FedRolex: Model-Heterogeneous Federated Learning with Rolling Sub-Model Extraction

MICHIGAN STATE UNIVERSITY
THE OHIO STATE UNIVERSITY
Google Research
香港中文大學（深圳） The Chinese University of Hong Kong, Shenzhen

Samiul Alam[1,2] Luyang Liu[3] Ming Yan[4,1] Mi Zhang[2,1]

[1]Michigan State University [2]The Ohio State University [3]Google Research [4]The Chinese University of Hong Kong, Shenzhen

Introduction

The majority of existing cross-device FL studies focus on the *model-homogeneous* setting [11, 9, 8, 2] , in which the server model and the client models across all the participating client devices are *identical*. However, model-homogeneous FL has two fundamental constraints: (1) It excludes clients with low-end devices who could otherwise make unique contributions to model training from their own local data. (2) Restricting server and client models to be the same inevitably causes model-homogeneous FL to fail to train **large models** due to the resource constraint of client devices. To relax the fundamental constraints of model-homogeneous FL, in this work, we propose a **model-heterogeneous** FL approach where heterogeneous models with different capacities across the server and the clients are trained during the federated training process.

Related Work

Existing works on model-heterogeneous FL can be generally categorized into knowledge distillation (KD)-based and partial training (PT)-based methods.

Knowledge Distillation (KD). One category of approaches used Knowledge Distillation (KD) [5, 10, 7, 3], where the client models serve as teachers, and the server ensembles the knowledge distilled from the individual client models. However, these methods require public data to achieve competitive accuracy and are incompatible with secure aggregation protocols.

Partial Training (PT). In partial training (PT)-based approaches, each client trains a smaller sub-model extracted from the larger global server model, and the server model is updated by aggregating those trained sub-models. Depending on how the sub-models are extracted from the global server model, existing PT-based methods can be in general categorized into two groups: **random** sub-model extraction [1] and **static** sub-model extraction [4, 6]. PT-based algorithms overcome the issues of KD-based approaches. However, the fundamental issue of existing PT-based methods is that the sub-models are extracted in ways such that the parameters of the global server model are not evenly trained. This makes the server model vulnerable to client drift induced by the inconsistency between individual client model and server model architectures.

Table 1:Comparison of `FedRolex` with model-homogeneous and model-heterogeneous FL methods.

	Model Heterogeneity	Aggregation Scheme	Sub-model Extraction Scheme	Need of Public Data	Server Model Size	Compatibility with Secure Aggregation
FedAvg [11]	No	-	-	No	$=$ Client Model	Yes
FedProx [9]				No	$=$ Client Model	Yes
SCAFFOLD [8]				No	$=$ Client Model	Yes
FedBE [2]				Unlabeled	$=$ Client Model	No
FedGKT [5]	Yes	Knowledge Distillation	-	No	$\geq$ Largest Client Model	No
FedDF [10]				Unlabeled	$=$ Largest Client Model	No
DS-FL [7]				Unlabeled	$=$ Largest Client Model	No
Fed-ET [3]				Unlabeled	$\geq$ Largest Client Model	No
Federated Dropout [1]	Yes	Partial Training	Random	No	$\geq$ Largest Client Model	Yes
HeteroFL [4]			Static	No	$=$ Largest Client Model	Yes
FjORD [6]			Static	No	$=$ Largest Client Model	Yes
FedRolex (Our Approach)			**Rolling**	**No**	$\geq$ **Largest Client Model**	**Yes**

References

[1] Sebastian Caldas et al. "Expanding the reach of federated learning by reducing client resource requirements." In: *arXiv preprint arXiv:1812.07210* (2018).
[2] Hong-You Chen and Wei-Lun Chao. "Fedbe: Making bayesian model ensemble applicable to federated learning." In: *arXiv preprint arXiv:2009.01974* (2020).
[3] Yae Jee Cho et al. "Heterogeneous Ensemble Knowledge Transfer for Training Large Models in Federated Learning." In: *International Joint Conference on Artificial Intelligence (IJCAI)* (2022).
[4] Enmao Diao, Jie Ding, and Vahid Tarokh. "Heterofl: Computation and communication efficient federated learning for heterogeneous clients." In: *arXiv preprint arXiv:2010.01264* (2020).
[5] Chaoyang He, Murali Annavaram, and Salman Avestimehr. "Group knowledge transfer: Federated learning of large cnns at the edge." In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 14068–14080.
[6] Samuel Horvath et al. "Fjord: Fair and accurate federated learning under heterogeneous targets with ordered dropout." In: *Advances in Neural Information Processing Systems* 34 (2021).
[7] Sohei Itahara et al. "Distillation-based semi-supervised federated learning for communication-efficient collaborative training with non-iid private data." In: *arXiv preprint arXiv:2008.06180* (2020).
[8] Sai Praneeth Karimireddy et al. "Scaffold: Stochastic controlled averaging for federated learning." In: *International Conference on Machine Learning*. PMLR. 2020, pp. 5132–5143.
[9] Tian Li et al. "Federated optimization in heterogeneous networks." In: *Proceedings of Machine Learning and Systems* 2 (2020), pp. 429–450.
[10] Tao Lin et al. "Ensemble distillation for robust model fusion in federated learning." In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 2351–2363.
[11] Brendan McMahan et al. "Communication-efficient learning of deep networks from decentralized data." In: *Artificial intelligence and statistics*. PMLR. 2017, pp. 1273–1282.

Our Method: FedRolex

The key to the design of `FedRolex` is a **rolling sub-model extraction scheme**. At the server, `FedRolex` utilizes a rolling window to extract the sub-model from the global model. The rolling window advances in each round, and loops over all parts of the global model *in sequence* across different rounds. This process iterates such that the global model is evenly trained until convergence.

Taking Figure 1 as an example: in round j, the large-capacity $\{a, b, c, d\}$ and small-capacity $\{c, d, e\}$ client model are extracted from the global model. In round $j+1$, the rolling window advances 1 step and the large-capacity, and small-capacity client model becomes $\{b, c, d, e\}$ and $\{d, e, a\}$, respectively. Similarly, in round $j+2$, the rolling window advances one step further, and the models become $\{c, d, e, a\}$ and $\{e, a, b\}$.

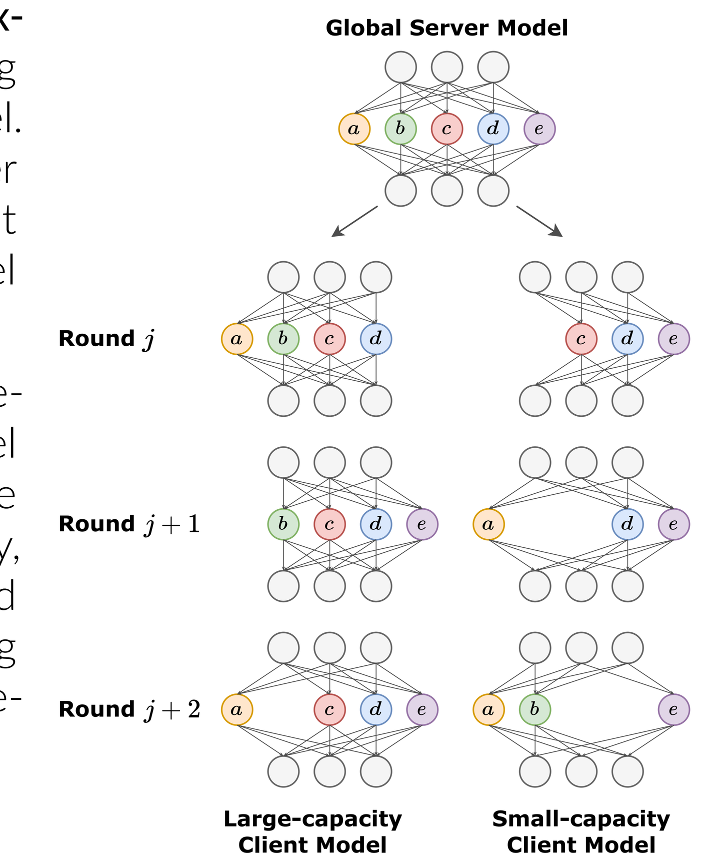

Figure 1:Overview of `FedRolex`.

Key Merits of `FedRolex`.

- Mitigates client drift induced by model heterogeneity by evenly training the global model.
- Enables training a server model that is larger than the largest client model, allowing FL to benefit from the superior performance brought by large models.
- Reduces communication costs as it only transmits the sub-model instead of the full server model to the client.
- Is fully compatible with existing secure aggregation protocols that enhance the privacy properties of FL systems.

Comparison with Random and Static Sub-model Extraction Schemes. Similar to the proposed rolling-based scheme, the sub-models extracted across different rounds by random-based scheme have different architectures. However, due to its randomness in selecting sub-models in each round, the global model is trained less evenly, making it vulnerable to *client drift*. In static sub-model extraction scheme, on the other hand, the sub-models are *always* extracted from a *designated* part of the global model. The *same* sub-model is extracted for each client in *every* round. This restricts the server model size to the largest capacity client model. More importantly, depending on their resource demands, different sub-models can *only* be trained on clients whose on-device resources are matched. As a consequence, part of the global server model cannot be trained on data at low-end client devices, causing different parts of the global model to be trained on data with different distributions.

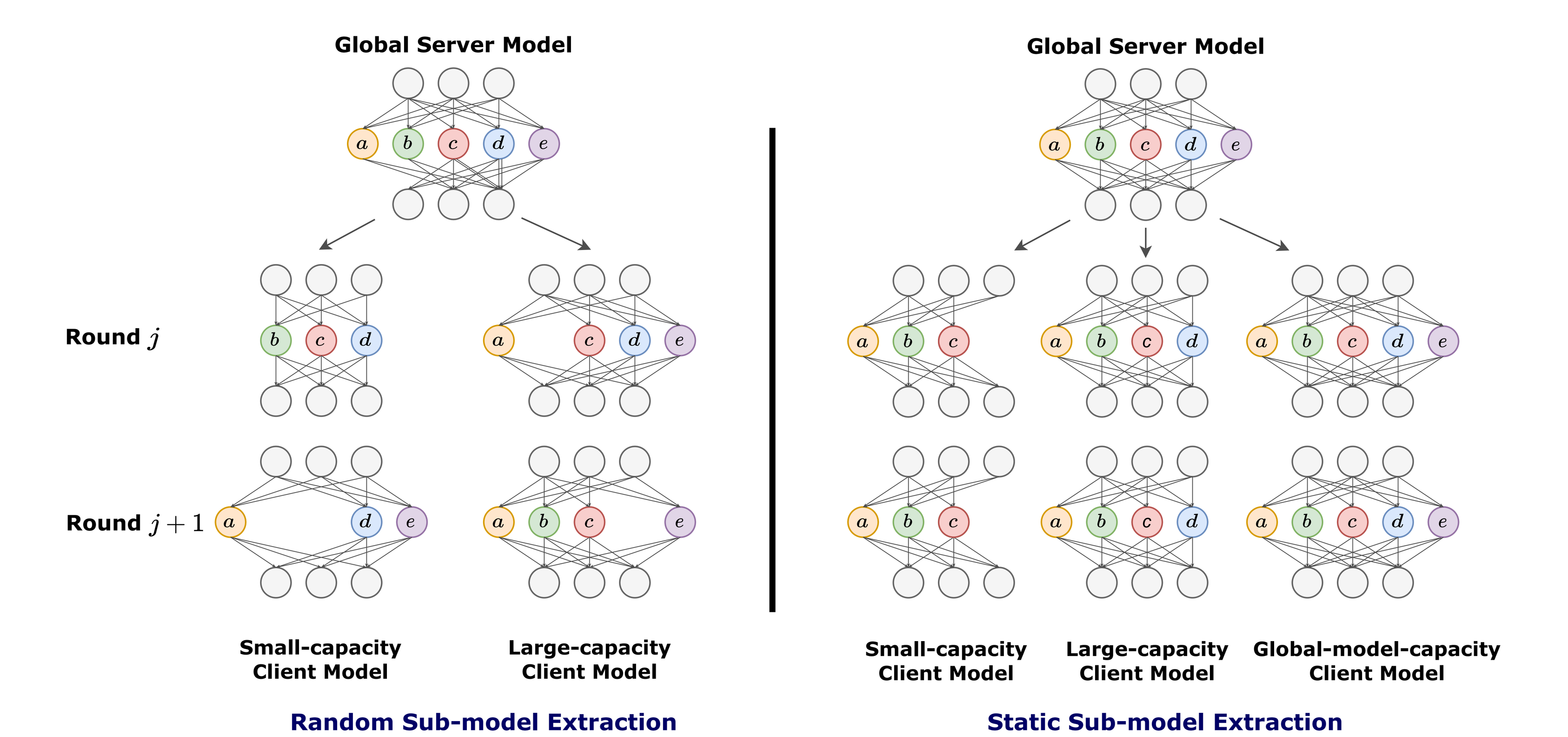

Figure 2:Illustration of random sub-model extraction scheme (Left) and static sub-model extraction scheme (Right).

Experiments

Table 2:Global model accuracy comparison between `FedRolex`, PT and KD-based model-heterogeneous FL methods, and model-homogeneous FL methods. For Stack Overflow, since KD-based methods cannot be directly used for language modeling tasks, their results are marked as N/A.

	Method	High Data Heterogeneity		Low Data Heterogeneity		Stack Overflow
		CIFAR-10	CIFAR-100	CIFAR-10	CIFAR-100	
KD-based	FedDF	73.81 (± 0.42)	31.87 (± 0.46)	76.55 (± 0.32)	37.87 (± 0.31)	N/A
	DS-FL	65.27 (± 0.53)	29.12 (± 0.51)	68.44 (± 0.47)	33.56 (± 0.55)	N/A
	Fed-ET	**78.66 (± 0.31)**	**35.78 (± 0.45)**	**81.13 (± 0.28)**	**41.58 (± 0.36)**	N/A
PT-based	HeteroFL	63.90 (± 2.74)	52.38 (± 0.80)	73.19 (± 1.71)	57.44 (± 0.42)	27.21 (± 0.22)
	Federated Dropout	46.64 (± 3.05)	45.07 (± 0.07)	76.20 (± 2.53)	46.40 (± 0.21)	23.46 (± 0.12)
	`FedRolex`	**69.44 (± 1.50)**	**56.57 (± 0.15)**	**84.45 (± 0.36)**	**58.73 (± 0.33)**	**29.22 (± 0.24)**
	Homogeneous (smallest)	38.82 (± 0.88)	12.69 (± 0.50)	46.86 (± 0.54)	19.70 (± 0.34)	27.32 (± 0.12)
	Homogeneous (largest)	**75.74 (± 0.42)**	**60.89 (± 0.60)**	**84.48 (± 0.58)**	**62.51 (± 0.20)**	**29.79 (± 0.32)**

- `FedRolex` consistently outperforms all other SOTA PT-based methods.
- In comparison with SOTA KD-based methods, `FedRolex` only performs worse than Fed-ET and FedDF on CIFAR-10 under high data heterogeneity but outperforms all the KD-based methods on the other benchmarks.

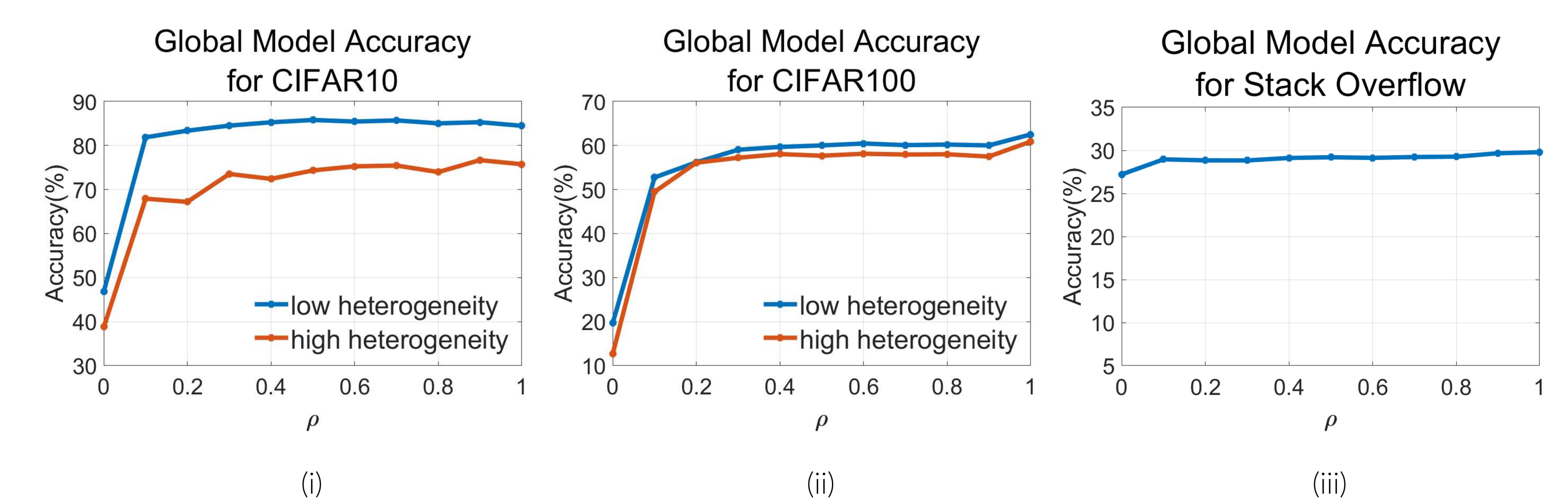

Figure 3:Impact of client model heterogeneity distribution on global model accuracy for (i) CIFAR-10, (ii) CIFAR-100, and (iii) Stack Overflow. Here a fraction, ρ of the federation use large models and the rest use small model. We can see here that having a small fraction of large-capacity models significantly boosts the global model accuracy, but further addition of large-capacity models has a limited contribution.

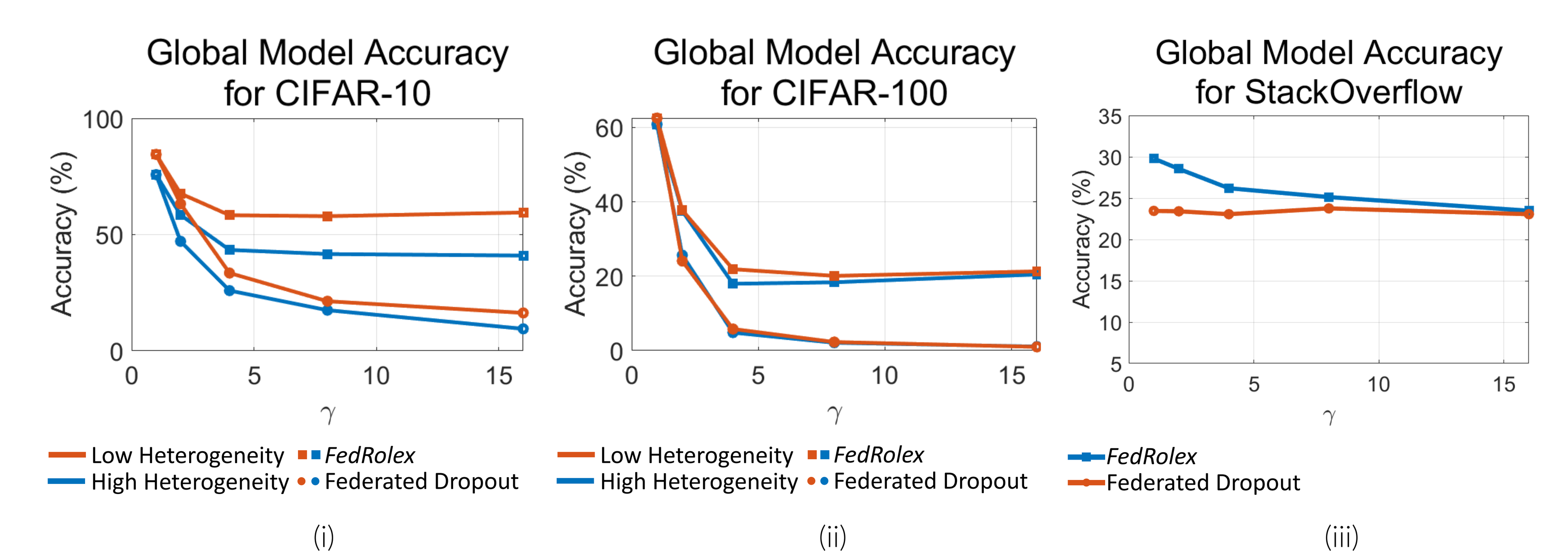

Figure 4:Performance on training larger server model when the server model is γ times the size of the client model for (i) CIFAR-10, (ii) CIFAR-100, and (iii) Stack Overflow. `FedRolex` consistently achieves higher global model accuracy than Federated Dropout across all three datasets.