SUPPLEMENTARY MATERIAL FOR
Federated Learning with Heterogeneous Label Noise: A Dual Structure Approach

A DETAIL OF FIGURE 1

We have tested the loss of clean samples and noisy label samples in FedAvg on the CIFAR10 dataset with symmetric label noise, where data heterogeneity is increasing from #class=10 (IID) to #class=2 (extermely non-IID), as shown Figure 1 of Section 1. In this section, we further compare the the performance of FedAvg with the centralized training on the CIFAR10 dataset with noise rate 0.8.

In the Figure 6 (a), we show the prediction accuracy of the model trained in IID distribution and non-IID distribution (# class=2) by FedAvg, and the prediction accuracy of the model trained by centralized. On the one hand, the performance of the model trained by FedAvg on the dataset with IID distribution is similar with the model trained by centralized. On the other hand, the performance of the model trained by FedAvg is broken and worse than random guess on the dataset with high noise rate and non-IID distribution (# class=2).

Figure 6 displays the loss of clean samples and noisy samples in *centralized training*, *FedAvg with IID distribution*, and *FedAvg with non-IID distribution (# class=2)* on dataset with noise rate 0.8. At the same noise rate, both *centralized training*, and *FedAvg with IID distribution* show the memorization effect but *FedAvg with non-IID distribution (# class=2)*. It shows that non-IID distribution will will break the memory effect in extreme cases (for example, non-IID distribution (# class=2), noise rate = 0.8).

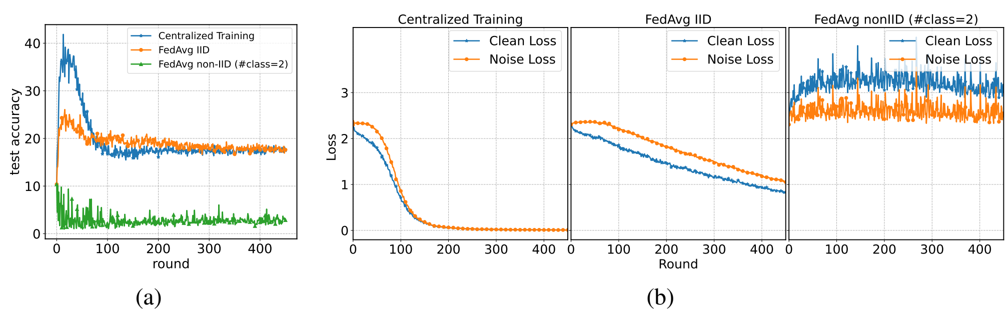


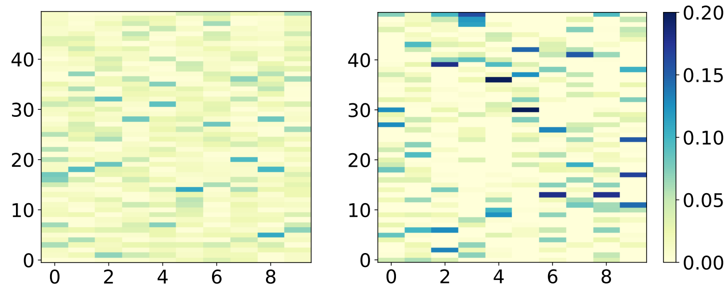
Figure 6: The performance of FedAvg and the centralized training on the CIFAR10 dataset with noise rate 0.8.

B DATA PARTITION MECHANISM

Given a dataset D with a sample size of N and L class labels. The number of samples corresponding to each class of labels are $\{N^1, N^2, ..., N^L\}$. Denote by h the data partition mechanism, where we partition data to IID distribution by shuffling into K clients and each receiving $\lfloor N/K \rfloor$ examples, and we partition data to non-IID distribution by a distribution matrix $M_{K\times L}$, where M_{ij} is the proportion of the samples labeled as j on the i-th client to the j-th class samples, and satisfies the Eqn. (2).

$$\begin{cases} 0 \leq M_{ij} \leq 1, i = 1, 2, .., K, j = 1, 2, ..., L \\ \sum_{i=1}^{K} M_{ij} \leq 1, j = 1, ..., L. \end{cases} \tag{2}$$

Let u_i be the number of categories in ith client, $\boldsymbol{p}_i$ be a vector of length L that sums to u_i. For each $0 < i < K$, $\boldsymbol{p}_i$ can be generated by by sampling randomly without replacement in range (0, L-1). Let $\boldsymbol{q}_j$ be a vector of length K which represents the proportion of j class samples on every client. For each $0 < j < L$, $\boldsymbol{q}_j$ can be generated by sampling from the Dirichlet distribution, where $\boldsymbol{q}_j(i) = M_{ij}$. For example, we visualize the distribution matrix of generated IID distribution and non-IID distribution in Figure 7.



(a) IID distribution. (b) non-IID distribution (#class=5).

Figure 7: Heat-map for client data distribution with class label. In Figure (b), non-IID distribution (#class=5) is the non-IID distribution of clients with 5 class labels.

C IMPLEMENTATION DETAILS

Dataset	CIFAR100	CIFAR10	MNIST
Feature size	32 * 32	32 * 32	28 * 28
Training instances	60,000	50,000	60,000
classes	100	10	10
clients	50	100	100
Rounds	450	450	300
Model Architecture	ResNet34	ResNet-18	LetNet-5

Table 3: Summary of the datasets.

Implementation details of CORES For PCORES, FedCORES, and FedTwinCORES on CIFAR10 and CIFAR100, we first train network on the dataset for 2 warm-up rounds with only CE (Cross Entropy) loss and the data selection is performed at the 3 round. Then β is linearly increased from 0 to 1 for 10 local epochs in every client and kept as 1 for the rest of the local epochs. For PKNN, FedKNNpretrain, FedKNN, and FedTwinKNN on MNIST, we first train network on the dataset for 10 warm-up rounds with only CE (Cross Entropy) loss and the data selection is performed at the 11 round. Then β is linearly increased from 0 to 0.1 for 50 local epochs in every client and kept as 0.1 for the rest of the local epochs.

Implementation details of KNN-based method For PKNN, FedKNNpretrain, FedKNN and FedTwinKNN on CIFAR10 and CIFAR100, we set $k = 10$. For PKNN, FedKNNpretrain, FedKNN and FedTwinKNN on CIFAR10 and CIFAR100, we set $k = 5$.

D COMPARISON WITH STATE-OF-THE-ART METHODS

The accuracies of various methods on MNIST and CIFAR100 with homogeneous and heterogeneous label noise at different noise levels. Both two label noise generation model (ANDC and DCAN) are displayed.

Table 4: The accuracies of various methods on CIFAR100 with homogeneous label noise and heterogeneous label noise at different noise levels. Both two label noise generation model (ANDC and DCAN) are tested.

	Method	IID					non-IID				
			Symmetric		Pairflip			Symmetric		Pairflip	
		0.0	0.2	0.5	0.2	0.4	0.0	0.2	0.5	0.2	0.4
ANDC	FedAvg	67.60	48.06	25.98	52.99	37.76	66.64	47.62	23.91	52.09	36.09
	FedCorAvg	29.01	23.72	17.80	24.84	18.66	28.08	23.11	17.07	23.56	18.43
	FedProx	65.78	49.37	27.76	53.63	38.72	63.09	46.91	25.95	51.88	36.34
	FedPCORES	42.30	31.16	17.35	33.27	25.83	40.17	29.94	16.16	32.89	26.68
	FedPKNN	66.88	52.48	27.14	56.19	39.49	64.66	51.51	28.09	53.15	36.94
	FedCORES	70.83	59.92	30.72	56.65	42.15	66.89	52.69	24.40	51.45	37.38
	FedTwinCORES	69.26	55.23	27.81	58.47	42.9	66.00	52.19	24.40	54.49	39.55
	FedKNN	68.69	59.68	29.87	60.78	44.02	66.75	58.50	31.73	57.83	40.37
	FedTwinKNN	68.60	54.32	26.54	57.18	39.66	66.49	53.68	27.59	54.44	37.43
	FedTwinKNNpretrain	67.36	49.97	25.46	52.73	36.91	66.49	47.75	24.18	50.97	37.19
DCAN	FedAvg	67.60	52.74	26.66	52.56	35.90	66.64	47.00	22.57	47.69	37.56
	FedCorAvg	29.01	24.47	16.97	24.38	16.41	28.08	24.12	17.25	23.70	18.92
	FedProx	65.78	52.06	29.93	52.86	37.17	63.09	48.51	24.20	50.51	39.00
	FedPCORES	42.30	32.35	17.16	32.82	25.51	40.17	29.52	14.77	31.37	24.40
	FedPKNN	66.88	56.05	30.59	54.57	37.83	64.66	52.31	26.16	52.23	36.89
	FedCORES	70.83	57.88	29.71	56.96	41.16	66.89	54.43	20.48	53.16	35.66
	FedTwinCORES	69.26	55.06	31.27	54.99	40.57	66.00	49.88	21.95	53.73	40.08
	FedKNN	68.69	61.23	36.73	60.56	41.75	66.75	61.27	31.71	58.71	44.25
	FedTwinKNN	68.60	57.93	29.94	55.49	39.44	66.49	56.03	25.94	55.15	39.94
	FedTwinKNNpretrain	67.36	53.22	29.15	52.13	37.54	66.49	47.85	22.46	50.79	38.84

Table 5: The accuracies of various methods on MNIST with homogeneous label noise and heterogeneous label noise at different noise levels. Both two label noise generation model (ANDC and DCAN) are tested.

	Method	IID					non-IID				
			Symmetric		Pairflip			Symmetric		Pairflip	
		0.0	0.2	0.5	0.2	0.4	0.0	0.2	0.5	0.2	0.4
ANDC	FedAvg	99.03	96.32	80.68	91.64	66.51	98.74	87.82	66.04	83.72	65.93
	FedCorAvg	97.47	76.25	93.48	96.66	84.41	80.95	77.78	83.31	82.16	61.23
	FedProx	98.71	98.25	97.24	97.67	85.21	98.29	96.97	91.64	90.29	71.39
	FedPCORES	62.02	57.42	50.82	52.05	45.34	61.76	53.91	42.92	49.85	42.13
	FedPKNN	98.11	97.79	92.79	97.69	80.71	98.10	95.6	84.24	89.75	68.95
	FedCORES	99.0	97.38	70.08	91.8	68.56	98.97	89.96	70.34	81.58	66.59
	FedTwinCORES	99.07	96.55	50.95	92.12	69.57	99.01	89.29	64.24	81.4	66.86
	FedKNN	98.21	98.08	93.57	97.78	80.27	98.16	95.4	84.63	89.55	69.97
	FedTwinKNN	98.09	97.92	93.14	97.68	78.99	98.09	95.93	82.48	91.52	68.71
	FedKNNpretrain	98.99	96.26	81.24	91.82	66.54	98.71	88.92	67.03	82.37	64.56
DCAN	FedAvg	99.04	96.03	79.97	93.41	67.94	98.48	91.72	75.93	96.39	91.33
	FedCorAvg	97.47	96.63	93.82	96.49	85.17	82.58	81.24	37.85	92.68	56.87
	FedProx	98.63	98.17	97.38	97.71	88.44	98.03	92.38	85.03	96.10	92.14
	FedPCORES	61.85	57.53	50.15	55.62	46.79	62.98	56.58	46.89	57.68	55.17
	FedPKNN	98.23	98.01	93.28	97.71	77.83	98.07	94.78	84.34	95.62	88.12
	FedCORES	99.03	97.87	79.15	93.2	66.93	98.74	95.93	66.84	96.29	93.37
	FedTwinCORES	98.96	96.65	64.89	93.9	67.11	98.84	93.49	68.81	96.35	92.47
	FedKNN	98.19	98.2	93.51	97.89	81.82	98.07	97.72	86.57	97.52	92.29
	FedTwinKNN	98.08	98.06	93.5	98.02	82.5	98.12	97.79	88.85	97.49	90.44
	FedKNNpretrain	99.02	95.99	80.92	93.03	66.6	98.70	95.18	88.13	96.48	88.81