A. Implementation Details

Computing Platform. In this paper, we use PyTorch to implement all the algorithms and experiments in both the main paper and this supplementary material. We run our experiments using an Nvidia RTX3090 GPU with 24 GB of memory.

Model Architecture. In Section 5, we adopt a convolutional neural network (ConvNet) that follows the same architecture as reported in [37]. The encoder of this model consists of three convolutional layers, each followed by a ReLU activation function and average pooling. To facilitate comparison with FedBN, batch normalization is incorporated into the model. A fully-connected layer serves as the classifier and is attached on top of the encoder.

Training Details. In addition to the configurations and hyperparameters detailed in Sections 5.1 and 5.2, we have set specific values for other algorithms. For FedProx and MOON, the hyperparameter μ is set to 0.001 and 1.0, respectively. In FedDyn, we use 0.01 for the hyperparameter α. For experiments on the DomainNet dataset with FedDM and FedAF, we employ an image-per-class (IPC) of 20, to achieve a similar condensation ratio as label-skew scenarios. Additionally, we resize the images of DomainNet into 64×64 resolution. For FedAF, the regularization weights $(\lambda_{\text{loc}}, \lambda_{\text{glob}})$ for collaborative data condensation and local-global knowledge matching are set to (0.0001,0.01), (0.0001,0.01), and (0.001, 2.0) on CIFAR10, CIFAR100, and FMNIST, respectively. On DomainNet, we use (0.01, 0.1) for $(\lambda_{\text{loc}}, \lambda_{\text{glob}})$. Furthermore, for all algorithms, we employ PyTorch's built-in SGD optimizer with a momentum of 0.9. The same implementation is also used for the optimizer of local data condensation in FedDM and FedAF. The accuracy evaluation for all algorithms is conducted over the testing split of each benchmark dataset, to simulate centralized validation or test data at the server. For all experiments, we average the accuracy and learning curves over three trial runs, each with a different random seed.

Visualizing Data Distribution. For label-skew data heterogeneity, we explore three degrees of non-IID in cross-client data distribution, represented by α values of 0.02, 0.05, and 0.1, where a smaller α indicating stronger heterogeneity. Figures 6, 7, and 8 show the class-wise data distribution per client for CIFAR10, CIFAR100, and FMNIST datasets, respectively, using a random seed from our experiments. In these figures, the size of the blue circles corresponds to the number of data samples. We observe that with a smaller α, clients tend to possess data concentrated in fewer classes and share fewer common classes, indicating a more pronounced label-skew non-IID distribution. For feature-skew heterogeneity, we analyze feature distribution in Figure 9. Here, domain features are extracted using the encoder of a ResNet50 [12] pre-trained on ImageNet-1K [6]. The features are then fitted into a 2D space using t-SNE [28] for visualization. Figure 9 clearly shows that, despite identical classes across six domains, the feature distribution of each clas varies significantly from one domain to another.

Methods	α=0.02	α=0.05	α=0.1
FedAvg	26.48±0.58	32.72±2.47	35.85±3.73
FedProx	26.86±2.69	32.73±2.45	36.25±2.96
FedBN	27.00±2.49	30.29±3.38	35.48±3.45
MOON	29.59±3.57	33.11±3.74	37.26±2.66
FedDyn	22.67±1.54	29.89±4.48	35.38±1.56
FedGen	26.63±2.07	32.48±3.04	38.85±2.00
FedDM	39.18±0.29	39.47±0.66	40.83±0.67
FedAF	**41.10±0.50**	**41.40±0.66**	**42.93±0.29**

Table 6. Comparison of accuracy achieved by various state-of-the-art baselines by training ResNet18 on CIFAR10. Three different degrees of label-skew data heterogeneity are implemented.

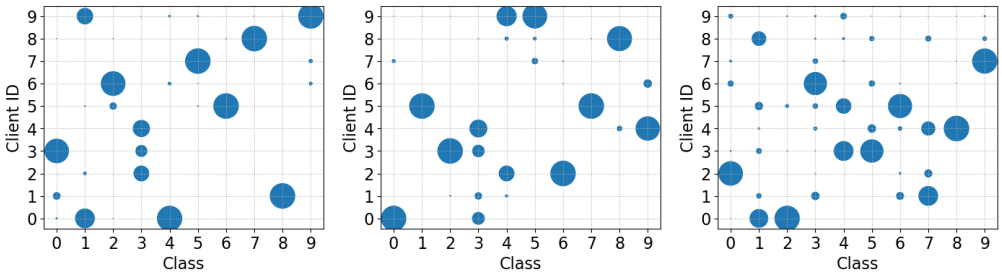
B. More Experiment Results with ResNet18

In further experimentation, we evaluate the performance of FedAF and compare it to baseline methods using a ResNet18 model [12] on CIFAR10 dataset. Conducting 20 communication rounds for all algorithms, the resulting accuracy are presented in Table 6. As expected, FedAF consistently outperforms the baseline methods in both accuracy and related standard deviation across three different degrees of data heterogeneity. This advantage is particularly pronounced under strong heterogeneity, such as at α=0.02, where FedAF achieves a 14.62% higher accuracy than FedAvg and an 11.51% improvement over MOON, the top performer of *aggregate-then-adapt* baselines. Additionally, FedAF maintains steady accuracy advantages of 2% over FedDM throughout all the α settings.

C. Communication Cost Analysis

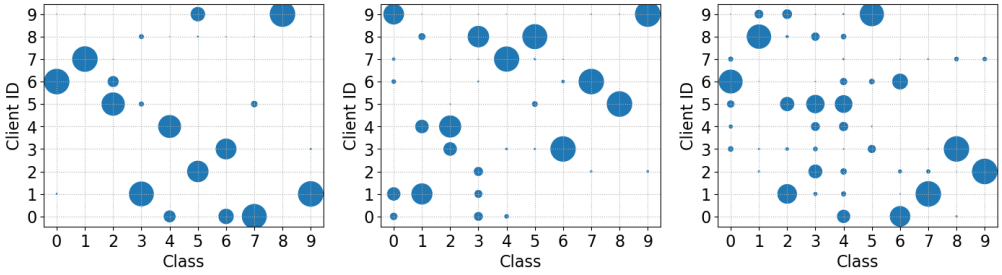
Baseline Methods and Model Size. For typical *aggregate-then-adapt* baseline methods like FedAvg, only the model parameters are communicated back and forth between clients and the server. In our experiments, the ConvNet and ResNet18 model has 381,450 and 11,181,642 parameters, respectively. With float32 precision, each parameter takes 4 bytes, so that the size of these two models is evaluated at 1.46 MB, and 42.65 MB, respectively.

FedAF and Size of Condensed Data. In FedAF's upstream communication, each client k sends three items to the server: 1) the local condensed data $\mathcal{S}_k$, 2) the class-wise mean logit $\mathcal{V}_k$, and 3) the class-wise mean soft labels $\mathcal{R}_k$, whereas in the downstream communication, each client k downloads two items: 1) the global model $\mathbf{w}$ which shares the same architecture as that in *aggregate-then-adapt* baselines, and 2) the class-wise mean logits from all other



(a) CIFAR10 α=0.02 (b) CIFAR10 α=0.05 (c) CIFAR10 α=0.1

Figure 6. Visualization of cross-client data distribution for CIFAR10 dataset under three different degrees of label-skew heterogeneity.



(a) FMNIST α=0.02 (b) FMNIST α=0.05 (c) FMNIST α=0.1

Figure 7. Visualization of cross-client data distribution for FMNIST dataset under three different degrees of label-skew heterogeneity.

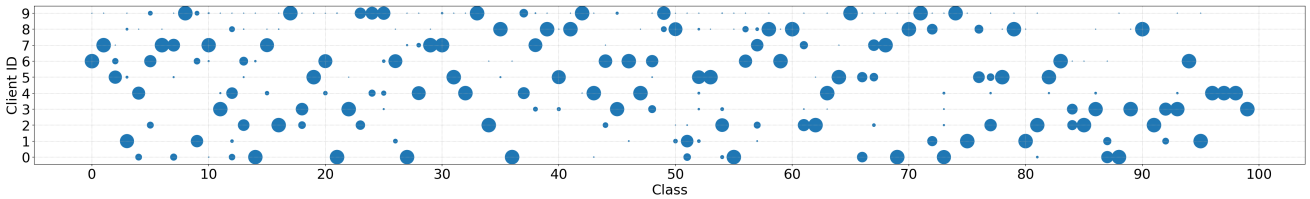
clients, denoted by $\mathcal{V}$ in (9). The matrices $\mathcal{V}_k$ and $\mathcal{R}_k$ share the same size, for ten-class datasets like CIFAR10, FMNIST, and the sub-dataset we extracted from the DomainNet, both $\mathcal{V}_k$ and $\mathcal{R}_k$ include ten vectors with ten values in float32, making the size of them is approximately 4×10^{-4} MB each. Whereas for CIFAR100 that contains data of 100 classes, the size of $\mathcal{V}_k$ and $\mathcal{R}_k$ altogether is then evaluated at approximately 0.076 MB. Assuming that the condensed data is stored and transmitted in the PIL format so that one can use 8-bit unsigned integer (or 1 byte) for each pixel per channel, the size of every ten such condensed data samples from FMNIST is evaluated at 7.5×10^{-3} MB. Similarly, the size of every ten condensed data learned from CIFAR10 or CIFAR100 is about 0.03 MB.

Comparison with FedAvg. With the above calculation as a base, we compare the per-round upstream communication cost incurred by FedAF and that of typical *aggregate-then-adapt* method such as FedAvg in Table 7, where FedAF using an image-per-class (IPC) of 50. As described earlier, we use three random seeds to generate the three sets of data distribution and report the average communication overhead. Note that the communication cost of transmitting $\mathcal{V}_k$, $\mathcal{V}$, and $\mathcal{R}_k$ is negligible compared to transmitting the condensed data and the model, so the downstream communication cost is essentially the same as that incurred by downloading the global model from the server, which is the same for FedAvg and FedAF. From Table 7, one can observe that for training the ResNet18 model, FedAF is much more efficient in communication cost compared to FedAvg. When learning the

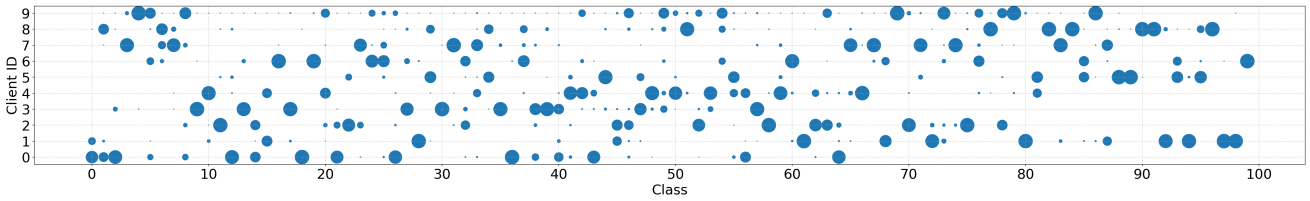
Dataset	α	CNN		ResNet18	
		FedAvg	FedAF	FedAvg	FedAF
FMNIST	0.02	1.46 MB	0.06 MB	42.65 MB	0.06 MB
	0.05		0.09 MB		0.09 MB
	0.1		0.14 MB		0.14 MB
CIFAR10	0.02	1.46 MB	0.22 MB	42.65 MB	0.22 MB
	0.05		0.31 MB		0.31 MB
	0.1		0.44 MB		0.44 MB
CIFAR100	0.02	1.46 MB	1.93 MB	42.65 MB	1.93 MB
	0.05		2.46 MB		2.46 MB
	0.1		3.22 MB		3.22 MB

Table 7. Per-round upstream communication cost incurred by FedAvg and FedAF for learning CNN and ResNet18 on FMNIST, CIFAR10, and CIFAF100. FedAF uses an IPC of 50.

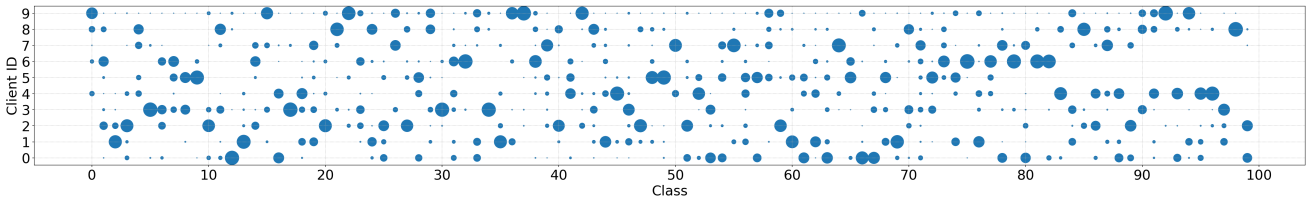
ConvNet model, which is relative smaller in size, FedAF still achieves significantly higher communication efficiency than FedAvg, especially on FMNIST and CIFAR10. While FedAvg incurs slightly less communication than FedAF for learning the ConvNet model on CIFAR100, FedAF drastically outperforms FedAvg in accuracy and convergence (see performance comparison in Section 5.1). Moreover, unlike FedAvg, where the communication cost is solely determined by the model size and thus becomes increasingly expensive when a larger model is being learned, FedAF's communication cost is irrespective of the size of the underlying model. More interestingly, FedAF incurs less communication overhead in stronger label-skew data heterogeneity scenarios. These merits mark the extraordinary cost-effectiveness of FedAF.



(a) CIFAR100 α=0.02



(b) CIFAR100 α=0.05



(c) CIFAR100 α=0.1

Figure 8. Visualization of cross-client data distribution for CIFAR100 dataset under three different degrees of label-skew heterogeneity.

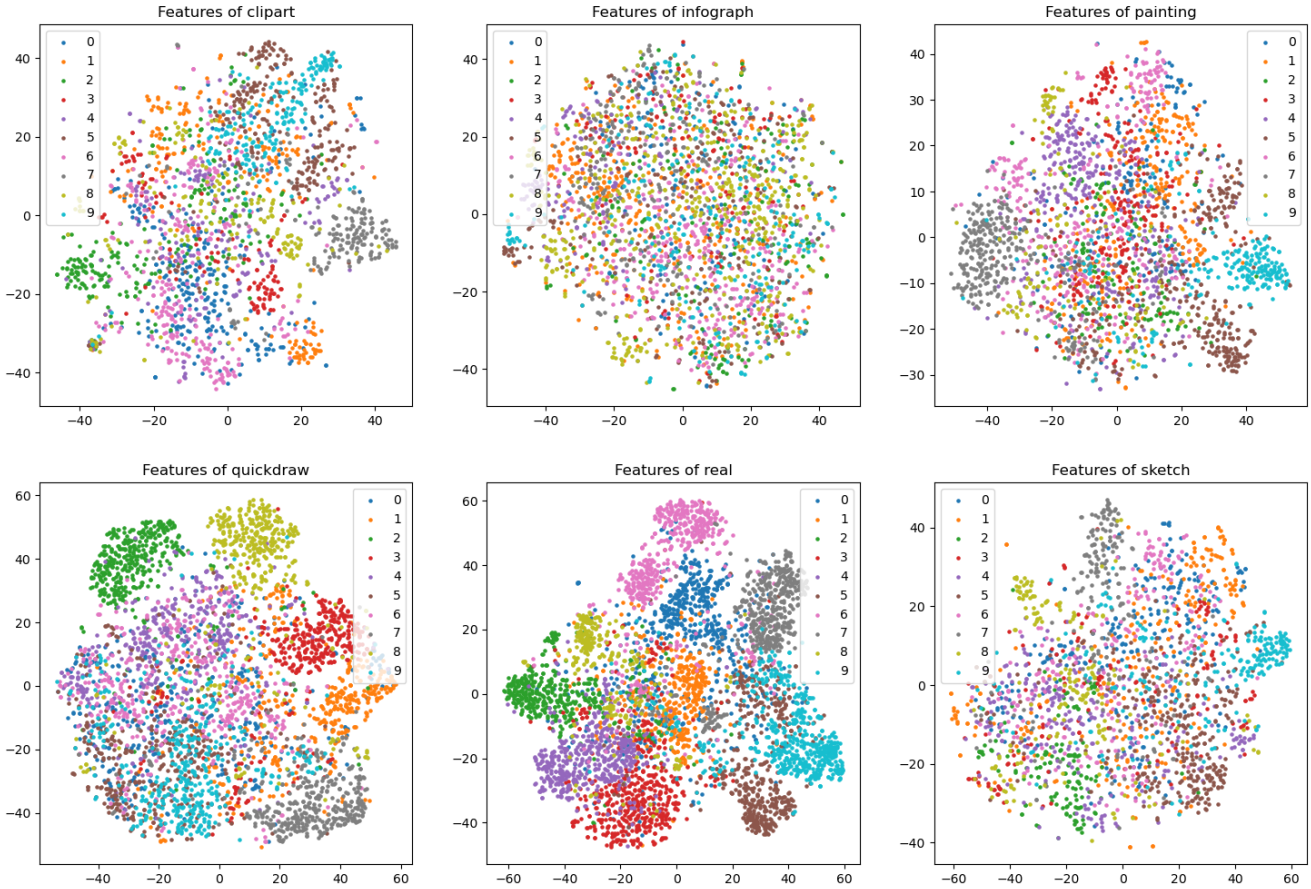


Figure 9. T-SNE visualization of features extracted from DomainNet data, using a sub-dataset split from [23].