Supplementary Material of Workie-Talkie: Accelerating Federated Learning by Overlapping Computing and Communications via Contrastive Regularization

Rui Chen[1], Qiyu Wan[1], Pavana Prakash[1], Lan Zhang[2], Xu Yuan[3], Yanmin Gong[4], Xin Fu[1], and Miao Pan[1]

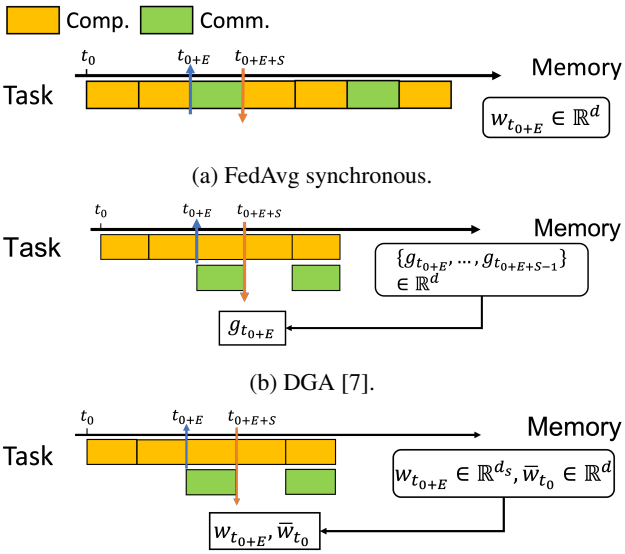
[1]University of Houston, [2]Michigan Technological University, [3]University of Delaware, [4]University of Texas at San Antonio

1. Notation

Indices	
x, y	data input (i.e., image data in image classification) and data label
n	index for edge devices ($n \in \{1, \cdots, N\}$)
r	index for rounds ($r \in \{1, \cdots, R\}$)
k	index for local training iterations ($k \in \{1, \cdots, K\}$)
c	index for class for image classification ($c \in \mathcal{Y}$)
$\mathcal{X}, \mathcal{Y}$	input space and output space
$\mathcal{Z}$	representation embedding space
$\mathcal{S}_c$	the set of indices satisfying $y = c$
Data and Weights	
$\mathcal{D}$	whole dataset
$\mathcal{D}_n(D_n)$	local dataset (the size of local dataset) of device n
$\mathbf{w}_g^r$	weight of server model on the round r
$\mathbf{w}_n^{r,k}$	weight of device n on the round r and local iteration k
$\mathbf{z}_n \in \mathcal{Z}$	embedding of device n
β	Parameter for the Dirichlet Distribution
Contrastive Regularization	
p_c	Evaluation accuracy that reflects the performance of the global model on class c.
τ	Temperature on contrastive loss
λ	Hyperparameter for the effect of weight contrastive regularization

2. Weakness of existing overlapping schemes

We describes different overlapping schemes that are recently proposed in FL frameworks in Section 2.2. The training timeline of different federated learning frameworks is shown in Fig. 1.



(a) FedAvg synchronous.

(b) DGA [7].

(c) Our proposed FedCR.

Figure 1: Comparison between different FL frameworks. The memory block represents the memory storage for local training. d represents the total number of model parameters. In (c), since our FedCR can support heterogeneous model training on-device, hence $0 < d_s \leq d$.

FedAvg (Fig. 1a) waits for the latest global parameters at t_{0+E} before the next round local computing. The overlapping schemes (Figs. 1b & 1c) allow devices to continuously perform local computing, using global model from t_0 instead of the latest model at t_{0+E}, while communicating with the FL server. Different from FedAvg, overlapping schemes have staleness issue because current round local training is not based on the latest global parameters, but the global parameters from the previous round.

Next, we provide detailed discussion and experiments on the two downside of the existing overlapping schemes, i.e., memory inefficiency and accuracy degradation.

2.1. Memory inefficiency

Memory footprint/usage is one of the key design factors of FL training over edge devices, since those devices usually have much less memory capacity compared with GPU clusters. However, the state of art overlapping techniques, DGA, is not memory-efficient for edge devices: to compensate for the stale model parameters (e.g., model differentials or model weights), each edge device has to locally store multiple copies of its old local model updates for update correction. Thus, the memory consumption increases linearly with the staleness level. For severe staleness, the memory usage for keeping these gradient copies becomes enormous. As demonstrated in Table 1, a large staleness greatly increases the memory usage for local training. Considering a edge device like Raspberry Pi 4 Model B with only 1GB or 2GB DRAM memory, it cannot support the training when the staleness $S \geq 10$, where the memory requirement for training ResNet is almost equal to its maximum memory capacity.

2.2. Accuracy degradation

We empirically examine how data heterogeneity and staleness affect their performance by training a ResNet20 on CIFAR10 dataset. The number of FL clients is set to 10. All parties engage in every round to eliminate the effect of randomness introduced by client sampling. The label ratios in each FL client's dataset follow Dirichlet distribution with a concentration parameter β, where a small value of β represents high level of data heterogeneity. The same experiment setting is used in Fig. 1 that is shown in the main paper.

Table 1: Memory usage per local training iteration (Mem) and the total number of Multiply Accumulate operations per local training (MACs) of DGA [7] under different staleness levels S. The batch size is 64.

	S=0	S=5	S=10	S=20
		CIFAR100/ResNet20		
Mem	60MB	253MB	492MB	972MB
MACs	555M	555M	555M	555M
		CIFAR10 / CNN		
Mem	16MB	20.28MB	24.56MB	33.12MB
MACs	216K	216K	216K	216K

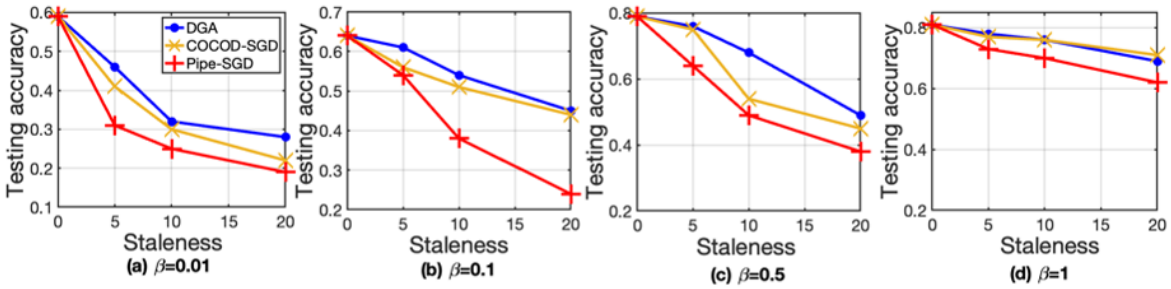


Figure 2: FL global model performance with different levels of data heterogeneity and staleness. The models are trained with 800 communication rounds and $K = 10$.

The testing accuracy of three existing overlapping approaches over varying degrees of data heterogeneity and staleness is given in Fig. 2. When staleness value equals 0, the overlapping approaches reduce to FedAvg. We can see that when data distributions of clients differ greatly (small β), the testing accuracy decreases dramatically with staleness level. Moreover, for all four non-IID settings, those overlapping designs have no advantage over FedAvg as they have lower testing accuracy than FedAvg (staleness equals to 0). Taking $\beta = 0.1$ for example, the accuracy of Pipe-SGD falls dramatically from 64% to 26% when the staleness level increases from 5 to 20, and the accuracy of DGA falls to 46% when staleness increases to 20. On the other hand, with less data heterogeneity ($\beta = 1$), the accuracy of DGA only has a small accuracy drop. This implies that the local update modification is more suitable for IID cases.

3. Regularization and Contrastive Learning

Table 2: Comparison between FedCR and peer FL designs.

	Update Rules	Data Het.	Device Het.	Het. Subnet
MOON	Interleaved	✓	×	×
FedProx	Interleaved	✓	✓	×
DGA	Overlapping	×	×	×
FedCR	Overlapping	✓	✓	✓

Adding regularizer to local on-device training can help mitigate model divergence across devices with heterogenous training data. For example, FedProx [5] introduces a proximal term into the local training objective to restrict the local model updates. The proximal term aims to minimize the ℓ_2 norm distance between the current global model and local models. SCAFFOLD [2]

corrects the local updates by introducing trainable control variables. MOON [3] is a recent regularization approach inspired by contrastive learning (CL), which creates invariance of the input-output mapping to improves generalization. The key idea of CL is to pull positive pairs together and push negative pairs apart. Thus, it helps learn the local feature representation similar to the global feature representation. Inspired by the CL [1, 6], our FedCR approach treats global model weights and local model weights as positive pair to regularize the local model training. Thus, FedCR is expected to automatically regularize the local model divergences in the presence of straggler issue and data heterogeneity, in a memory-efficient manner. Table 2 provides a comparison between FedCR and status que FL models.

4. Experiment Setup

Hyperparameter setting. We use the SGD optimizer with a learning rate of 0.01 for all approaches. The SGD weight decay and momentum are set to $5 * 10^{-4}$ and 0.9, respectively. For global aggregation, we follow the weighted averaging strategy used in FedAvg [4]. The dimension of the fixed projector in FedCR is 256 as in [1].
Subnetwork setting. The MACs and model size of subnetworks are summarized in Fig. 3.

Table 3: MACs and parameters of each subnetwork.

Methods	CNN			ResNet20			ResNet34		
	$p_c = 0.2$	$p_c = 0.6$	$p_c = 1$	$p_c = 0.2$	$p_c = 0.6$	$p_c = 1$	$p_c = 0.2$	$p_c = 0.6$	$p_c = 1$
MACs	8M	28M	60M	1.2G	3.2G	6G	94M	2.3G	4.8G
Model size	46KB	126KB	235KB	8.7MB	23.56MB	42.83MB	14.34MB	45.23MB	81.59MB

Edge device in Testbed. The FedCR is implemented on testbed, as illustrated in the main paper. On FL client side, we consider three types of edge device developer kits: NVIDIA Jetson Xavier NX, NVIDIA Jetson TX2, and NVIDIA Jetson Nano. For experiments, our testbed used twenty edge devices in total. Their profiles are summarized in Table 4.

Table 4: Edge Device Specification.

Name	CPU	GPU	RAM
Jetson Xavier NX	Carmel Arm v8.2	Volta GPU (21TOPs)	8 GB
Jetson TX2	Cortex-A57	Pascal GPU (1.33TFLOPs)	8 GB
Jetson Nano	Cortex-A57	Maxwell GPU (472 GFLOPs)	4 GB

In Tiny-ImageNet task, we consider a simulated environment where there are 100 edge devices in total and we only randomly sampled 20% devices to participate in each round of the FL training process.

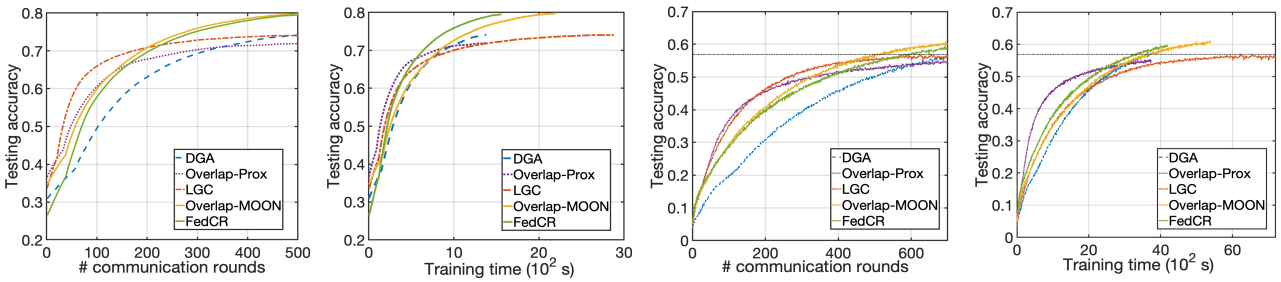
5. More experiments

5.1. Training time budgets.

Table 5: The testing accuracy of different approaches when training for the same amount of time with $\beta = 0.5$.

Methods	CIFAR10			Tiny-ImageNet		
	400s	800s	1600s	2000s	4000s	8000s
DGA	61.15	75.3	81.34	5.45	12.54	19.42
Overlap-Prox	73.2	79.03	81.97	8.23	12.62	17.33
Overlap-MOON	68.83	80.17	86.12	10.89	14.61	20.52
FedCR	79.4	84.56	87.23	15.97	18.81	24.3

We have conducted experiments with different time budgets, and shown the results in Table 5 above. We can observe that, given the fixed time budget, FedCR can consistently yield better testing accuracy than its peer approaches.



(a) Acc. vs round (CIFAR-10) (b) Acc. vs time (CIFAR-10) (c) Acc. vs round (CIFAR-100) (d) Acc. vs round (CIFAR-100)

Figure 3: The testing accuracy of different schemes under dynamic network condition.

5.2. Dynamic communication condition.

In the main paper, we have evaluated the performance under a fixed network latency scenario. We further the experimental results with dynamic network latency, as shown in Fig. 3. Here, we simulate the transmission time of different mobile devices with the mean 0.71s and standard deviation 0.24s to characterize the communication uncertainty. The other experimental settings are the same. From Fig. 3, we see that FedCR outperforms DGA, Overlap Prox, and LGC, in terms of both overall training time and model accuracy. FedCR has less training time than Overlap-MOON to reach the same target testing accuracy.

5.3. Projection Head.

Table 6: The test accuracy with/without projection head.

Methods	CIFAR-10	CIFAR-100	Tiny-ImageNet
No projection head	80.3%	61.77%	25.01%
Fixed projection head	84.2%	62.81%	25.17%
Learnable head	87.2%	66.48%	26.17%

Here we study the effect of the projection head. We compare FedCR with that without any projection head and conduct experiments on CIFAR-10, CIFAR-100, and Tiny-ImageNet. The results are shown in Table6. We can observe that the projection head help improve the performance of FedCR. The accuracy of FedCR can be improved by about 4% on average with a projection head. We also observe that the non-linear projection head can further improve the performance of FedCR. While as we show in main paper, the computation complexity and memory of non-linear projection head is much large than the fixed projection head in FedCR. Hence, we choose FedCR with fixed projection head in resource-limited edge devices.

6. Hyper-Parameters Study

Different values of λ. We show the accuracy of FedCR with different λ in Table 7. Note that when $\lambda = 0$, FedCR is Pipe-SGD and we will replace with DGA scheme. The results are shown in Table 7. The best λ for CIFAR-10, CIFAR-100, and Tiny-ImageNet are 1, and 1, 5, respectively. First, we can observe that the accuracy of FedCR with a small value of λ ($\lambda = 0.1$) is close to that of DGA (i.e., λ = 0). It is because the effect of contrastive regularization is small. When $\lambda \geq 1$, FedCR can benefit from the contrastive regularization.

Table 7: The test accuracy of FedCR with $\beta = 0.5$.

λ	CIFAR-10	CIFAR-100	Tiny-ImageNet
0 (DGA)	81.4%	58.7%	23.1%
0.1	83.5%	59.8%	24.7%
1	84.3%	62.5%	25.2%
5	84.1%	61.6%	25.5%
10	83.1%	61.1%	23.6%

Different output dimension of fixed projector. We tune the output dimension of fixed projection head from $\{128, 256, 384, 512\}$. The results are shown in Table 8. The best output dimension for CIFAR-10, CIFAR-100 and Tiny-ImageNet are 256, 256, 384. We can observe that fixed projection head with the higher dimension can improve the performance in more complex learning tasks. However, when the output dimension of fixed projector is set too large, the model converges to FedCR w/o projection head and the performance will degrade.

Table 8: The test accuracy of FedCR with $\beta = 0.5$.

Output dimension	CIFAR-10	CIFAR-100	Tiny-ImageNet
128	84.1%	62.2%	25.1%
256	84.3%	62.5%	25.1%
384	84.1%	62.3%	25.3%
512	83.7%	61.2%	24.7%

References

[1] Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian. Understanding dimensional collapse in contrastive self-supervised learning. In *Proceedings of the 10th International Conference on Learning Representations (ICLR'22)*, Baltimore MD, July 2022.

[2] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pages 5132–5143. PMLR, 2020.

[3] Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'21)*, pages 10713–10722, Virtual, June 2021. IEEE.

[4] 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 (AISTATS'17)*, Ft. Lauderdale, FL, April 2017.

[5] Anit Kumar Sahu, Tian Li, Maziar Sanjabi, Manzil Zaheer, Ameet Talwalkar, and Virginia Smith. On the convergence of federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127*, 3:3, 2018.

[6] Yuandong Tian. Deep contrastive learning is provably (almost) principal component analysis. *arXiv preprint arXiv:2201.12680*, 2022.

[7] Ligeng Zhu, Hongzhou Lin, Yao Lu, Yujun Lin, and Song Han. Delayed gradient averaging: Tolerate the communication latency for federated learning. In *Advances in Neural Information Processing Systems (NeurIPS'21)*, volume 34, New Orleans, Louisiana, December 2021.