed ever since. LoRA [17] adds two low-rank projection matrices parallel to feed-forward layers, and fine-tunes them. BitFit [18] is a sparse fine-tuning method that trains only the bias terms of the model. Figure 2 shows an illustration of full fine-tuning, and popular PEFT methods for Encoder-Decoder ASR models.

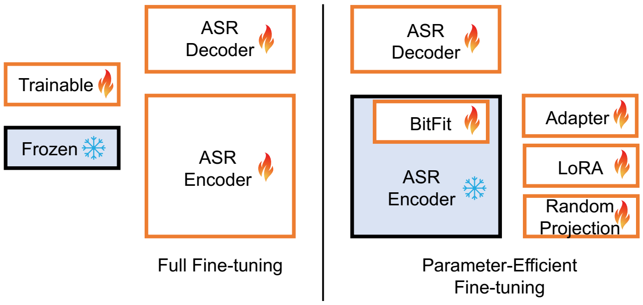


Fig. 2: Illustration of full fine-tuning and popular PEFT methods for Encoder-Decoder ASR models.

DP-PEFT methods are also widely studied since they attempt to solve the optimization problem in a lower-dimensional space, which can be beneficial for DP training [10], [26]. Yu *et al.* [27] proposed reparameterized gradient perturbation, which only perturbs the decomposed low-rank matrices. Yu *et al.* [15] conducted a study of DP-PEFT methods on large language models, highlighting the potential of DP-PEFT to beat DP-FT in the language domain. Bu *et al.* [28] proposed DP-BitFit which achieves the best memory efficiency among all DP-PEFT methods via customized computation graph optimization.

III. DP-FT and DP-PEFTs for ASR

In this section, we conduct comprehensive experiments to compare DP full fine-tuning (DP-FT) versus DP-PEFT methods on a large ASR model. The PEFT methods we evaluate include Adapter, LoRA, and BitFit. We also introduce another PEFT method called random projection (RP), which can be implemented with a slight modification to LoRA, i.e., by freezing the downscale projection matrix, and training only the upscale projection matrix. By doing so, we halve the number of trainable parameters in LoRA while achieving comparable performance in the non-private setting. Our motivation for RP for DP-PEFT is reducing the dimensionality for DP training. Our experiments evaluate each method to understand the trade-offs between privacy, model quality, and memory efficiency.

A. Experimental Setup

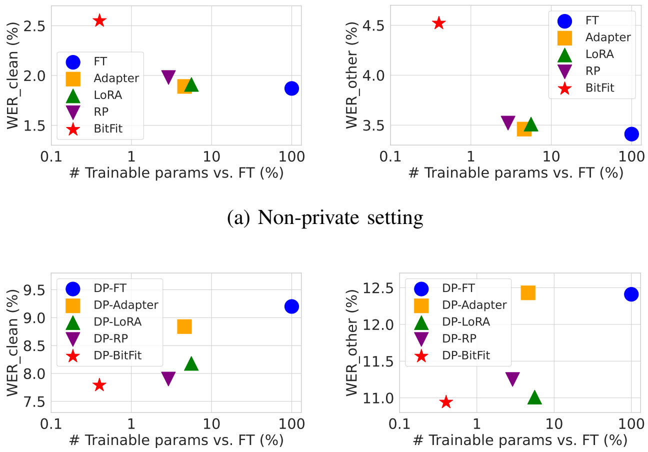
Training Recipe: We use a 600M parameter Conformer encoder model [1] for our experiments. We replace batch normalization [29] (BN) layers in the model with group normalization [30] (GN) since BN enables information mixing across training examples in the mini-batch, and thus results in worse DP guarantees. We pre-train the encoder using the BEST-RQ algorithm [31] on LibriLight [32] for one million steps. We then attach a CTC decoder [33] and the PEFT-specific parameters, if any, to the pre-trained encoder, and fine-tune the model using LibriSpeech [34] for 100k steps unless noted. Our experiments are implemented in PAX [35], and run on Dragonfish TPUs with 8x8 topology.

Hyperparameters: Across all our (ε, δ)-DP experiments, we set the clipping bound C to 2.5. We fix $\delta = 3.52\text{e}{-6}$ to ensure it is smaller than n^{-1}, where n is the number of training samples, as is standard in literature. For Librispeech, we have $n = 281241$. The noise multiplier is set to achieve (ε, δ)-DP with $\varepsilon = 10.0$ for our fixed δ. Note that according to recent work [36], such a level of DP can be classified in the "Tier 2: Reasonable privacy guarantees". To optimize performance, we grid search key method-specific hyperparameters for each PEFT (if any), e.g., the rank in LoRA. Since the trainable parameters differ among FT and PEFT methods (as shown in Figure 2), we separately grid search learning rates for the encoder/added module(s) and the decoder to find the optimal values for each DP FT/PEFT method. Unless otherwise stated, we report the best experimental results for each DP FT/PEFT method after the hyperparameter search mentioned above.

Evaluation Metrics: We report the word error rate (WER) on both LibriSpeech test splits (*i.e.* test-clean and test-other), denoted as WER_clean and WER_other, for our main results. Due to space constraints, ablation studies will only report WER_other, as it reflects a more challenging evaluation scenario. However, we emphasize that performance trends generally align across both test splits.

B. Experimental Results

FT vs. PEFT: We first compare FT vs. PEFT without DP, as shown in Figure 3a. In line with the PEFT literature, FT achieves the best WER_clean of 1.87% and WER_other of 3.41%. Among all the PEFTs, Adapter achieves the best WER_clean of 1.89% and WER_other of 3.46%. BitFit, while being the most parameter-efficient, suffers the strongest performance regression. Another interesting observation is that LoRA achieves 1.91% WER_clean and 3.51% WER_other while RP, with half the number of trainable parameters in encoder, achieves 1.98% WER_clean and 3.52% WER_other, which indicates the potential of RP to replace LoRA in ASR.



(a) Non-private setting

(b) $(10, 3.52\text{e}{-}6)$-DP

Fig. 3: Comparison results of WER_clean and WER_other for DP FT/PEFT methods for the same number of fine-tuning steps.

Next, we compare WER_other between DP-FT and DP-PEFTs at $(10, 3.52\text{e}{-}6)$-DP, as summarized in Figure 3b. To our surprise, DP-BiTFiT provides the best performance in terms of both parameter efficiency and WER. It fine-tunes only 2.9% parameters compared to FT, and achieves the best WER_other of 10.94% and WER_clean of 7.79%. All DP-PEFTs outperform DP-FT in terms of WER_clean, and except for DP-Adapter, they also outperform DP-FT in WER_other. This conforms with the observation by Yu et al. [15] in the language domain. The conjectured reason is that PEFT methods explore a much lower dimension space than FT, and thus the impact of the added noise in DP is smaller.

1) Tuning details for specific DP-PEFTs: **Bias Terms in DP-BitFit:** In DP-BitFit, we observe that the default training of all bias terms in the ASR encoder leads to divergence of training loss. We further conduct an architecture search for it. Specifically, we freeze three types of bias terms in the ASR encoder separately: those in layer normalization, convolutional layers, and feed-forward layers. We find that freezing bias terms in layer normalization prevents divergence. Thus, we use this setting in all experiments involving DP-BitFit.

Initialization in DP-LoRA and DP-RP: Following Hu et al. [17], we use a random Gaussian initialization with zero mean and σ standard deviation for every downscale projection matrix in DP-LoRA and DP-RP. Figure 4 shows the impact of σ on DP-LoRA and DP-RP. We observe that for both DP-LoRA and DP-RP, the WER_other first decreases then increases as σ increases. For example, DP-LoRA achieved the lowest WER_other of 11.0% when $\sigma = 0.4$, and DP-RP achieved the lowest WER_other of 11.3% when $\sigma = 0.3$. Our results indicate that appropriately setting σ for initialization can be crucial for the performance of DP-LoRA and DP-RP.

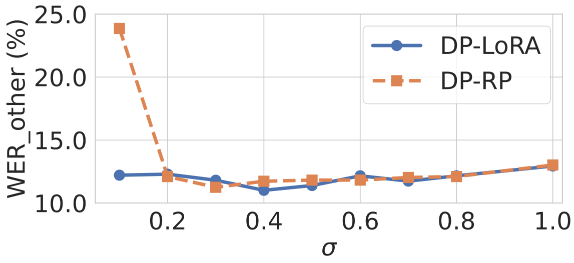
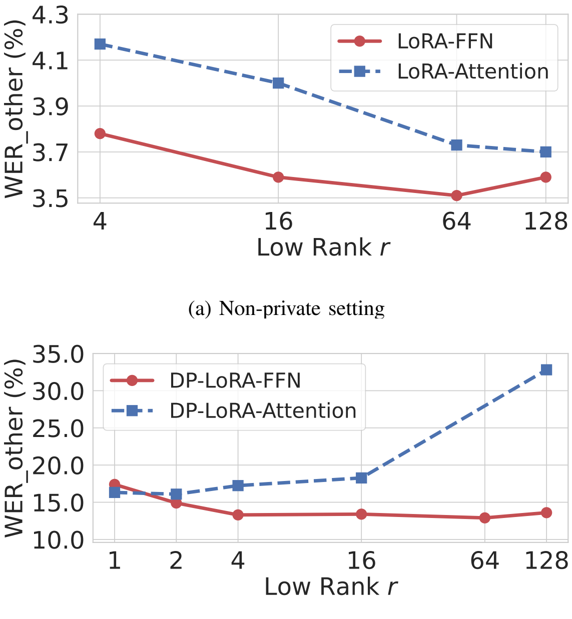


Fig. 4: Impact of σ in Gaussian initialization on DP-LoRA/DP-RP with $(10, 3.52\text{e}{-}6)$-DP.

LoRA Locations: We study the impact of adding LoRA into different modules within the Conformer. Specifically, we add LoRA either into the feed-forward (FFN) modules (LoRA-FFN), or the self-attention modules (LoRA-Attention). Figure 5a shows the comparison results in the non-private setting. We see that LoRA-FFN always achieves lower WER than LoRA-Attention at the same rank, indicating better utility when adding LoRA to FFN modules. This differs from Transformers, where LoRA is typically added into self-attention modules [17].

Under $(10, 3.52\text{e}{-}6)$-DP, Figure 5b shows DP-LoRA-FFN almost always outperforms DP-LoRA-Attention. To test the statistical significance of the performance of DP-LoRA-FFN



(a) Non-private setting

(b) $(10, 3.52e{-}6)$-DP

Fig. 5: Comparison results of WER_other for LoRA when adding to self-attention and FFN.

over DP-LoRA-Attention, we conduct a paired t-test with 3 repeats under their best-performing settings. DP-LoRA-FFN achieves $12.94\% \pm 0.05$ WER_other, whereas DP-LoRA-Attention achieves $15.60\% \pm 0.02$, where $\pm$ is followed by standard deviations. When $\alpha = 0.05$, the resulting p-value is $7.11e{-}5$, which indicates a significant difference in WER_other. Thus, DP-LoRA-FFN achieves statistically better utility than DP-LoRA-Attention at $(10, 3.52e{-}6)$-DP.

C. Ablation Studies

Mini-Batch Size for DP-PEFT: Prior work [37], [38], [12] shows that using larger mini-batch sizes is beneficial for private training in the language and vision domains. PEFT methods improve both memory and time efficiency of fine-tuning, so it can be feasible to increase mini-batch sizes of DP-PEFT methods when placing the same compute constraints as DP-FT. We conduct an ablation study by increasing the batch size for all DP-PEFT methods while consuming the same TPU-hours as DP-FT. Concretely, we experiment by increasing the default 512 mini-batch size by a factor of $\{2, 4, 8, 12\}$, and adjust the training steps and noise multiplier accordingly to make sure all the runs consume almost equal TPU-hours and satisfy $(10, 3.52e{-}6)$-DP.

Table I compares DP-FT and DP-PEFT methods by showing WER_clean, WER_other and the optimal mini-batch size multiplier for each method. Equating for TPU hours, we see that all DP-PEFT methods achieve lower WER_other and WER_clean than DP-FT. Their performance ranking is mostly consistent with that when equating the number of fine-tuning steps (Fig. 3b), except for DP-Adapter which improves from the worst (with equal fine-tuning steps) to comparable to DP-BitFit. DP-BitFit still provides the best WER_other of 9.1%.

TABLE I: Comparison results for DP FT/PEFT methods for the same TPU-hours under $(10, 3.52e{-}6)$-DP. **Bold** highlights optimal results.

Method	WER		Optimal Mini-Batch Size Multiplier
	clean	other	
DP FT	9.2	12.4	1
DP Adapter	**6.0**	9.2	8
DP LoRA	7.0	9.9	12
DP RP	8.5	11.3	2
DP BitFit	6.1	**9.1**	4

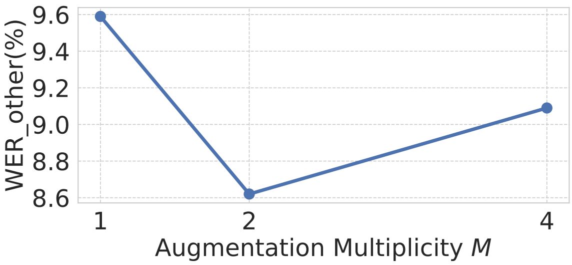


Fig. 6: Impact of augmentation multiplicity on DP-BitFit within the same TPU-hours under $(10, 3.52e{-}6)$-DP.

Augmentation Multiplicity for DP-BitFit: De et al. [12] demonstrate that DP training for vision models benefits from averaging gradients across M augmented versions of the same training image before gradient clipping, where M denotes the augmentation multiplicity. Here, we study the impact of augmentation multiplicity for our consistently best-performing DP-PEFT, namely, DP-BitFit. Figure 6 shows the impact of M on DP-BitFit, keeping the TPU-hours constant. When $M = 2$, WER_other decreases to 8.6%, which indicates some augmentation multiplicity benefits DP-BitFit. However, $M = 4$ leads to higher WER_other. This is because larger M results in fewer fine-tuning steps, and we see its negative effects outweigh the benefits of larger augmentation multiplicity.

IV. Improvements Using Synthetic Data

In this section, we aim to further improve the utility of DP BitFit. Inspired by recent works [26], [39], [40], [41] showing the benefits of public/synthetic data in DP training, we leverage TTS-generated [42] random-transcript utterances (i.e., synthetic data) to fine-tune the bias terms and decoder prior to DP-BitFit using LibriSpeech. Our intuition is that fine-tuning with synthetic (non-private) data can lead to a better initialization for DP-BitFit.

To this end, we generate synthetic training data as follows. We first sample the 10,000 most frequent words from the transcripts of LibriSpeech test-other, and assume this list of common words is public knowledge that may not need privacy protection. Focusing on only the most common vocabulary words allows us to mimic natural speech patterns while keeping the synthetic data generation process efficient. We then randomly generate 20,000 transcripts, each consisting of 7 words randomly sampled from this top vocabulary set. A 7-word length was chosen because it can capture short natural phrases while still allowing substantial linguistic variety across the 20,000 samples. Finally, we pass these transcripts through a TTS pipeline [42] with four speaker voices (2 male, 2 female) to generate the synthetic utterances.

After getting the synthetic data, we first train both the pre-trained ASR encoder and a randomly initialized ASR decoder using the synthetic data for 3,000 steps. After this initial fine-tuning on the synthetic data, we run DP-BitFit on LibriSpeech.

A. Experimental Results

Figure 7 shows the impact of synthetic data on DP-BitFit. Leveraging synthetic data, the WER_other is further improved to 8.0% for training with a mini-batch size of 2048. This is the best WER_other we manage to achieve for $(10, 3.52\mathrm{e}{-6})$-DP.

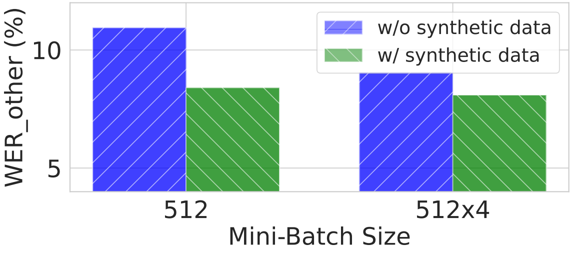


Fig. 7: Impact of synthetic data on DP BitFit when using different mini-batch sizes under $(10, 3.52\mathrm{e}{-6})$-DP.

V. Conclusion

In this work, we extensively study different methods for privately fine-tuning large ASR models. We evaluate DP full fine-tuning and DP-PEFT methods, including DP-Adapter, DP-LoRA, DP-RP, and DP-BitFit, on a state-of-the-art (SOTA) 600M Conformer-based ASR model. Our results demonstrate DP-BitFit achieves the best memory-efficiency as well as model quality. To further improve the privacy-utility trade-off, we propose a novel method to leverage synthetic data. Under a strong privacy guarantee of $(10, 3.52\mathrm{e}{-6})$-DP, our proposed method combining DP-BitFit and synthetic data warmstarting achieves a SOTA WER of 8.0% on LibriSpeech test-other. An interesting future direction is leveraging public text corpuses to generate higher-quality synthetic data, potentially improving privacy-utility trade-offs for training large ASR models with strong DP guarantees even further.

References

[1] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu *et al.*, “Conformer: Convolution-augmented transformer for speech recognition,” in *InterSpeech*, 2020.

[2] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, “wav2vec 2.0: A framework for self-supervised learning of speech representations,” in *NeurIPS*, 2020.

[3] S. Bansal, H. Kamper, K. Livescu, A. Lopez, and S. Goldwater, “Pre-training on high-resource speech recognition improves low-resource speech-to-text translation,” in *NAACL*, 2019.

[4] Y. Zhang, W. Han, J. Qin, Y. Wang, A. Bapna, Z. Chen, N. Chen, B. Li, V. Axelrod, G. Wang, Z. Meng, K. Hu, A. Rosenberg, R. Prabhavalkar, D. S. Park, P. Haghani, J. Riesa, G. Perng, H. Soltau, T. Strohman, B. Ramabhadran, T. N. Sainath, P. J. Moreno, C. Chiu, J. Schalkwyk, F. Beaufays, and Y. Wu, “Google USM: scaling automatic speech recognition beyond 100 languages,” *CoRR*, 2023.

[5] Z. Fan, M. Li, S. Zhou, and B. Xu, “Exploring wav2vec 2.0 on speaker verification and language identification,” in *InterSpeech*, 2021.

[6] N. Vaessen and D. A. Van Leeuwen, “Fine-tuning wav2vec2 for speaker recognition,” in *ICASSP*, 2022.

[7] E. Amid, O. Thakkar, A. Narayanan, R. Mathews, and F. Beaufays, “Extracting targeted training data from asr models, and how to mitigate it,” *Interspeech*, 2022.

[8] L. Wang, O. Thakkar, and R. Mathews, “Unintended memorization in large asr models, and how to mitigate it,” in *ICASSP*, 2024.

[9] M. Jagielski, T. Om, and L. Wang, “Noise masking attacks and defenses for pretrained speech models.” *ICASSP*, 2023.

[10] R. Bassily, A. D. Smith, and A. Thakurta, “Private empirical risk minimization: Efficient algorithms and tight error bounds,” in *FOCS*, 2014.

[11] X. Li, D. Liu, T. B. Hashimoto, H. A. Inan, J. Kulkarni, Y. T. Lee, and A. G. Thakurta, “When does differentially private learning not suffer in high dimensions?” in *NeurIPS*, 2022.

[12] S. De, L. Berrada, J. Hayes, S. L. Smith, and B. Balle, “Unlocking high-accuracy differentially private image classification through scale,” *arXiv*, 2022.

[13] P. Kairouz, B. McMahan, S. Song, O. Thakkar, A. Thakurta, and Z. Xu, “Practical and private (deep) learning without sampling or shuffling,” in *ICML*, 2021.

[14] D. Yu, H. Zhang, W. Chen, and T.-Y. Liu, “Do not let privacy overbill utility: Gradient embedding perturbation for private learning,” *ICLR*, 2021.

[15] D. Yu, S. Naik, A. Backurs, S. Gopi, H. A. Inan, G. Kamath, J. Kulkarni, Y. T. Lee, A. Manoel, L. Wutschitz *et al.*, “Differentially private fine-tuning of language models,” *ICLR*, 2021.

[16] 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 *ICML*, 2019.

[17] E. J. Hu, yelong shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, “LoRA: Low-rank adaptation of large language models,” in *ICLR*, 2022.

[18] E. B. Zaken, S. Ravfogel, and Y. Goldberg, “Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models,” in *ACL*, 2022.

[19] C. Dwork, F. McSherry, K. Nissim, and A. Smith, “Calibrating noise to sensitivity in private data analysis,” in *Theory of Cryptography*, 2006.

[20] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, “Deep learning with differential privacy,” in *CCS*, 2016.

[21] I. Mironov, K. Talwar, and L. Zhang, “Rényi differential privacy of the sampled gaussian mechanism,” *arXiv*, 2019.

[22] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv*, 2014.

[23] Q. Li, B. Li, D. Hwang, T. N. Sainath, and P. M. Mengibar, “Modular domain adaptation for conformer-based streaming asr,” *arXiv*, 2023.

[24] N. Chen, I. Shafran, Y. Zhang, C.-C. Chiu, H. Soltau, J. Qin, and Y. Wu, “Efficient adapters for giant speech models,” *arXiv*, 2023.

[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” *NeurIPS*, 2017.

[26] X. Li, D. Liu, T. Hashimoto, H. A. Inan, J. Kulkarni, Y. Lee, and A. G. Thakurta, “When does differentially private learning not suffer in high dimensions?” in *NeurIPS*, 2022.

[27] D. Yu, H. Zhang, W. Chen, J. Yin, and T.-Y. Liu, “Large scale private learning via low-rank reparametrization,” in *ICML*, 2021.

[28] Z. Bu, Y.-X. Wang, S. Zha, and G. Karypis, “Differentially private bias-term only fine-tuning of foundation models,” *arXiv*, 2022.

[29] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *ICML*, 2015.

[30] Y. Wu and K. He, “Group normalization,” in *ECCV*, 2018.

[31] C.-C. Chiu, J. Qin, Y. Zhang, J. Yu, and Y. Wu, “Self-supervised learning with random-projection quantizer for speech recognition,” in *ICML*, 2022.

[32] J. Kahn, M. Rivière, W. Zheng, E. Kharitonov, Q. Xu, P.-E. Mazaré, J. Karadayi, V. Liptchinsky, R. Collobert, C. Fuegen *et al.*, “Libri-light: A benchmark for asr with limited or no supervision,” in *ICASSP*, 2020.

[33] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, “Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks,” in *ICML*, 2006.

[34] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “Librispeech: an asr corpus based on public domain audio books,” in *ICASSP*, 2015.

[35] "PAXML," https://github.com/google/paxml, 2022.

[36] N. Ponomareva, H. Hazimeh, A. Kurakin, Z. Xu, C. Denison, H. B. McMahan, S. Vassilvitskii, S. Chien, and A. G. Thakurta, "How to dp-fy ml: A practical guide to machine learning with differential privacy," *Journal of Artificial Intelligence Research*, 2023.

[37] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang, "Learning differentially private recurrent language models," *arXiv*, 2017.

[38] P. Kairouz, B. McMahan, S. Song, O. Thakkar, A. Thakurta, and Z. Xu, "Practical and private (deep) learning without sampling or shuffling," in *ICML*, 2021.

[39] 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 *ICML*, 2022.

[40] 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 *ICML*, 2023.

[41] X. Tang, A. Panda, V. Sehwag, and P. Mittal, "Differentially private image classification by learning priors from random processes," *NeurIPS*, 2024.

[42] A. Oord, Y. Li, I. Babuschkin, K. Simonyan, O. Vinyals, K. Kavukcuoglu, G. Driessche, E. Lockhart, L. Cobo, F. Stimberg *et al.*, "Parallel wavenet: Fast high-fidelity speech synthesis," in *ICML*, 2018.