SPARSITY FOR COMMUNICATION-EFFICIENT LORA

***Kevin Kuo, *Arian Raje, †Kousik Rajesh, †Virginia Smith**
*Department of Computer Science, †Department of Machine Learning
Carnegie Mellon University

ABSTRACT

Recently, several works have used unstructured pruning to augment adapter methods. However, these "sparse adapter" methods have limited communication benefits in federated learning (FL). In this work, we propose a simple baseline which combines low-rank adaptation (LoRA) with a constant sparsity during communication only. On three FL image and text tasks, our method reduces communication costs by up to $10\times$ over vanilla LoRA and up to $5\times$ over more complex sparse LoRA baselines while achieving greater utility. Our work highlights the importance of considering system-specific constraints when developing efficient fine-tuning approaches, and serves as a competitive baseline for future work in federated fine-tuning.

1 INTRODUCTION

As pretrained models (e.g. large language models or LLMs) continue to advance state-of-the-art performance in a variety of domains, it is critical to develop methods for efficiently fine-tuning LLMs in low-resource settings. Federated learning (FL) is an increasingly important setting that considers training models across a heterogeneous network of edge devices (McMahan et al., 2017). A primary bottleneck in FL is client-to-server *upload communication*, which scales with the number of trainable model parameters and makes fine-tuning large models prohibitive (Konečný et al., 2017). Although many FL methods based on pruning and quantization have been proposed to solve this issue, *adapters* have emerged as an effective way to reduce costs while retaining the performance of full fine-tuning in both centralized and federated settings (Houlsby et al., 2019; Zhang et al., 2023c).

Recent works in the centralized setting have proposed using *unstructured pruning* to boost the parameter efficiency of adapters by zeroing and freezing a large fraction of the adapter weights (Wu & Chen, 2022; He et al., 2022). However, we show that these schemes for "pruning adapters" transfer poorly to FL because of two key limitations: a) compressing upload is more important than download (Konečný et al., 2017; Ro et al., 2022) and b) weight freezing tends to harm training (Raihan & Aamodt, 2020). To address these issues, we propose to simply apply *sparse communication to low-rank adaptation* (LoRA). Rather than directly pruning the LoRA parameters, our method allows for separate configuration of download and upload sparsity, making it well-suited for FL settings constrained by upload bandwidth. Our work makes the following contributions:

1. To the best of our knowledge, we are the first to apply *unstructured sparsity to LoRA for efficient federated fine-tuning*. We focus on unstructured (weight-level) sparsity because it has been shown to outperform structured (block-level) sparsity in centralized settings (Liu et al., 2018; Siswanto et al., 2021).
2. We identify several limitations of "sparse LoRA" in FL. Primarily, we find that naively pruning methods significantly degrades the model's utility. Furthermore, existing methods use equal sizes for upload and download, resulting in an upload bottleneck.
3. We propose a simple baseline that applies a constant Top-K sparsity only to communication. Our method can reduce communication costs up to $10\times$ while matching the performance of dense LoRA on several FL image and text tasks.

*Direct correspondence to Kevin Kuo: kkuo2@andrew.cmu.edu

2 RELATED WORK

Efficient federated learning. Fine-tuning a large pretrained model is highly useful for both centralized and federated learning (Radford et al., 2018; Nguyen et al., 2022). Many types of methods have been explored to reduce FL communication costs, including quantization (Reisizadeh et al., 2020; Ozkara et al., 2021), sparsity (Caldas et al., 2018b; Horvath et al., 2021; Bibikar et al., 2022; Stripelis et al., 2022; Isik et al., 2022), and parameter-efficient fine-tuning (PET) (Chen et al., 2023; Babakniya et al., 2023a). Prior work has shown that PET methods are surprisingly efficient in FL. For example, LoRA can train a module $100\times$ smaller than the original model, while sparse and quantized FL methods degrade noticeably beyond $10\times$ compression (Qiu et al., 2021; Babakniya et al., 2023b; Ro et al., 2022).

Parameter-efficient fine-tuning (PET) reduces the costs of fine-tuning LLMs by training a small number of parameters and freezing the rest of the model (Ding et al., 2022). In this work, we focus on *low-rank adaptation* (LoRA), a reparameterization-based method which has two advantages: First, unlike prior *adapter* methods, LoRA can be merged with the backbone after training, eliminating additional inference costs (Houlsby et al., 2019; Hu et al., 2021). Second, prior work has shown that LoRA achieves better efficiency-utility trade-offs than other PET methods based on partial backbone fine-tuning (Guo et al., 2021; Zaken et al., 2022; Sung et al., 2021; Gong et al., 2022).

LoRA significantly reduces communication costs in FL. Assuming that all clients have a copy of the pretrained model, only updates to the LoRA modules need to be communicated at each round (Sun et al., 2022; Zhang et al., 2023d). In our work, we consider this "dense LoRA" as a naive baseline and study how to further reduce its message size using sparsity. Orthogonal works on FL and LoRA focus on personalizing the rank or weights to individual clients (Kim et al., 2023b; Yi et al., 2023; Cho et al., 2023). While these methods converge in fewer rounds, they do not focus on reducing per-round communication. Finally, a recent work uses sparsity and quantization to reduce communication when merging PET modules, but is focused on the specific application of one-shot federated learning (Yadav et al., 2023).

Pruning methods set a large fraction of model parameters to zero and compactly represent the model in a sparse matrix format. We study two key algorithms which produce extremely sparse networks while retaining utility. *Iterative magnitude pruning* is a classical baseline which gradually sparsifies the model while re-training the remaining weights (Renda et al., 2019). In contrast, *pruning-at-initialization* performs one-shot pruning followed by a single sparse re-training stage (Lee et al., 2018; Tanaka et al., 2020; Wang et al., 2019).

LoRA with pruning. There has been recent interest in using pruning methods to increase the efficiency of LoRA and vice versa. Most of these methods are based on structured (rank-level) pruning (Ding et al., 2023; Liu et al., 2024). Some search for more efficient ways to allocate ranks between layers (Zhang et al., 2023a; 2022). Others use LoRA to help prune or re-train the model backbone (Zhao et al., 2023; Zhang et al., 2023b; Zhao et al., 2024). To our knowledge, there are two existing methods that extend unstructured pruning to LoRA training (Wu & Chen, 2022; He et al., 2022). While these works improve the storage efficiency of LoRA, they otherwise have marginal practical benefits in centralized settings since:

- **The compute and memory costs of adapters are small** compared to the costs of the backbone (Kim et al., 2023a). Additionally, reparameterization-based PET modules such as LoRA can be merged with the backbone once training is complete (Luo et al., 2023). This eliminates adapter inference costs and makes it less important to produce an extremely sparse adapter.
- **Unstructured sparsity often requires specialized hardware and software** to accelerate computation. Otherwise, training and inference are no more efficient than that of a dense counterpart (Muralidharan, 2023).

Despite these limitations, we argue that combining unstructured sparsity with LoRA is particularly effective for handling issues of communication in FL. Additionally, we show that it is important to carefully incorporate sparsity with LoRA to see benefits in terms of communication. For example, we find that it is highly effective to target upload and download communication at varying rates rather than communicating the same sparse model in both directions. This makes our method a natural choice for practical FL settings where upload speeds are typically much slower than download speeds (up to $8\times$) (Konečný et al., 2017; Lai et al., 2022).

3 METHODS

In our experiments, we fine-tune a pretrained model using **FedAdam** and **LoRA**. On top of this combined method, we apply two baselines **(Adapter LTH, SparseAdapter)** that propose unstructured **pruning** of LoRA in a centralized setting (Wu & Chen, 2022; He et al., 2022).

FedAdam is an adaptive FL method that accelerates convergence by manipulating the aggregated update at the server (Reddi et al., 2020). Given a model with trainable weights W, at each round participating client i will download a copy of W, fine-tune it to obtain updated weights W_i', and upload $\Delta W_i = W - W_i'$ to the server. The server then computes a global update $\Delta W = \frac{1}{n}\sum_{i=1}^{n} \Delta W_i$, where n is the number of clients sampled per round. ΔW can be interpreted as a global pseudo-gradient; for example, the update rule for FedAvg is to set $W \leftarrow W - \Delta W$ for the next round. In the case of FedAdam, the server maintains a stateful Adam optimizer that takes ΔW as input and outputs an adapted global update at each round.

LoRA is a reparameterization-based PET method that updates a weight matrix $W \in \mathbb{R}^{d\times k}$ in a low-rank subspace. The update $\Delta W \in \mathbb{R}^{d\times k}$ is defined as a product BA where $B \in \mathbb{R}^{d\times r}$ and $A \in \mathbb{R}^{r\times k}$. A is initialized using a normal distribution and B is initialized to zero. By freezing W and selecting r to be a small constant, LoRA significantly reduces the number of trainable parameters.

Pruning methods rank the model parameters W according to a scoring function and prune a fraction of parameters with the lowest scores *(i.e. set them to zero and freeze them for the rest of training)*. **In the context of LoRA**, we focus on pruning baselines that leave W both frozen and dense while pruning entries in the adapters A and B. **In the context of FL**, pruning naturally reduces communication costs as *clients do not have to upload or download zeroed/frozen weights*. However, as we show later, applying sparsity only to communication without strictly freezing the model leads to significant gains in performance.

Adapter LTH (Lottery Ticket Hypothesis) iterates between pruning away a small fraction of the lowest magnitude weights and retraining the remaining weights of an adapter module such as LoRA (Frankle & Carbin, 2018; Wu & Chen, 2022). To use this method in FL, we consider training LoRA weights A and B using FedAdam. After each aggregation round, we apply increasingly sparse magnitude pruning to the LoRA weights. We use the efficient "fine-tuning" version of LTH which continues training from the pruned state rather than rewinding the weights after pruning (Renda et al., 2019). This allows the model to recover from pruning within fewer rounds and is necessary to keep communication costs competitive with the dense LoRA baseline.

SparseAdapter generally proposes pruning adapters once at initialization (Wu & Chen, 2022; He et al., 2022). For the choice of parameter scoring function, SNIP (gradient-magnitude product) was found to work the best among other baselines (Lee et al., 2018). However, magnitude-based scoring does not directly extend to LoRA. Because the B matrix in LoRA is initialized to all zeros, pruning below a given density would remove all of the B weights and prevent the LoRA modules from training. To address this issue, we perform an initial round of dense LoRA training, apply magnitude pruning to the aggregated weights, then train the remaining sparse weights as usual.

Algorithm 1: PyTorch-like LoRA training with FedAdam and sparse communication

Require: $d_{\text{down}}, d_{\text{up}}$ (download and upload density)
$P \leftarrow$ Initialize LoRA parameters
`optim` $\leftarrow$ `torch.nn.optim.Adam(params=`P`)`
for $r = 1, ..., R$ **do**
  $M_{\text{down}} \leftarrow$ mask of top d_{down} fraction entries of P by magnitude
  Sample clients $c_1, ..., c_n$ uniformly at random without replacement
  for $i = 1, ..., n$ *in parallel* **do**
    $P_i = P \odot M_{\text{down}}$ # sparse download
    $P_i' \leftarrow$ update P_i with 1 SGD epoch on data of c_i # fine-tuning all entries of P_i
    $\Delta P_i \leftarrow P_i - P_i'$
    $M_{\text{up},i} \leftarrow$ mask of top d_{up} fraction entries of ΔP_i by magnitude
    $\Delta P_i \leftarrow \Delta P_i \odot M_{\text{up},i}$ # sparse upload
  `optim.grad` $\leftarrow \frac{1}{n}\sum_{i=1}^{n} \Delta P_i$ # set Adam pseudo-gradient
  `optim.step()` (updates P) # take one step of Adam

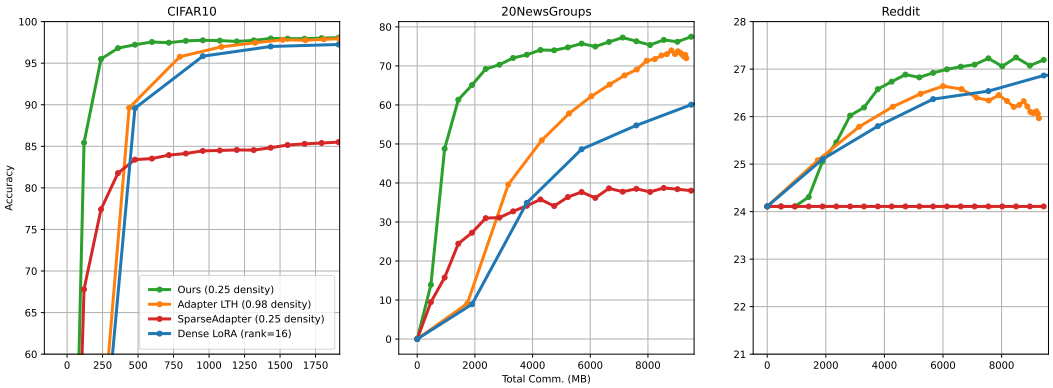

Figure 1: We compare accuracy vs. total communication when augmenting LoRA (rank $r = 16$) with sparsity. Each curve shows the averaged trace over 3 random seeds. Our method improves the overall accuracy-communication trade-offs of Dense LoRA. In contrast, Adapter LTH is inefficient early in training and SparseAdapter fails to reach high accuracy.

Our method is in Algorithm 1. There are three key differences from pruning: a) local fine-tuning uses dense gradients, b) the upload and download sparsity masks can be different, and c) the download mask can change across rounds. For the pruning baselines described above, the same mask M_{down} is applied for all operations, including local SGD steps (red comments, 8-12).

We use P to refer to the flattened and concatenated vector of LoRA weights $\{A_l, B_l\}_{l=1}^{L}$ where L is the number of layers LoRA is applied to. We apply *global* sparsity i.e. retain the Top-K entries of P. An alternative approach is to uniformly sparsify each layer (A_l, B_l) in a *layer-wise* way before concatenation, but we found that global sparsity tended to perform better.

4 Results

We present experiments on three datasets: CIFAR10, 20NewsGroups, and Reddit (Krizhevsky, 2009; Lang, 1995; Caldas et al., 2018a). We resize the CIFAR10 images to 224×224 to match ImageNet, the pretraining dataset for the ViT model architecture we chose. We use the GPT2 tokenizer to preprocess the examples of 20Newsgroups and Reddit into sequences with length 128 and 25 respectively. We partition CIFAR10 and 20NewsGroups using a Dirichlet($\alpha = 0.1$) distribution (Hsu et al., 2019). The Reddit comments are naturally partitioned by user.

In all experiments, we sample 10 clients at each round and perform one epoch of local training with a batch size of 16. We fine-tune all models for 200 rounds. For the pretrained models, we used VIT-B-16 (85M params) and GPT2-Small (124M params) (Dosovitskiy et al., 2021; Radford et al., 2019). For all datasets, we report the accuracy on the validation partition. More details on the task setups can be found in Table 1.

Dataset	Backbone	Task	#Clients	#Examples	#Classes
CIFAR10	ViT-B-16	Image Classification	500	50K	10
20NewsGroups	GPT2-Small	Sequence Classification	350	20K	20
Reddit	GPT2-Small	Next Token Prediction	40K	1.1M	50257

Table 1: Statistics of the datasets used in the experiments.

We used LoRA with rank $r = 16$ for all experiments. In general, smaller ranks tend to be more communication-efficient, while larger ranks tend to achieve higher accuracy. We found that $r = 16$ achieved a good accuracy vs. communication trade-off across the three datasets we considered and thus fixed this for the dense LoRA baseline. To fairly compare against this baseline, we do not re-tune the rank and instead only tune the sparsity of each method to efficiently match dense performance (when possible).

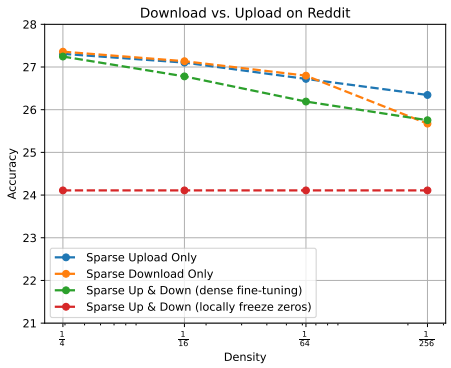

Figure 2: We compare several ways to apply constant Top-K sparsity during FL. Our method is robust to extreme sparsity, while freezing zeroed weights during local fine-tuning (red) fails with relatively little sparsity.

Figure 3: Beyond efficiency in terms of total communication $(1\times)$, our method (green) provides robustness to extremely slow upload speed $(16\times)$. We set $d_{\text{down}} = 0.25$ and adaptively set $d_{\text{up}} = 0.25 * (\text{UL speed reduction})$.

In Figure 1, we measure the total (download + upload) communication from fine-tuning LoRA for 200 FL rounds. We compare our method (Algorithm 1 with $d_{\text{down}} = d_{\text{up}} = 0.25$) to three other baselines (Adapter LTH, SparseAdapter, and Dense LoRA). Adapter LTH and SparseAdapter have limited improvements; Adapter LTH only achieves significant efficiency gains on 20NewsGroups, while SparseAdapter fails to match dense LoRA on all 3 datasets. Our method outperforms dense LoRA using up to $10\times$ less communication than dense LoRA and up to $5\times$ less than Adapter LTH.

In Figure 2, we run an ablation to show the impact of applying sparsity at different stages of an FL round: upload, download, and local fine-tuning. "Sparse Upload" and "Sparse Download" are specific configurations of Algorithm 1, while "Sparse Up and Down" is the configuration presented in Figure 1. By allowing for dense local updates, our method is robust to extremely sparse communication. In contrast, "Freeze Zeros" i.e. SparseAdapter performs poorly even at $\frac{1}{4}$ density, since it additionally constrains Algorithm 1 by freezing all zeroed weights in P_i.

In Figure 3, we assume that all clients have identical network bandwidths and fix the total time spent on communication across all methods (equal to 50 FL rounds of dense LoRA). We then simulate slower upload speeds; for example, under symmetric and constant-per-round communication sizes, a $16\times$ slower upload corresponds to $(1 + 16)/2 = 9.5\times$ fewer FL rounds. To handle slow uploads, our method uses a simple and effective heuristic of setting $d_{\text{up}}, d_{\text{down}}$ to be proportional to their respective bandwidths (e.g. if upload is $16\times$ slower, we set $d_{up} = \frac{d_{\text{down}}}{16}$).

5 Conclusion and Future Work

In this paper, we introduce a communication-efficient FL method that trains LoRA while only sparsifying communication. Our method performs much better than existing pruning-based methods and serves as a strong baseline for future works in federated fine-tuning. Our results show that efficient fine-tuning approaches can be made an order of magnitude more efficient when considering FL constraints, highlighting the importance of tailoring efficiency to the setting at hand.

Still, many important questions remain on how to make LoRA even more efficient in FL. In order to make "communication-efficient" methods truly practical, future work should consider more comprehensive settings that evaluate total training times and realistic bandwidth constraints. Another important question is how to cheaply configure hyperparameters such as the rank and sparsity because they affect both communication and utility. In the future, we aim to investigate such questions and design methods to make high-quality models more accessible to low-resource users.

REFERENCES

Sara Babakniya, Ahmed Elkordy, Yahya Ezzeldin, Qingfeng Liu, Kee-Bong Song, MOSTAFA EL-Khamy, and Salman Avestimehr. SLoRA: Federated parameter efficient fine-tuning of language models. In *International Workshop on Federated Learning in the Age of Foundation Models in Conjunction with NeurIPS 2023*, 2023a. URL https://openreview.net/forum?id=06quMTmtRV.

Sara Babakniya, Souvik Kundu, Saurav Prakash, Yue Niu, and Salman Avestimehr. Revisiting sparsity hunting in federated learning: Why does sparsity consensus matter? *Transactions on Machine Learning Research*, 2023b.

Sameer Bibikar, Haris Vikalo, Zhangyang Wang, and Xiaohan Chen. Federated dynamic sparse training: Computing less, communicating less, yet learning better. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 6080–6088, 2022.

Sebastian Caldas, Sai Meher Karthik Duddu, Peter Wu, Tian Li, Jakub Konečnỳ, H Brendan McMahan, Virginia Smith, and Ameet Talwalkar. Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097*, 2018a.

Sebastian Caldas, Jakub Konečny, H Brendan McMahan, and Ameet Talwalkar. Expanding the reach of federated learning by reducing client resource requirements. *arXiv preprint arXiv:1812.07210*, 2018b.

Chaochao Chen, Xiaohua Feng, Jun Zhou, Jianwei Yin, and Xiaolin Zheng. Federated large language model: A position paper. *arXiv preprint arXiv:2307.08925*, 2023.

Yae Jee Cho, Luyang Liu, Zheng Xu, Aldi Fahrezi, Matt Barnes, and Gauri Joshi. Heterogeneous lora for federated fine-tuning of on-device foundation models. In *International Workshop on Federated Learning in the Age of Foundation Models in Conjunction with NeurIPS 2023*, 2023.

Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, et al. Delta tuning: A comprehensive study of parameter efficient methods for pre-trained language models. *arXiv preprint arXiv:2203.06904*, 2022.

Ning Ding, Xingtai Lv, Qiaosen Wang, Yulin Chen, Bowen Zhou, Zhiyuan Liu, and Maosong Sun. Sparse low-rank adaptation of pre-trained language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 4133–4145, 2023.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*, 2018.

Zhuocheng Gong, Di He, Yelong Shen, Tie-Yan Liu, Weizhu Chen, Dongyan Zhao, Ji-Rong Wen, and Rui Yan. Finding the dominant winning ticket in pre-trained language models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 1459–1472, 2022.

Demi Guo, Alexander M Rush, and Yoon Kim. Parameter-efficient transfer learning with diff pruning. 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)*, pp. 4884–4896, 2021.

Shwai He, Liang Ding, Daize Dong, Jeremy Zhang, and Dacheng Tao. Sparseadapter: An easy approach for improving the parameter-efficiency of adapters. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 2184–2190, 2022.

Samuel Horvath, Stefanos Laskaridis, Mario Almeida, Ilias Leontiadis, Stylianos Venieris, and Nicholas Lane. Fjord: Fair and accurate federated learning under heterogeneous targets with ordered dropout. *Advances in Neural Information Processing Systems*, 34:12876–12889, 2021.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pp. 2790–2799. PMLR, 2019.

Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Berivan Isik, Francesco Pase, Deniz Gunduz, Tsachy Weissman, and Michele Zorzi. Sparse random networks for communication-efficient federated learning. *arXiv preprint arXiv:2209.15328*, 2022.

Jeonghoon Kim, Jung Hyun Lee, Sungdong Kim, Joonsuk Park, Kang Min Yoo, Se Jung Kwon, and Dongsoo Lee. Memory-efficient fine-tuning of compressed large language models via sub-4-bit integer quantization. *arXiv preprint arXiv:2305.14152*, 2023a.

Yeachan Kim, Junho Kim, Wing-Lam Mok, Jun-Hyung Park, and SangKeun Lee. Client-customized adaptation for parameter-efficient federated learning. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 1159–1172, 2023b.

Jakub Konečný, H. Brendan McMahan, Felix X. Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency, 2017.

Alex Krizhevsky. Learning multiple layers of features from tiny images. *Master's thesis, University of Toronto*, 2009.

Fan Lai, Yinwei Dai, Sanjay Singapuram, Jiachen Liu, Xiangfeng Zhu, Harsha Madhyastha, and Mosharaf Chowdhury. Fedscale: Benchmarking model and system performance of federated learning at scale. In *International Conference on Machine Learning*, pp. 11814–11827. PMLR, 2022.

Ken Lang. Newsweeder: Learning to filter netnews. In *Machine learning proceedings 1995*, pp. 331–339. Elsevier, 1995.

Namhoon Lee, Thalaiyasingam Ajanthan, and Philip Torr. Snip: Single-shot network pruning based on connection sensitivity. In *International Conference on Learning Representations*, 2018.

Wei Liu, Ying Qin, Zhiyuan Peng, and Tan Lee. Sparsely shared lora on whisper for child speech recognition. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 11751–11755. IEEE, 2024.

Zhuang Liu, Mingjie Sun, Tinghui Zhou, Gao Huang, and Trevor Darrell. Rethinking the value of network pruning. In *International Conference on Learning Representations*, 2018.

Gen Luo, Minglang Huang, Yiyi Zhou, Xiaoshuai Sun, Guannan Jiang, Zhiyu Wang, and Rongrong Ji. Towards efficient visual adaption via structural re-parameterization. *arXiv preprint arXiv:2302.08106*, 2023.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pp. 1273–1282. PMLR, 2017.

Saurav Muralidharan. Uniform sparsity in deep neural networks. *Proceedings of Machine Learning and Systems*, 5, 2023.

John Nguyen, Jianyu Wang, Kshitiz Malik, Maziar Sanjabi, and Michael Rabbat. Where to begin? on the impact of pre-training and initialization in federated learning. In *Workshop on Federated Learning: Recent Advances and New Challenges (in Conjunction with NeurIPS 2022)*, 2022.

Kaan Ozkara, Navjot Singh, Deepesh Data, and Suhas Diggavi. Quped: Quantized personalization via distillation with applications to federated learning. *Advances in Neural Information Processing Systems*, 34:3622–3634, 2021.

Xinchi Qiu, Javier Fernandez-Marques, Pedro PB Gusmao, Yan Gao, Titouan Parcollet, and Nicholas Donald Lane. Zerofl: Efficient on-device training for federated learning with local sparsity. In *International Conference on Learning Representations*, 2021.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Md Aamir Raihan and Tor Aamodt. Sparse weight activation training. *Advances in Neural Information Processing Systems*, 33:15625–15638, 2020.

Sashank J Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečnỳ, Sanjiv Kumar, and Hugh Brendan McMahan. Adaptive federated optimization. In *International Conference on Learning Representations*, 2020.

Amirhossein Reisizadeh, Aryan Mokhtari, Hamed Hassani, Ali Jadbabaie, and Ramtin Pedarsani. Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization. In *International Conference on Artificial Intelligence and Statistics*, pp. 2021–2031. PMLR, 2020.

Alex Renda, Jonathan Frankle, and Michael Carbin. Comparing rewinding and fine-tuning in neural network pruning. In *International Conference on Learning Representations*, 2019.

Jae Hun Ro, Theresa Breiner, Lara McConnaughey, Mingqing Chen, Ananda Theertha Suresh, Shankar Kumar, and Rajiv Mathews. Scaling language model size in cross-device federated learning. *arXiv preprint arXiv:2204.09715*, 2022.

Arlene Siswanto, Jonathan Frankle, and Michael Carbin. Reconciling sparse and structured pruning: A scientific study of block sparsity. In *Workshop paper at the 9th International Conference on Learning Representations (ICLR 2021)*, 2021.

Dimitris Stripelis, Umang Gupta, Greg Ver Steeg, and Jose Luis Ambite. Federated progressive sparsification (purge-merge-tune)+. In *Workshop on Federated Learning: Recent Advances and New Challenges (in Conjunction with NeurIPS 2022)*, 2022.

Guangyu Sun, Matias Mendieta, Taojiannan Yang, and Chen Chen. Conquering the communication constraints to enable large pre-trained models in federated learning. *arXiv preprint arXiv:2210.01708*, 2022.

Yi-Lin Sung, Varun Nair, and Colin A Raffel. Training neural networks with fixed sparse masks. *Advances in Neural Information Processing Systems*, 34:24193–24205, 2021.

Hidenori Tanaka, Daniel Kunin, Daniel L Yamins, and Surya Ganguli. Pruning neural networks without any data by iteratively conserving synaptic flow. *Advances in neural information processing systems*, 33:6377–6389, 2020.

Chaoqi Wang, Guodong Zhang, and Roger Grosse. Picking winning tickets before training by preserving gradient flow. In *International Conference on Learning Representations*, 2019.

Jiarun Wu and Qingliang Chen. Pruning adapters with lottery ticket. *Algorithms*, 15(2):63, 2022.

Prateek Yadav, Leshem Choshen, Colin Raffel, and Mohit Bansal. Compeft: Compression for communicating parameter efficient updates via sparsification and quantization. *arXiv preprint arXiv:2311.13171*, 2023.

Liping Yi, Han Yu, Gang Wang, and Xiaoguang Liu. Fedlora: Model-heterogeneous personalized federated learning with lora tuning. *arXiv preprint arXiv:2310.13283*, 2023.

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. 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)*, pp. 1–9, 2022.

Feiyu Zhang, Liangzhi Li, Junhao Chen, Zhouqiang Jiang, Bowen Wang, and Yiming Qian. Increlora: Incremental parameter allocation method for parameter-efficient fine-tuning. *arXiv preprint arXiv:2308.12043*, 2023a.

Mingyang Zhang, Chunhua Shen, Zhen Yang, Linlin Ou, Xinyi Yu, Bohan Zhuang, et al. Pruning meets low-rank parameter-efficient fine-tuning. *arXiv preprint arXiv:2305.18403*, 2023b.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adaptive budget allocation for parameter-efficient fine-tuning. In *The Eleventh International Conference on Learning Representations*, 2022.

Xuechen Zhang, Mingchen Li, Xiangyu Chang, Jiasi Chen, Amit K Roy-Chowdhury, Ananda Theertha Suresh, and Samet Oymak. Fedyolo: Augmenting federated learning with pretrained transformers. *arXiv preprint arXiv:2307.04905*, 2023c.

Zhuo Zhang, Yuanhang Yang, Yong Dai, Qifan Wang, Yue Yu, Lizhen Qu, and Zenglin Xu. Fedpetuning: When federated learning meets the parameter-efficient tuning methods of pre-trained language models. In *Annual Meeting of the Association of Computational Linguistics 2023*, pp. 9963–9977. Association for Computational Linguistics (ACL), 2023d.

Bowen Zhao, Hannaneh Hajishirzi, and Qingqing Cao. Apt: Adaptive pruning and tuning pretrained language models for efficient training and inference. *arXiv preprint arXiv:2401.12200*, 2024.

Weilin Zhao, Yuxiang Huang, Xu Han, Zhiyuan Liu, Zhengyan Zhang, and Maosong Sun. Cpet: Effective parameter-efficient tuning for compressed large language models. *arXiv preprint arXiv:2307.07705*, 2023.

A APPENDIX

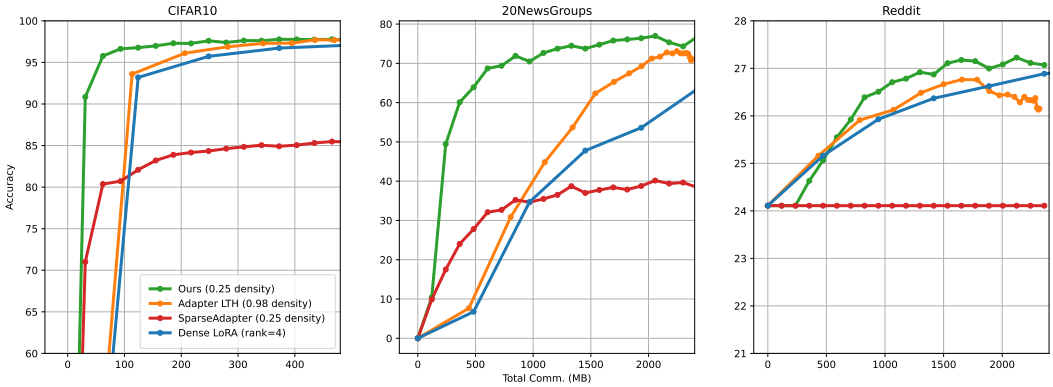

Figure 4: Comparison of total communication with LoRA rank 4.

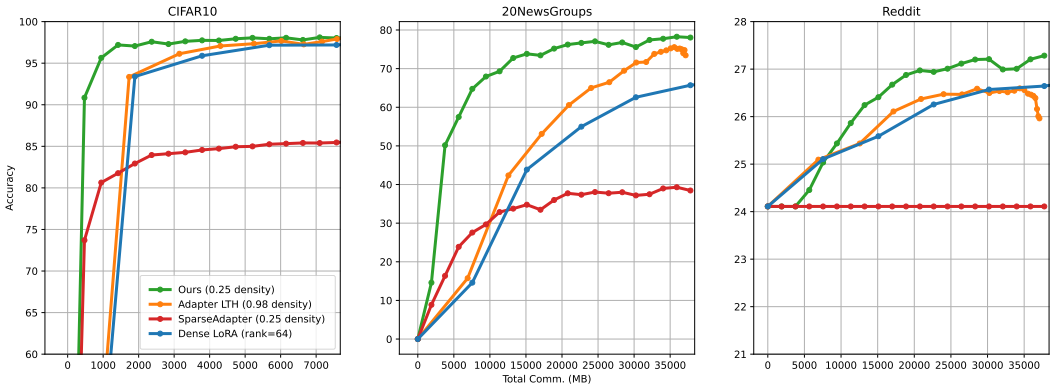

Figure 5: Comparison of total communication with LoRA rank 64.

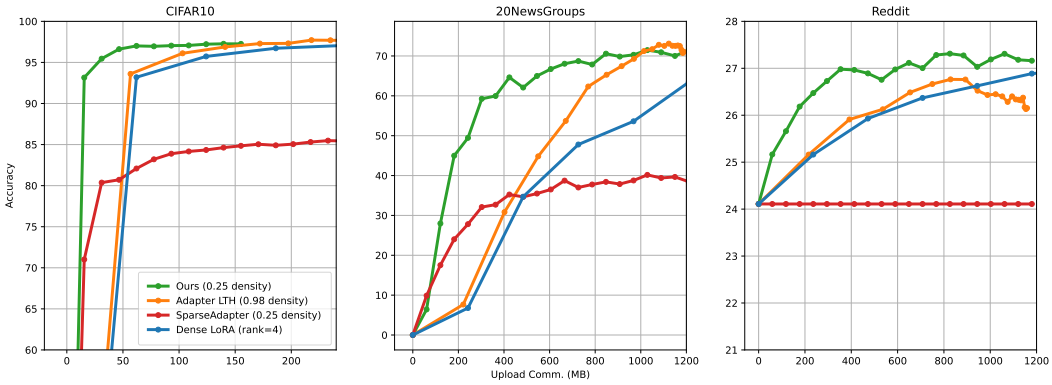

Figure 6: Comparison of upload communication with LoRA rank 4.

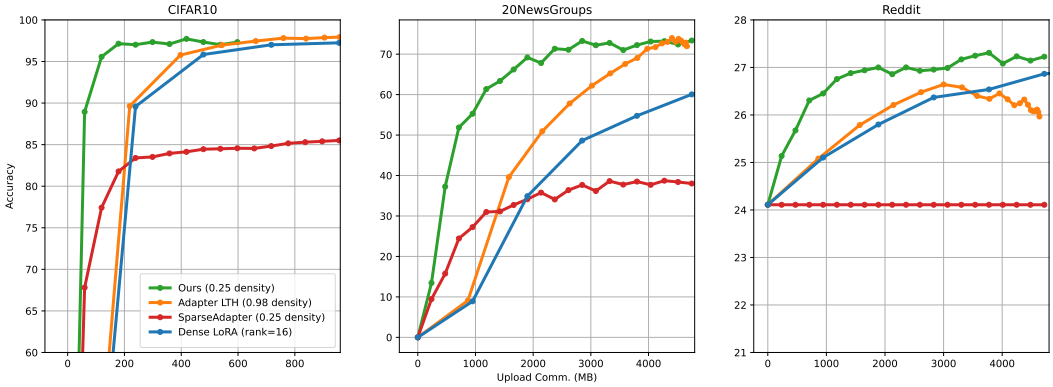

Figure 7: Comparison of upload communication with LoRA rank 16.

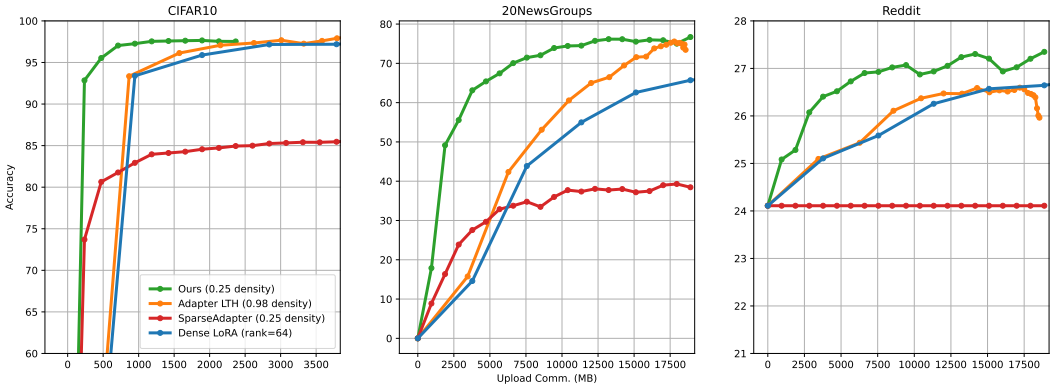

Figure 8: Comparison of upload communication with LoRA rank 64.