Towards Federated Learning Under Resource Constraints via Layer-wise Training and Depth Dropout

Pengfei Guo[*1], Warren Richard Morningstar[2], Raviteja Vemulapalli[2], Karan Singhal[2], Vishal M. Patel[1], and Philip Andrew Mansfield[2]

[1]*Johns Hopkins University*
[2]*Google Research*
{*pguo4,vpatel36*}*@jhu.edu,*
{*wmorning,ravitejavemu,karansinghal,memes*}*@google.com*

ABSTRACT

Large machine learning models trained on diverse data have recently seen unprecedented success. Federated learning enables training on private data that may otherwise be inaccessible, such as domain-specific datasets decentralized across many clients. However, federated learning can be difficult to scale to large models when clients have limited resources. This challenge often results in a trade-off between model size and access to diverse data. To mitigate this issue and facilitate training of large models on edge devices, we introduce a simple yet effective strategy, *Federated Layer-wise Learning*, to simultaneously reduce per-client memory, computation, and communication costs. Clients train just a single layer each round, reducing resource costs considerably with minimal performance degradation. We also introduce *Federated Depth Dropout*, a complementary technique that randomly drops frozen layers during training, to further reduce resource usage. Coupling these two techniques enables us to effectively train significantly larger models on edge devices. Specifically, we reduce training memory usage by $5\times$ or more in federated self-supervised representation learning, and demonstrate that performance in downstream tasks is comparable to conventional federated self-supervised learning.

1 INTRODUCTION

Over the last several years, deep learning has witnessed a rapid paradigm shift towards large foundational models trained on massive datasets (Brown et al., 2020; Chowdhery et al., 2022). These models learn representations which often extend to diverse downstream tasks. However, when pre-training data is distributed across a large number of devices, it becomes impractical to train models using centralized learning. In these cases, Federated Learning (FL; Konečný et al., 2016) allows participating clients to train a model together without exchanging raw data. This privacy-preserving property makes FL a popular choice for a range of applications, including face recognition (Mei et al., 2022a), autonomous driving (Li et al., 2021), recommendation systems (Ning et al., 2021), and self-supervised representation learning (Vemulapalli et al., 2022). In self-supervised learning, SimCLR (Chen et al., 2020), BYOL (Grill et al., 2020), and SimSiam (Chen & He, 2021) are widely used approaches that can be adapted for use in FL settings using algorithms like Federated Averaging (FedAvg; McMahan et al., 2017). Representation learning benefits from large models due to their capacity to learn more nuanced and reliable representations of the data (Chen et al., 2022; Tran et al., 2022). However, in cross-device FL settings, the limited resources of edge devices (including memory, computation capacity, and network bandwidth) impedes the development of large models (Wang et al., 2021; Kairouz et al., 2021). In this work, we focus on federated training of large representation learning models on a large number of edge devices under resource constraints.

---

[*]Work done during an internship at Google Research.

Typically in FL, clients' models share a single global architecture and perform end-to-end training in each communication round (McMahan et al., 2017). However, many edge devices (e.g., Internet of Things (IoT) devices, mobile phones, tablets, and personal computers) lack sufficient memory and compute to train most existing large ML models. For example, the Google Pixel 6 has 12 GB of memory, which is insufficient to naively train a multi-billion parameter model. Communication of such a model and its gradient updates during every round of FL is also prohibitively data-intensive and time-consuming. These resource constraints create obstacles for real-world federated learning applications with large-scale models.

Related Work   One direction to manage resource constraints for federated learning on edge devices is to carefully select model architecture and hyperparameters (Cheng et al., 2022) to ensure that it can be trained and run efficiently on edge devices. Another direction is to use techniques such as model compression (Xu et al., 2020) and pruning (Jiang et al., 2022) to reduce the size and complexity of the model, making it more suited for training and deployment on edge devices. This can be done by removing redundant or unnecessary layers within the model, or by using low-precision arithmetic to reduce the amount of memory and computation required. In both cases, model performance degradation is usually unavoidable. Some methods rely on partially local models to avoid communicating entire models with a central server (Singhal et al., 2021), but these approaches do not reduce local memory usage on edge devices. Other approaches involve retaining part of a model on a central server (Augenstein et al., 2022), which can reduce the need for local resource usage and is thus complementary to our work but does not itself enable training larger local models. In addition, FL methods designed for heterogeneous systems (Caldas et al., 2018; Horvath et al., 2021; Mei et al., 2022b) are able to construct sub-models at different complexities from one unified base model. One of the early works in this direction is Federated Dropout (Caldas et al., 2018), which allows users to train using smaller subsets of the global model, reducing the client communication and computation costs. Empirically, this method can drop up to 50% of model parameters, but will degrade model performance. FjORD (Horvath et al., 2021) improves upon Federated Dropout by introducing an ordered dropout technique that drops adjacent components of the model instead of random neurons. Experiments by Horvath et al. (2021) show that ordered dropout can bring computational benefits and better model performance. More recently, FLANC (Mei et al., 2022b) formulates networks at different capacities as linear combinations of a shared neural basis set, so sub-models can be composed by using capacity-specific coefficients. While these methods can reduce average local resource usage in FL, full model training is still needed for certain clients, and thus the resource usage upper bound is still determined by the base model size.

Our Contributions   Full utilization of available resources in cross-device FL remains a challenging task. In this paper, we propose *Federated Layer-wise Learning*, a strategy for resource-saving federated training. In particular, training is divided into several phases. In each phase, we update only one active layer and freeze parameters in fixed layers. As shown in Fig. 2(b), our experimental evaluation demonstrates that Federated Layer-wise Learning (FLL) can significantly reduce the resource usage of a single client compared to federated end-to-end learning (FEL) in all aspects. Specifically, FLL only uses 7–22% memory, 8–39% computation, and 8–54% communication compared to FEL. In addition, we demonstrate that *Depth Dropout* is an effective complementary strategy in federated layer-wise learning, which further reduces resource usage upper bounds without degrading model performance.

2   METHODS

We consider the canonical cross-device FL scenario, in which a large distributed population of clients contributes to training a global model using their locally stored data (Kairouz et al., 2021). In such scenarios, the general training process involves the following steps: first, a coordinating server sends the current set of model parameters to each contributing device. Next, each device runs a local training algorithm and sends the result back to the server. Finally, the server aggregates the model updates received from all devices to determine the new set of model parameters and restarts the cycle. As previously discussed, device resource constraints limit real-world large-scale federated learning applications and lead to a trade-off between model complexity and data accessibility.
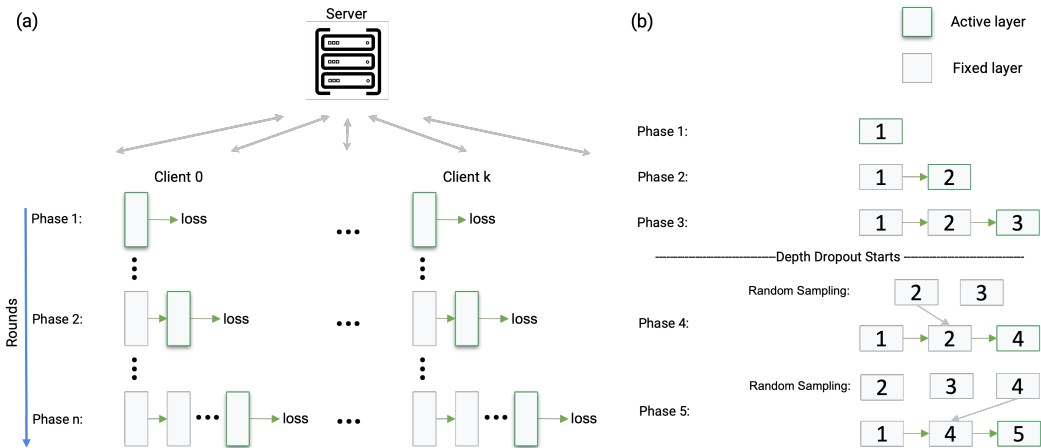
Figure 1: (a) Overview of Federated Layer-wise Learning. (b) Schematics of training procedure with Depth Dropout for a 5-layer model with a budget of 3 layers.

2.1 FEDERATED LAYER-WISE LEARNING

To address resource constraints when training models in FL, we propose a simple yet effective Federated Layer-wise Learning technique. We motivate and apply the method to self-supervised learning (Chen et al., 2020) in this work, but this approach is also broadly applicable. In contrast to downstream vision tasks (*e.g.*, classification) that require the extraction of compact features (*i.e.*, the interpretation of input) from the output of neural networks, contrastive representation learning employs the principle of learning representations of the input data by comparing and contrasting it with other similar and dissimilar examples. Since this loss only refers to layer activations, it can be attached to any encoder layer. When applied to residual networks, we expect the effect of applying the loss on successive layers to be progressive. This motivates our Federated Layer-wise Learning method, as depicted in Fig. 1(a).

The proposed method divides the holistic training process into several phases and progressively grows the model in an incremental schedule, starting from the shallow layers and moving to deeper layers. Each layer is trained for a predefined number of communication rounds before proceeding to the next layer. We only need to compute gradients and upload them to the server for the active layer, which simultaneously reduces memory usage, compute, and communication costs. We can control resource usage by varying the number of active and fixed layers during training, potentially treating multiple layers as active at a given round. As an aside, in cross-silo FL (where clients participate repeatedly in training) the fact that only one active layer is being trained enables us to avoid communicating the rest of the model to devices on most rounds, further reducing communication.

2.2 DEPTH DROPOUT

While the proposed Federated Layer-wise Learning significantly alleviates resource usage, our target scenario is cross-device FL, in which only a relatively small subset of active clients are selected from a large pool of participants. It is likely that a given client will not be selected twice during the entire federated training process (Wang et al., 2021). Thus, it is necessary to download both fixed and active layers from the server to the clients. This can still present a challenge for clients with resource constraints, as downloading a large number of fixed layers and performing forward passes can be computationally intensive at the end of the training process. To this end, we propose *Depth Dropout* to address the increasing resource usage introduced by a large number of fixed layers. Fig. 1(b) shows how to apply Depth Dropout to a 5-layer model with a budget of 3 layers. It begins by progressively expanding the model to reach its maximum capacity, which in this case is 3 layers. During the initial three phases, we perform standard layer-wise training. In the last two phases, we randomly remove certain fixed layers. However, the first layer, which includes Transformer patch encoding and position embedding, is never removed. For example, in phase 4, we have the option to remove either layer 1 or layer 2, while in phase 5, we have three candidates to remove. This

randomization process is akin to the Dropout technique used in neural networks, and is only applied during training. During inference, the full model with 5 layers is utilized.

3 EXPERIMENTS

Datasets and Implementation   We partition the standard CIFAR-100 (Krizhevsky et al., 2009) training set into 125 clients to simulate a cross-device FL scenario. The original test set in CIFAR-100 (Krizhevsky et al., 2009) is considered the global test set used to measure performance. ViT-Ti/16 (Dosovitskiy et al., 2020) is used as the representation learning backbone. All models are trained using the following settings: SGD optimizer for the server and clients; client learning rate of 1×10^{-3}; batch size of 16; 32 active clients per round.

The Effectiveness of Layer-wise Training   Here we compare our approach with federated end-to-end learning on standard benchmarks. Results with different setups on CIFAR-100 are shown in Table 1. We can make the following observations: (i) Both models pre-trained by Federated Layer-wise Learning and federated end-to-end learning can significantly outperform the model without pre-training, indicating the effectiveness of self-supervised representation learning in federated settings. (ii) While the Federated Layer-wise Learning approach is an approximation of federated end-to-end learning, it can achieve performance on par with the end-to-end method in downstream evaluation tasks. In particular, the performance gap is less than 1% when using the representation from the last layer (layer 12) of the network. (iii) we found that intermediate representations from the Federated Layer-wise Learning model performed better than those from the federated end-to-end learning model in certain downstream tasks. For example, in a linear downstream task using the representation from layer 3, the Federated Layer-wise Learning model achieved 28.3% accuracy, while the federated end-to-end learning model achieved 23.9%. This trend was also observed in other downstream tasks using different intermediate representations. This superior performance of intermediate representations is due to the contrastive loss being applied to all layers during the layer-wise pre-training process. These results suggest that models trained using the proposed method can easily compose sub-models of varying complexities.

We conducted additional experiments to further investigate the effect of model size (number of layers) and number of training rounds per layer on the performance of Federated Layer-wise and end-to-end learning. The results of these experiments are shown in Figure 2(a). Increasing the number of layers generally led to improved performance for both learning approaches. We also found that the difference in performance between the two approaches was minimal when the number of training rounds per layer was small (4k) but became more pronounced when the number of training rounds per layer was increased (12k). Based on these results, it appears that layer-wise learning may require slightly more training rounds per layer to reach the same performance as end-to-end learning. This may be due to the fact that layer-wise learning is an approximation of end-to-end learning. However, the performance gap between the two approaches is generally less than 1%.

Table 1: Experimental results on CIFAR-100 with different pre-training strategies. For image classification, we report standard Top-1 accuracy (%).

Downstream Representation From	Linear				Finetune			
	Layer 1	Layer 3	Layer 6	Layer 12	Layer 1	Layer 3	Layer 6	Layer 12
Pre-training Method	Federated Layer-wise Learning							
Accuracy	25.3	28.3	29.2	29.8	30.1	35.6	37.2	37.8
Pre-training Method	Federated End-to-end Learning							
Accuracy	18.0	23.9	27.8	30.3	25.4	32.2	35.7	38.6
Pre-training Method	Training from scratch (Without Pre-training)							
Accuracy	9.2	10.0	10.5	11.3	18.7	24.2	27.6	29.2

The Effectiveness of Depth Dropout.   We evaluate Depth Dropout with Federated Layer-wise Learning. We conducted two sets of experiments: applying Depth Dropout to a 6-layer model and a 12-layer model, with a fixed dropout rate of 50% (meaning half of the fixed layers were dropped). The results, shown in Tables 2 and 3, demonstrate that Depth Dropout does not significantly impact model performance. For example, the 6-layer model with Depth Dropout achieved 37.0% accuracy
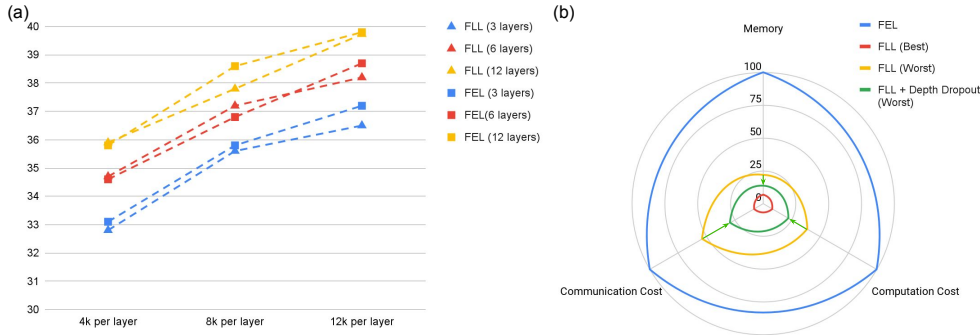
Figure 2: (a) Comparison between Federated Layer-wise Learning (FLL) and federated end-to-end learning (FEL) under different numbers of training rounds and model sizes. Results are reported on CIFAR-100 with downstream finetuning evaluation. (b) Resource usage comparison of a client per-round after combining Federated Layer-wise Learning and Depth Dropout.

Table 2: Accuracy of depth dropout with Federated Layer-wise Learning, under finetuning downstream evaluation. The budget specifies the max number of layers involved in training.

Model Size	6 layers	6 layers (Budget: 3 layers)	3 layers	12 layers	12 layers (Budget: 6 layers)	6 layers
Accuracy	37.2	37.0	32.8	37.8	37.6	37.2

Table 3: Accuracy of depth dropout with Federated Layer-wise Learning, under linear downstream evaluation. The budget specifies the max number of layers involved in training.

Model Size	6 layers	6 layers (Budget: 3 layers)	3 layers	12 layers	12 layers (Budget: 6 layers)	6 layers
Accuracy	29.2	29.1	28.3	29.8	29.7	29.2

after finetuning, while the 6-layer model trained with only Layer-wise Learning achieved 37.2% accuracy. We observed similar results for the 12-layer model with Depth Dropout, which achieved 37.6% accuracy after finetuning, compared to 37.8% for the model trained with normal Layer-wise Learning. Additionally, Depth Dropout significantly reduced resource usage. It is worth noting that the resource usage of the 12-layer model with a budget of 6 layers was equivalent to the resource usage of a 6-layer model without Depth Dropout. As shown in Fig. 2, depth dropout reduced the upper bounds of resource usage in all three categories, especially communication cost. The original upper bound for Layer-wise training was 54%, but it was reduced to 29% when the dropout rate was set to 50%.

4 CONCLUSION

Our study presents Federated Layer-wise Learning for devices with limited resources, which simultaneously reduces the demands on memory, computation, and communication for individual clients without significantly compromising performance in comparison to end-to-end training. We demonstrate that our proposed Depth Dropout technique is an effective complement to Federated Layer-wise Learning, as it further reduces resource usage across all categories with minimal loss of performance, even when dropping half of the fixed layers. Future work can evaluate these methods on larger-scale and naturally partitioned datasets, which would enable more realistic analysis of generalization performance across devices (Yuan et al., 2021). Additionally, we intend to investigate the effects of varying dropout rate for the Depth Dropout technique. Furthermore, our method can be integrated with other memory-efficient training techniques, such as model compression (Deng et al., 2020) and activation paging (Patil et al., 2022) to potentially further reduce resource usage.

REFERENCES

Sean Augenstein, Andrew Hard, Lin Ning, Karan Singhal, Satyen Kale, Kurt Partridge, and Rajiv Mathews. Mixed federated learning: Joint decentralized and centralized learning. *arXiv preprint arXiv:2205.13655*, 2022.

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

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*, 2018.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, et al. Pali: A jointly-scaled multilingual language-image model. *arXiv preprint arXiv:2209.06794*, 2022.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15750–15758, 2021.

Gary Cheng, Zachary Charles, Zachary Garrett, and Keith Rush. Does federated dropout actually work? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3387–3395, 2022.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

Lei Deng, Guoqi Li, Song Han, Luping Shi, and Yuan Xie. Model compression and hardware acceleration for neural networks: A comprehensive survey. *Proceedings of the IEEE*, 108(4): 485–532, 2020.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

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.

Yuang Jiang, Shiqiang Wang, Victor Valls, Bong Jun Ko, Wei-Han Lee, Kin K Leung, and Leandros Tassiulas. Model pruning enables efficient federated learning on edge devices. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista A. Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Hubert Eichner, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaïd Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konečný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Hang Qi, Daniel Ramage, Ramesh Raskar, Mariana Raykova, Dawn Song, Weikang Song,

Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. Advances and open problems in federated learning. *Found. Trends Mach. Learn.*, 14(1-2):1–210, 2021.

Jakub Konečný, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Yijing Li, Xiaofeng Tao, Xuefei Zhang, Junjie Liu, and Jin Xu. Privacy-preserved federated learning for autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 2021.

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.

Yiqun Mei, Pengfei Guo, and Vishal M Patel. Escaping data scarcity for high-resolution heterogeneous face hallucination. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18676–18686, 2022a.

Yiqun Mei, Pengfei Guo, Mo Zhou, and Vishal Patel. Resource-adaptive federated learning with all-in-one neural composition. In *Advances in Neural Information Processing Systems*, 2022b.

Lin Ning, Karan Singhal, Ellie X Zhou, and Sushant Prakash. Learning federated representations and recommendations with limited negatives. *arXiv preprint arXiv:2108.07931*, 2021.

Shishir G Patil, Paras Jain, Prabal Dutta, Ion Stoica, and Joseph Gonzalez. Poet: Training neural networks on tiny devices with integrated rematerialization and paging. In *International Conference on Machine Learning*, pp. 17573–17583. PMLR, 2022.

Karan Singhal, Hakim Sidahmed, Zachary Garrett, Shanshan Wu, John Rush, and Sushant Prakash. Federated reconstruction: Partially local federated learning. *Advances in Neural Information Processing Systems*, 34:11220–11232, 2021.

Dustin Tran, Jeremiah Liu, Michael W Dusenberry, Du Phan, Mark Collier, Jie Ren, Kehang Han, Zi Wang, Zelda Mariet, Huiyi Hu, et al. Plex: Towards reliability using pretrained large model extensions. *arXiv preprint arXiv:2207.07411*, 2022.

Raviteja Vemulapalli, Warren Richard Morningstar, Philip Andrew Mansfield, Hubert Eichner, Karan Singhal, Arash Afkanpour, and Bradley Green. Federated training of dual encoding models on small non-iid client datasets. *arXiv preprint arXiv:2210.00092*, 2022.

Jianyu Wang, Zachary Charles, Zheng Xu, Gauri Joshi, H Brendan McMahan, Maruan Al-Shedivat, Galen Andrew, Salman Avestimehr, Katharine Daly, Deepesh Data, et al. A field guide to federated optimization. *arXiv preprint arXiv:2107.06917*, 2021.

Jinjin Xu, Wenli Du, Yaochu Jin, Wangli He, and Ran Cheng. Ternary compression for communication-efficient federated learning. *IEEE Transactions on Neural Networks and Learning Systems*, 33(3):1162–1176, 2020.

Honglin Yuan, Warren Morningstar, Lin Ning, and Karan Singhal. What do we mean by generalization in federated learning? *arXiv preprint arXiv:2110.14216*, 2021.