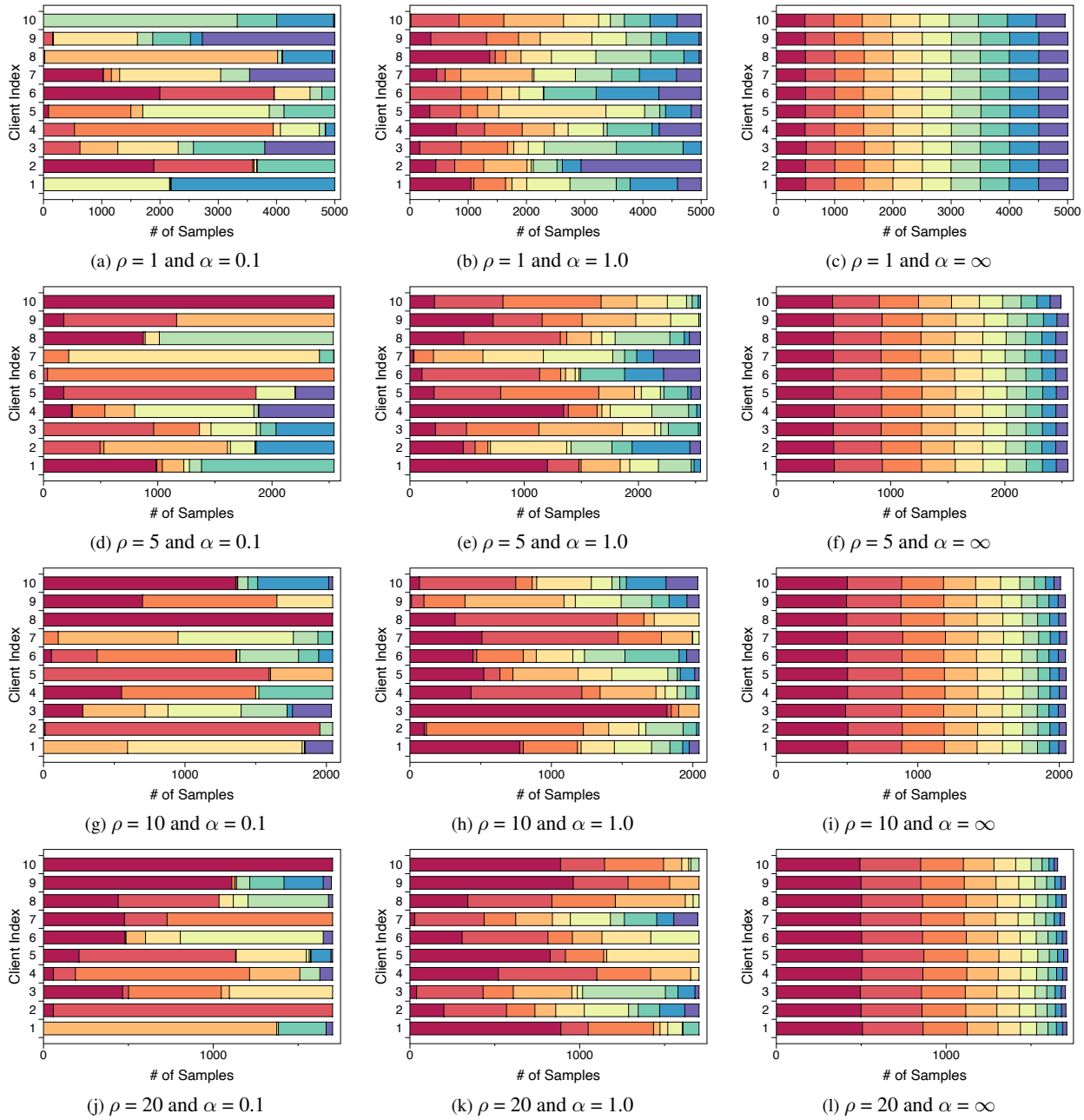
A. Detailed Local Data Distribution

We adopt a Latent Dirichlet Allocation (LDA) strategy for Non-IID setting [27, 42], where each client k is assigned the partition of classes by sampling $\mathbf{p}_k \sim Dir(\alpha \cdot \mathbb{1})$, where $\mathbb{1} \in \mathbb{R}^C$. α is a concentration parameter that controls the local heterogeneity level. The smaller α, the more heterogeneous data distribution. Since we consider a fairness issue in the FAL framework, the total number of samples should be equally partitioned for all clients. Therefore, we made a doubly stochastic matrix $P = [\tilde{\mathbf{p}}_1, \dots, \tilde{\mathbf{p}}_K]^\top$ by scaling $\mathbf{p}_k$ to $\tilde{\mathbf{p}}_k$, when the number of client and class are same (i.e., P is a square matrix). Note that we set the sum of columns and rows to the proper values for a non-square matrix. We visualized the examples of CIFAR-10 when the clients $K = 10$ in Figure 6.



(a) $\rho = 1$ and $\alpha = 0.1$ (b) $\rho = 1$ and $\alpha = 1.0$ (c) $\rho = 1$ and $\alpha = \infty$

(d) $\rho = 5$ and $\alpha = 0.1$ (e) $\rho = 5$ and $\alpha = 1.0$ (f) $\rho = 5$ and $\alpha = \infty$

(g) $\rho = 10$ and $\alpha = 0.1$ (h) $\rho = 10$ and $\alpha = 1.0$ (i) $\rho = 10$ and $\alpha = \infty$

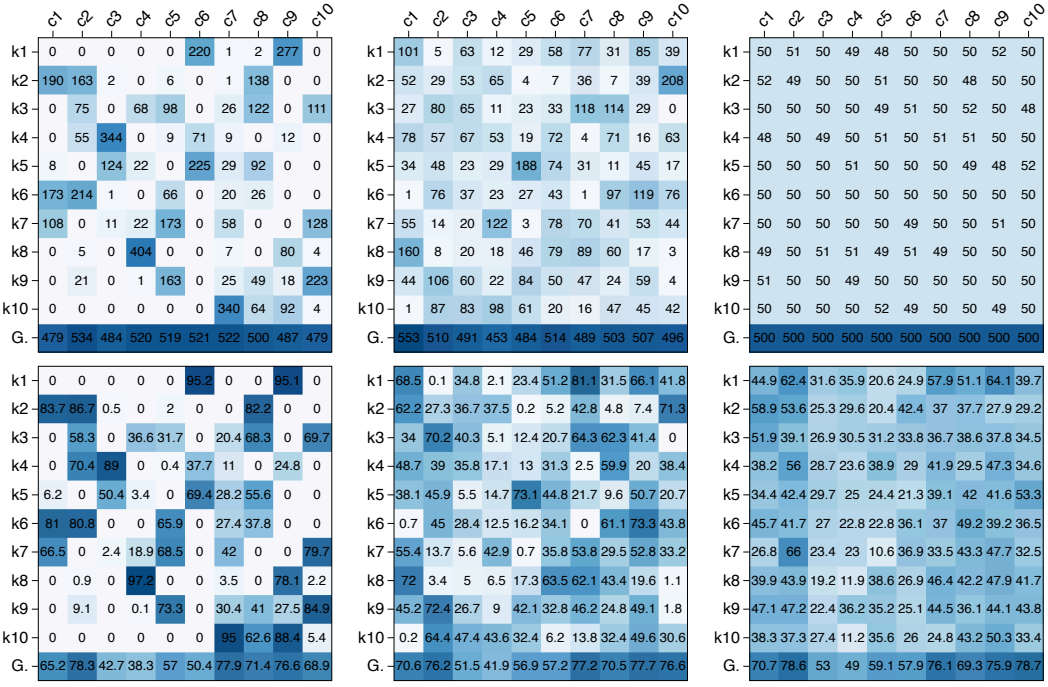
(j) $\rho = 20$ and $\alpha = 0.1$ (k) $\rho = 20$ and $\alpha = 1.0$ (l) $\rho = 20$ and $\alpha = \infty$

Figure 6. Visualization of the client data distribution on CIFAR-10. Each color represents a different class. The higher ρ denotes the more global imbalanced distribution. The higher α denotes the more locally balanced data.

B. Detailed Analysis Results

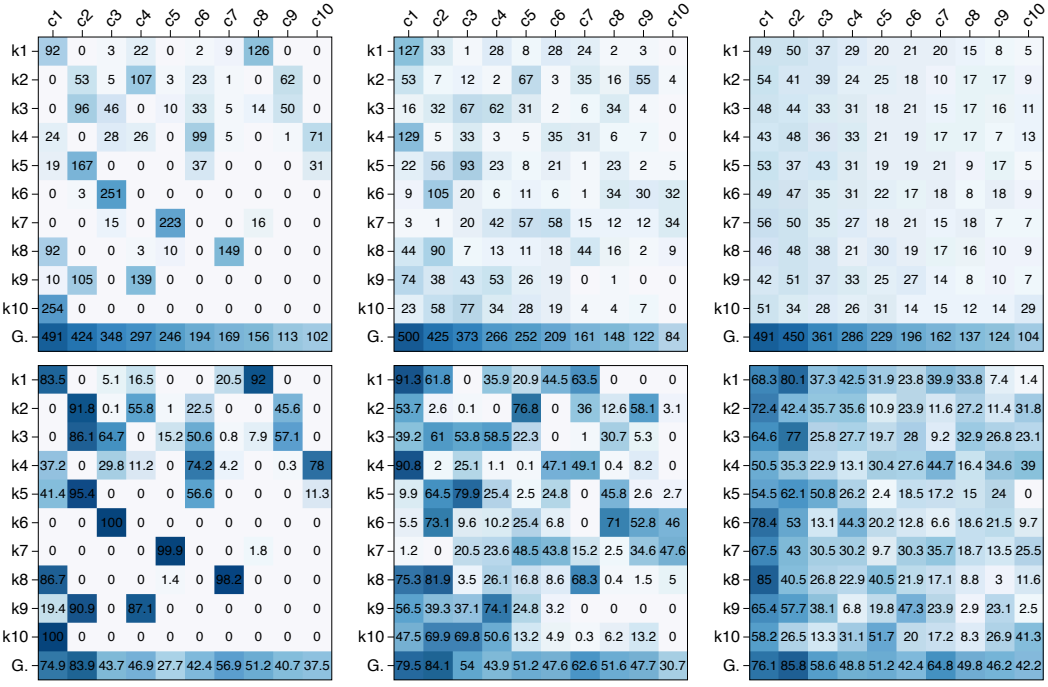
B.1. Detailed Matrices for Data Counts and Accuracy

We summarized the detailed matrices for the combinations of $\rho = \{1, 5, 10, 20\}$ and $\alpha = \{0.1, 1.0, \infty\}$.



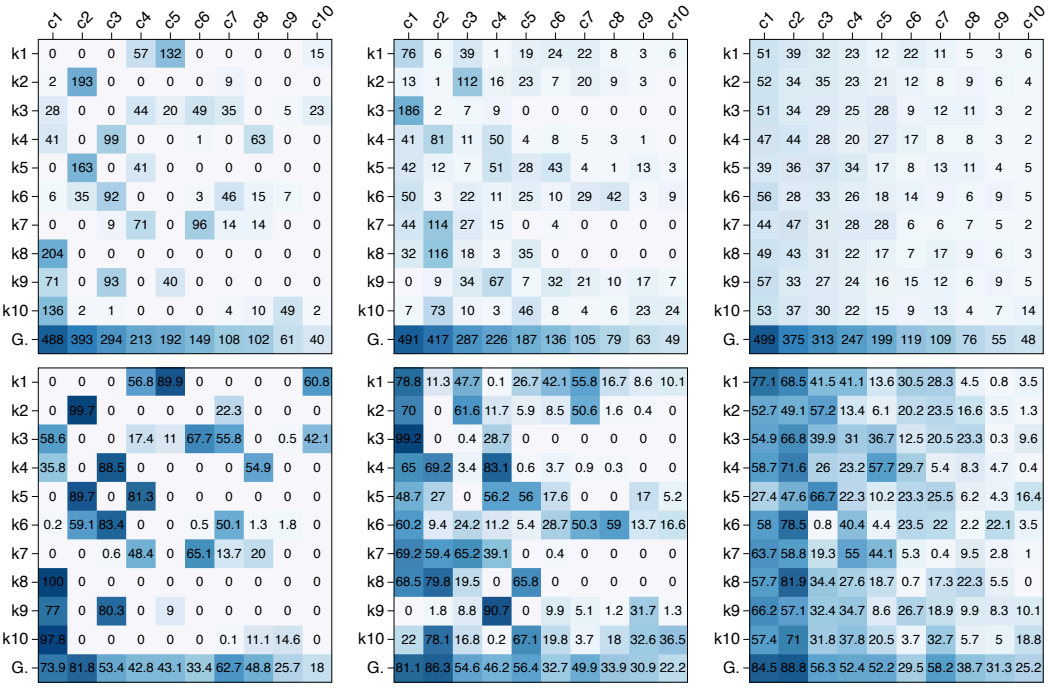
(a) $\rho = 1$ and $\alpha = 0.1$ (b) $\alpha = 1.0$ (c) $\alpha = \infty$

Figure 7. Matrices of data count (top) and class-wise accuracy (down) when $\rho = 1$.



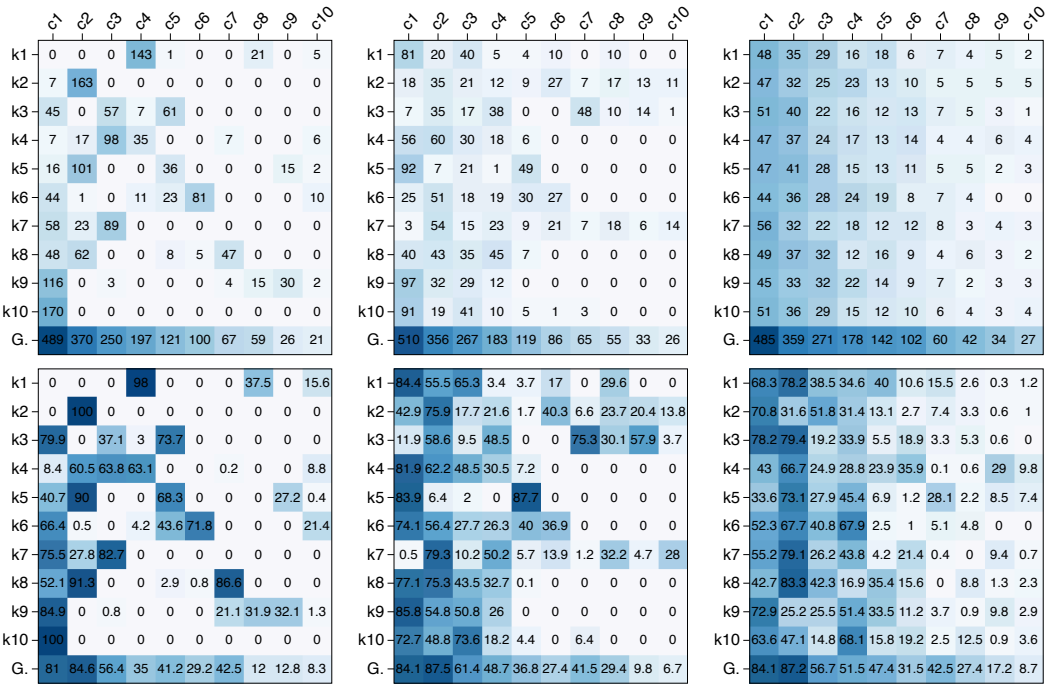
(a) $\alpha = 0.1$ (b) $\alpha = 1.0$ (c) $\alpha = \infty$

Figure 8. Matrices of data count (top) and class-wise accuracy (down) when $\rho = 5$.



(a) $\alpha = 0.1$ (b) $\alpha = 1.0$ (c) $\alpha = \infty$

Figure 9. Matrices of data count (top) and class-wise accuracy (down) when $\rho = 10$.



(a) $\alpha = 0.1$ (b) $\alpha = 1.0$ (c) $\alpha = \infty$

Figure 10. Matrices of data count (top) and class-wise accuracy (down) when $\rho = 20$.

B.2. Detailed Earth Mover Distance

Table 5 summarizes the detailed local and global EMD for the combinations of $\rho = \{1, 5, 10, 20\}$ and $\alpha = \{0.1, 1.0, \infty\}$.

ρ	α	model	Local EMD (↓)					Global EMD (↓)				
			10%	20%	30%	40%	50%	10%	20%	30%	40%	50%
1	0.1	G	0.632	0.638	0.641	0.643	0.646	0.019	0.064	0.086	0.095	0.091
		L	0.632	0.597	0.592	0.595	0.601	0.019	0.050	0.050	0.046	0.055
1	1.0	G	0.297	0.297	0.300	0.300	0.300	0.017	0.066	0.079	0.084	0.083
		L	0.297	0.248	0.232	0.235	0.241	0.017	0.053	0.065	0.068	0.074
1	∞	G	0.049	0.077	0.070	0.065	0.061	0.014	0.070	0.066	0.063	0.060
		L	0.049	0.042	0.054	0.059	0.066	0.014	0.025	0.044	0.053	0.062
5	0.1	G	0.662	0.663	0.666	0.666	0.669	0.211	0.201	0.196	0.194	0.195
		L	0.662	0.628	0.627	0.628	0.634	0.211	0.232	0.232	0.236	0.228
5	1.0	G	0.402	0.391	0.387	0.388	0.389	0.206	0.188	0.180	0.173	0.169
		L	0.402	0.309	0.306	0.306	0.341	0.206	0.200	0.201	0.196	0.196
5	∞	G	0.213	0.190	0.178	0.168	0.165	0.206	0.185	0.174	0.162	0.163
		L	0.213	0.179	0.176	0.180	0.180	0.206	0.176	0.173	0.178	0.180
10	0.1	G	0.692	0.685	0.687	0.685	0.685	0.280	0.268	0.267	0.265	0.267
		L	0.692	0.652	0.650	0.654	0.660	0.280	0.270	0.277	0.282	0.281
10	1.0	G	0.491	0.463	0.459	0.456	0.455	0.297	0.263	0.247	0.244	0.242
		L	0.491	0.408	0.402	0.405	0.415	0.297	0.256	0.257	0.255	0.255
10	∞	G	0.315	0.240	0.229	0.223	0.222	0.303	0.237	0.226	0.222	0.221
		L	0.315	0.238	0.237	0.239	0.240	0.303	0.237	0.234	0.237	0.239
20	0.1	G	0.692	0.680	0.676	0.674	0.677	0.377	0.300	0.294	0.294	0.298
		L	0.692	0.641	0.633	0.636	0.644	0.377	0.304	0.326	0.321	0.323
20	1.0	G	0.481	0.455	0.450	0.448	0.448	0.374	0.311	0.300	0.295	0.292
		L	0.481	0.448	0.437	0.431	0.437	0.374	0.354	0.342	0.303	0.304
20	∞	G	0.371	0.298	0.284	0.274	0.276	0.368	0.294	0.282	0.271	0.272
		L	0.371	0.313	0.293	0.290	0.289	0.368	0.309	0.287	0.288	0.289

Table 5. Local and global EMD on CIFAR-10 for 12 combinations of $\rho = \{1, 5, 10, 20\}$ and $\alpha = \{0.1, 1.0, \infty\}$.

C. Pseudo Algorithm of LoGo

Algorithm 1 is the overall pipeline of the FAL framework. Specifically, we summarize the detailed pseudocode of our LoGo algorithm.

Algorithm 1 FAL framework with LoGo algorithm

Input: initialized parameter Θ; unlabeled data U^1; sampling strategy $\mathcal{A}$; labeling budget B; clients number K; AL round R;
Output: trained parameter Θ^{R*}

 # Alternating AL and FL Procedure

for $k = 1, \ldots, K$ **do**
  Randomly sample $L_k^1 = \{x_1, \ldots, x_B\}$ from U_k^1, and $U_k^2 = U_k^1 \setminus L_k^1$
  Get the labeled set D_k^1 from the oracles
end for
$\Theta^{1*} = \texttt{FedAvg}(\Theta, D^1, K)$

for $r = 2, \ldots, R$ **do**
  for $k = 1, \ldots, K$ **do**
    $D_k^r, U_k^{r+1} = \texttt{LoGo}(\Theta^{(r-1)*}, D_k^{r-1}, U_k^r)$
  end for
  $\Theta^{r*} = \texttt{FedAvg}(\Theta, D^r, K)$
end for

Function `LoGo`:

 # Macro Step
Train a local-only model $\Theta_{k*}^{(r-1)}$ from the scratch only using D_k^{r-1}
For each $x \in U_k^r$, calculate the gradient embedding $g_{\hat{y}}^x$ by Eq. (7)
Cluster U_k^r into B clusters($\mathcal{C}_1, ..., \mathcal{C}_B$) by Eq. (8)

 # Micro Step
$L_k^r = \emptyset$
for $\mathcal{C}_i = \mathcal{C}_1, \ldots, \mathcal{C}_B$ **do**
  $L_k^r = L_k^r \cup \{\mathcal{A}(\mathcal{C}_i, \Theta^{(r-1)*}, 1)\}$
  $D_k^r = D_k^{r-1} \cup D_k^r$ and $U_k^{r+1} = U_k^r \setminus L_k^r$
end for
return D_k^r, U_k^{r+1}

Function `FedAvg`:

for $FL\ round$ **do**
  Distribute Θ to the all client
  for $k = 1, \ldots, K$ **do**
    Train Θ_k on D_k^r by minimizing $\mathbb{E}_{D_k^r}[\ell(x, y; \Theta_k)]$
  end for
  $\Theta = (\sum_k \Theta_k)/K$
end for
return Θ

D. Experimental Settings

D.1. Datasets

We mainly experimented on two natural image datasets (CIFAR-10[2], SVHN[3]) and three medical image datasets[4] (PathMNIST, DermaMNIST, OrganAMNIST). Table 6 provides a summary of the five datasets. For the details of partitioning data to each client, please refer to Appendix A.

	Dataset	# of Train	# of Test	# of Classes	ρ
Natural	CIFAR-10	50,000	10,000	10	1.0
	SVHN	73,257	26,032	10	2.97
Medical	PathMNIST	89,996	7,180	9	1.63
	DermaMNIST	7,007	2,005	7	58.66
	OrganAMNIST	34,581	17,778	11	4.54

Table 6. Summary of benchmark datasets.

D.2. Implementation Details

For the FL training pipeline, we set the number of FL rounds to 100 and local update epochs to 5. We used a SGD optimizer with the initial learning rate of 0.01 and the momentum of 0.9. The learning rate was decayed by 0.1 at half and three-quarters of federated learning rounds to ensure convergence, and we used a random horizontal flipping as data augmentation. For training local-only models, we trained the model using the aforementioned settings for 50 epochs. However, the training was terminated if the training accuracy reached 99%. It should be noted that we averaged the classification accuracy of the last 5 epochs in each round and repeated all experiments with four different seeds. All algorithms were implemented using PyTorch 1.11.0 and executed using NVIDIA RTX 3080 GPUs.

D.3. Experimental Categories

A total of six categories were considered in the evaluation:

1. 'Query selector' of whether to use a local-only or global model with the six compared strategies.
2. 'Heterogeneity level' of varying degree of class imbalance. We adopt a Latent Dirichlet Allocation (LDA) [27] strategy. For example, the smaller α, the more heterogeneous the data distribution.
3. 'Imbalance ratio' of used datasets. We classified five datasets for evaluation based on the imbalance ratio ρ. CIFAR-10 and PathMNIST belong to a low imbalance ratio ($\rho < 2$), and SVHN, DermaMNIST, and OrganAMNIST belong to a high imbalance ratio ($\rho \geq 2$).
4. 'Model architecture.' We employed four layers of convolution neural network for a base architecture and also experimented with ResNet-18 [18] and MobileNet [19].
5. 'Budget size' for labeling. We tested small (1%), medium (5%), and large (20%) budget sizes for each round.
6. 'Model initialization' of either learning from scratch (random) or from the checkpoint of the previous AL round (continue).

[2]https://www.cs.toronto.edu/ kriz/cifar.html
[3]http://ufldl.stanford.edu/housenumbers
[4]https://medmnist.com/

D.4. Combination of experimental settings

We compared our algorithms and baselines in 38 comprehensive experimental settings, which are the combinations of the aforementioned six categories. All the experimental combinations we performed are summarized in Table 7.

Query Selelctor	Dir(α)	Data Type	Model Arch.	Budget Size	Model Init.
Global	0.1	CIFAR-10	4CNN	5%	Random
Global	0.1	SVHN	4CNN	5%	Random
Global	0.1	PathMNIST	4CNN	5%	Random
Global	0.1	OrganAMNIST	4CNN	5%	Random
Global	0.1	DermaMNIST	4CNN	5%	Random
Global	1	CIFAR-10	4CNN	5%	Random
Global	1	SVHN	4CNN	5%	Random
Global	∞	CIFAR-10	4CNN	5%	Random
Global	∞	SVHN	4CNN	5%	Random
Global	0.1	CIFAR-10	4CNN	5%	Continue
Global	0.1	SVHN	4CNN	5%	Continue
Global	0.1	CIFAR-10	ResNet-18	5%	Random
Global	0.1	SVHN	ResNet-18	5%	Random
Global	0.1	CIFAR-10	MobileNet	5%	Random
Global	0.1	SVHN	MobileNet	5%	Random
Global	0.1	CIFAR-10	4CNN	1%	Random
Global	0.1	SVHN	4CNN	1%	Random
Global	0.1	CIFAR-10	4CNN	20%	Random
Global	0.1	SVHN	4CNN	20%	Random
Local-only	0.1	CIFAR-10	4CNN	5%	Random
Local-only	0.1	SVHN	4CNN	5%	Random
Local-only	0.1	PathMNIST	4CNN	5%	Random
Local-only	0.1	OrganAMNIST	4CNN	5%	Random
Local-only	0.1	DermaMNIST	4CNN	5%	Random
Local-only	1	CIFAR-10	4CNN	5%	Random
Local-only	1	SVHN	4CNN	5%	Random
Local-only	∞	CIFAR-10	4CNN	5%	Random
Local-only	∞	SVHN	4CNN	5%	Random
Local-only	0.1	CIFAR-10	4CNN	5%	Continue
Local-only	0.1	SVHN	4CNN	5%	Continue
Local-only	0.1	CIFAR-10	ResNet-18	5%	Random
Local-only	0.1	SVHN	ResNet-18	5%	Random
Local-only	0.1	CIFAR-10	MobileNet	5%	Random
Local-only	0.1	SVHN	MobileNet	5%	Random
Local-only	0.1	CIFAR-10	4CNN	1%	Random
Local-only	0.1	SVHN	4CNN	1%	Random
Local-only	0.1	CIFAR-10	4CNN	20%	Random
Local-only	0.1	SVHN	4CNN	20%	Random

Table 7. Summary of the entire experimental combinations.

E. Computational Cost of Query Selection

In Table 8, we measured the wallclock time for various combinations of the algorithm, query selector, and labeling ratio. We confirmed that as the percentage of labeled data increases, the time required to measure the importance score with the global model decreases due to the reduced amount of unlabeled data. Conversly, the local-only model takes more time as it requires training on a larger number of labeled samples. Our LoGo algorithm shows a comparable computational cost to the baselines that use the local-only model (L) for query selection. Note that we used a simple Entropy sampling within LoGo algorithm to measure the uncertainty, and the only possible bottleneck is k-means clustering in the Macro step.

Query ratio	Entropy		Coreset		BADGE		GCNAL		ALFA-Mix		LoGo
	G	L	G	L	G	L	G	L	G	L	G, L
5% → 10%	5.99	8.85	7.32	10.24	14.43	17.36	8.20	11.13	13.88	20.87	17.10
40% → 45%	4.17	33.59	7.02	33.99	10.01	39.11	8.11	35.46	11.94	41.99	37.42
75% → 80%	3.95	59.57	6.72	58.98	3.95	62.62	7.71	60.26	10.46	65.16	56.81

Table 8. Computational cost on CIFAR-10 with 4 layers of CNN. We averaged the query selection time (sec.) of all 10 clients, measured on a RTX 3090 GPU.

F. LoGo with Various FL Methods

We have further experimented with two federated learning algorithms, FedProx [28] and SCAFFOLD [23], in conjunction with AL strategies. Specifically, we compared our LoGo with baselines that demonstrated Top-1 or Top-2 performance more than once in Table 3. The experimental configurations are same to those used in Table 3. As summarized in Table 9, LoGo consistently outperforms the baselines for both federated learning algorithms. This observation suggests that LoGo is an orthogonal selection algorithm that can be integrated with any federated learning algorithm, having potential to improve the performance in various applications.

FL algo.	Method	Model	CIFAR-10			SVHN		
			20%	40%	60%	20%	30%	40%
FedProx	Entropy	G	62.89	67.52	70.38	82.22	84.34	85.42
		L	<u>65.72</u>	<u>70.57</u>	<u>72.42</u>	82.08	83.73	85.30
	BADGE	G	64.16	68.62	70.82	<u>83.09</u>	**84.65**	85.84
		L	65.54	70.56	72.30	81.99	84.17	85.17
	ALFA-Mix	G	63.77	68.34	70.78	82.63	84.48	<u>85.94</u>
		L	63.44	67.83	70.31	80.71	82.81	84.22
	LoGo	G, L	**65.79**	**70.61**	**72.61**	**83.12**	<u>84.61</u>	**86.09**
SCAFFOLD	Entropy	G	65.58	70.37	72.52	82.75	85.69	86.48
		L	67.96	<u>72.67</u>	<u>74.06</u>	83.24	84.30	85.82
	BADGE	G	66.33	70.68	72.79	83.80	84.72	**86.93**
		L	<u>68.27</u>	72.52	73.79	83.40	84.61	86.16
	ALFA-Mix	G	66.11	70.50	72.55	<u>84.11</u>	**85.72**	86.14
		L	66.11	70.00	71.91	82.15	82.89	84.74
	LoGo	G, L	**68.33**	**72.77**	**74.48**	**84.29**	<u>85.70</u>	<u>86.73</u>

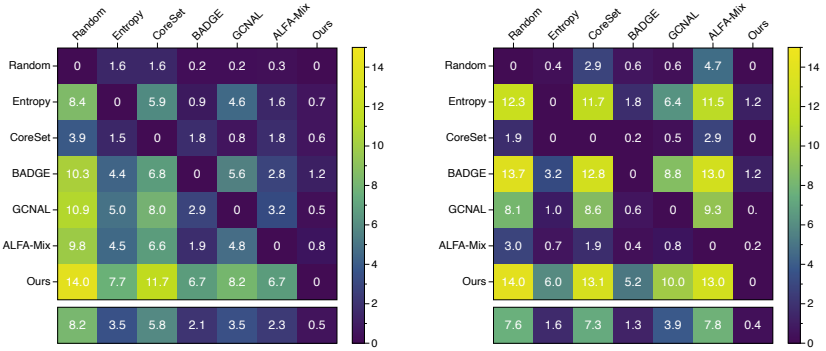
Table 9. Classification accuracy on two benchmarks with FedProx ($\mu = 0.01$) and SCAFFOLD. We compared to three overwhelming baselines and averaged three random seeds. **Bold** and <u>underline</u> mean Top-1 and Top-2, respectively.

G. Detailed Experimental Results

In this Section, G.1 summarizes all the comparison matrices results based on six categories: query selector, heterogeneity level, imbalance ratio, model architecture, budget size, and model initialization in Figure 4. Figure 11–16 are breakdowns of the matrix in Figure 5 into six categories. G.2 provides comprehensive line plots for 38 experimental settings. It can be seen that LoGo overwhelms the baselines in most cases at both each category and detailed experimental setting level.

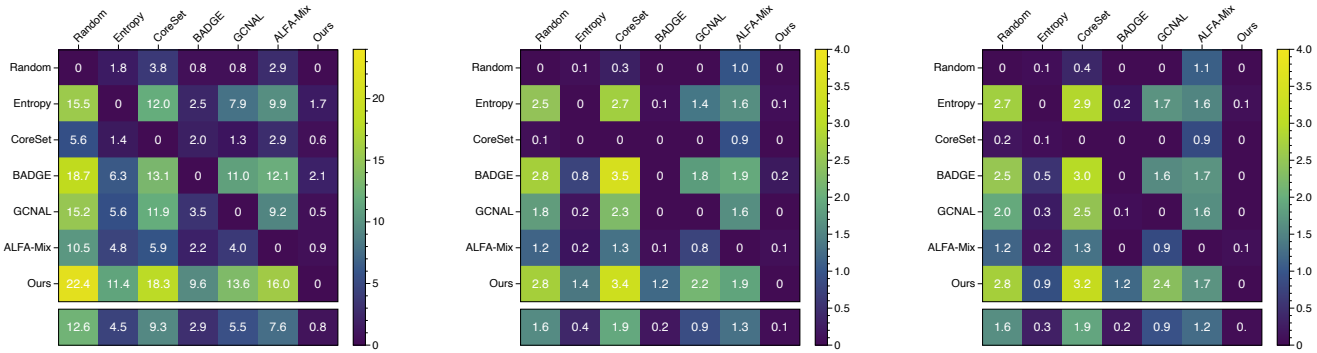
G.1. Detailed Penalty Comparision Matrix

A maximum value of each matrix corresponds to Table 7, and the bar plots in Figure 4 are calculated from these matrices.



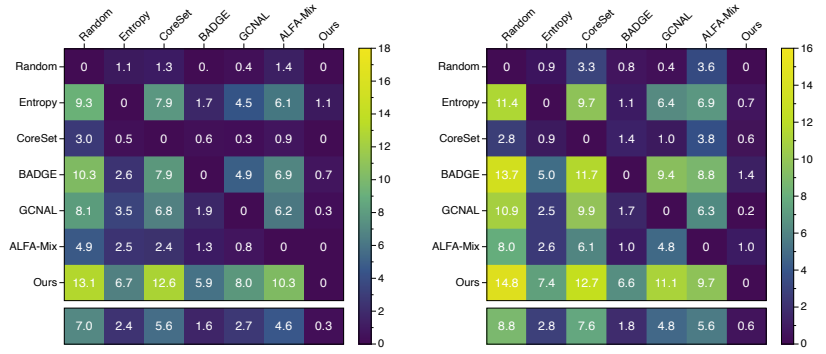
(a) Global (b) Local-only

Figure 11. Pairwise penalty matrix for a query selector category. The maximum value of both matrices is 19.



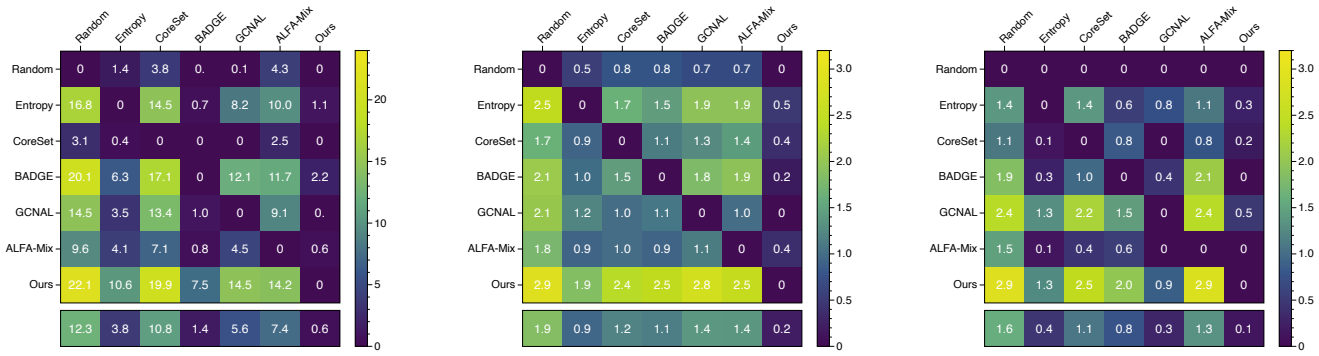
(a) $\alpha = 0.1$ (b) $\alpha = 1.0$ (c) $\alpha = \infty$

Figure 12. Pairwise penalty matrix for a heterogeneity level category. The maximum value of three matrices is 30, 4, and 4.



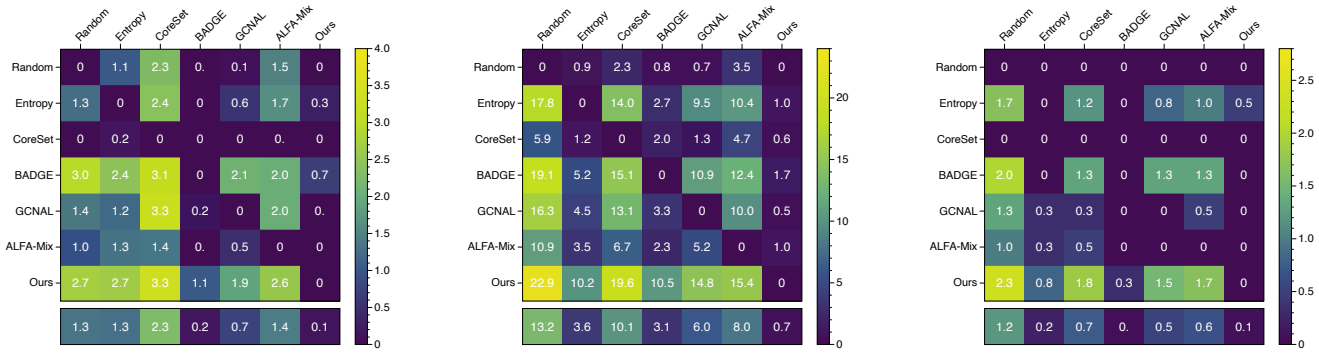
(a) $\rho < 2$ (b) $\rho \geq 2$

Figure 13. Pairwise penalty matrix for imbalance ratio category. The maximum value of two matrices is 18 and 20, respectively.



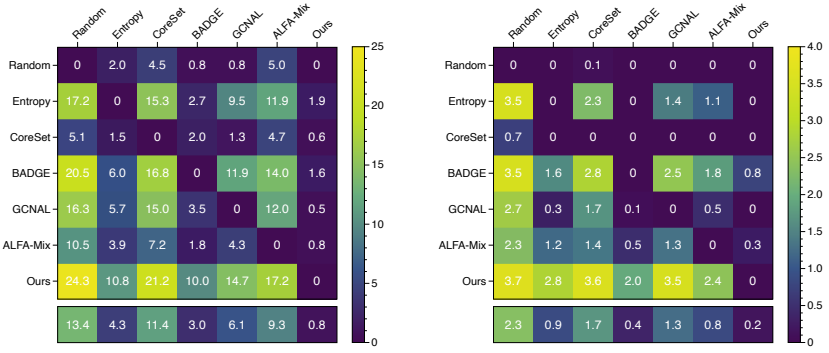
(a) Four Convolutional Neural Network (b) ResNet-18 (c) MobileNet

Figure 14. Pairwise penalty matrix for a model architecture category. The maximum value of three matrices is 30, 4, and 4.



(a) Budget 1% (b) Budget 5% (c) Budget 20%

Figure 15. Pairwise penalty matrix for a budget size category. The maximum value of three matrices is 4, 30, and 4, respectively.

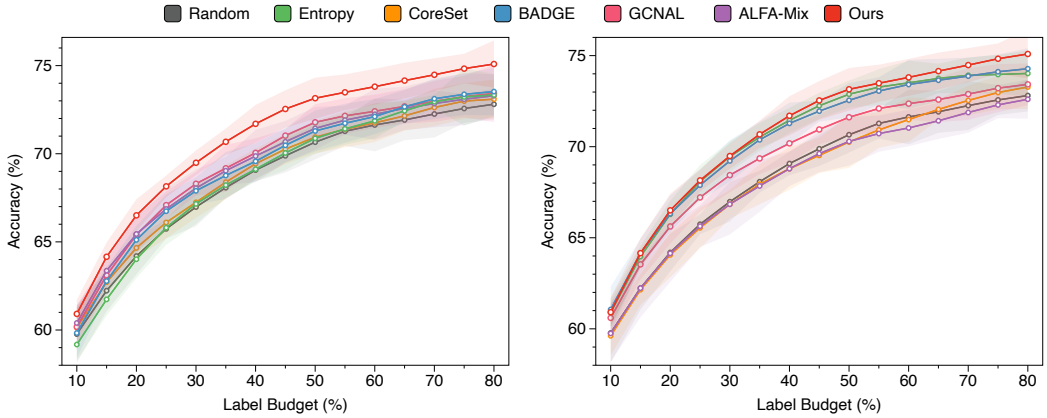


(a) Random initialization (b) Continue initialization

Figure 16. Pairwise penalty matrix for a model initialization category. The maximum value of two matrices is 34 and 4.

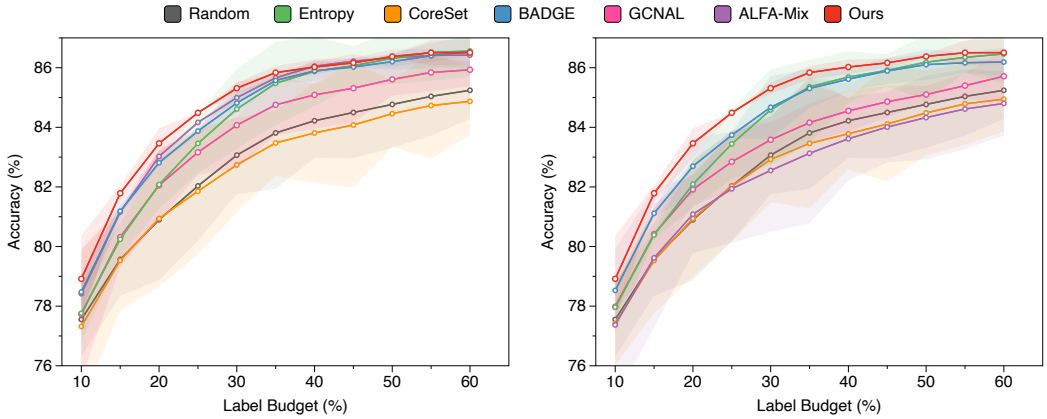
G.2. Detailed Performance Comparision

For the line plots, we note that 'Random' and 'Ours' are independent of the query selector type.



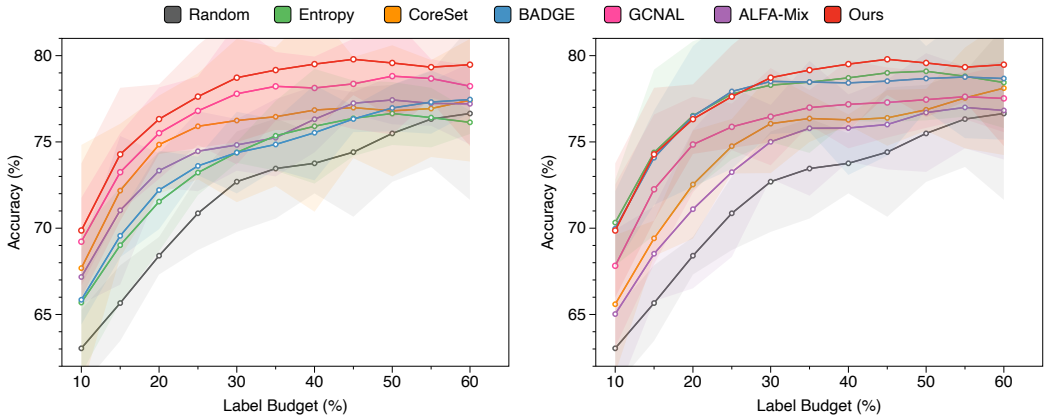
(a) Global Model (b) Local-only Model

Figure 17. Test accuracy on CIFAR-10, four layers of CNN, $\alpha = 0.1$, medium budget size (5%), and random initialization.



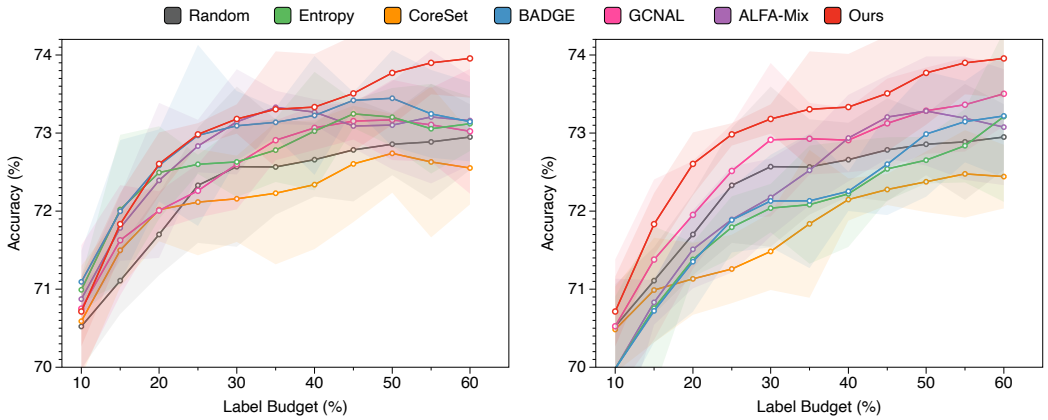
(a) Global Model (b) Local-only Model

Figure 18. Test accuracy on SVHN, four layers of CNN, $\alpha = 0.1$, medium budget size (5%), and random initialization.



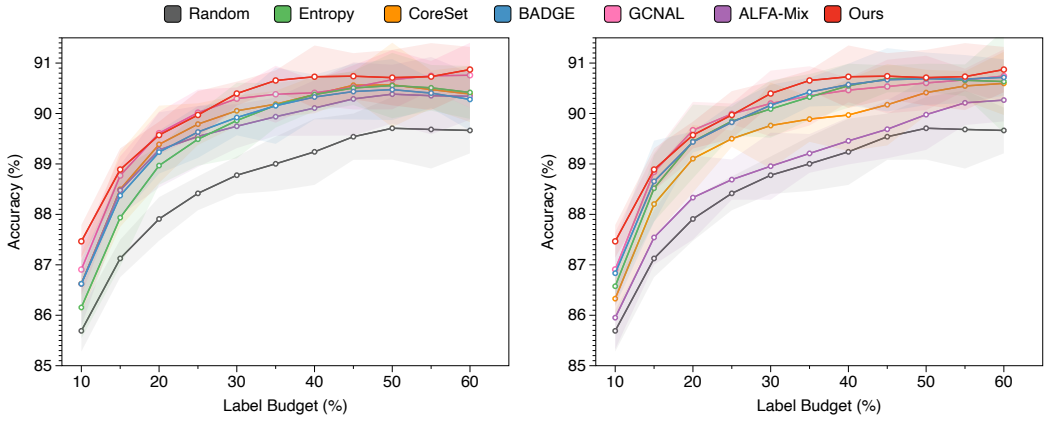
(a) Global Model (b) Local-only Model

Figure 19. Test accuracy on PathMNIST, four layers of CNN, $\alpha = 0.1$, medium budget size (5%), and random initialization.



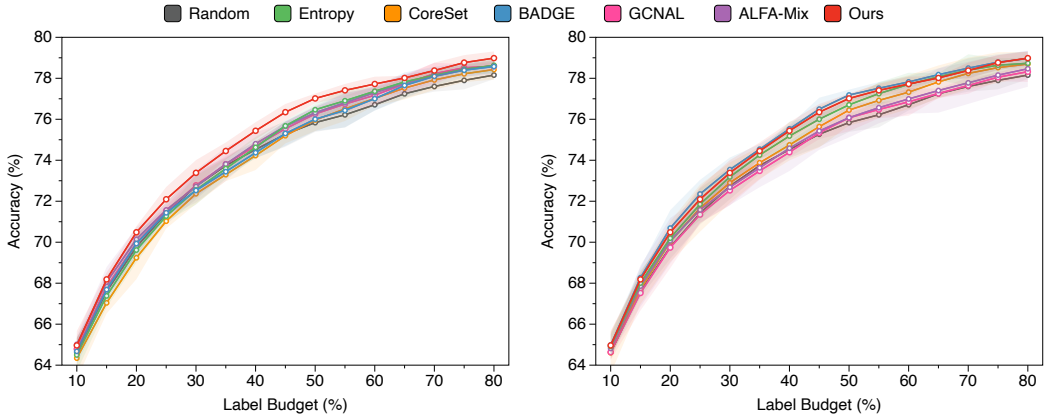
(a) Global Model (b) Local-only Model

Figure 20. Test accuracy on DermaMNIST, four layers of CNN, $\alpha = 0.1$, medium budget size (5%), and random initialization.



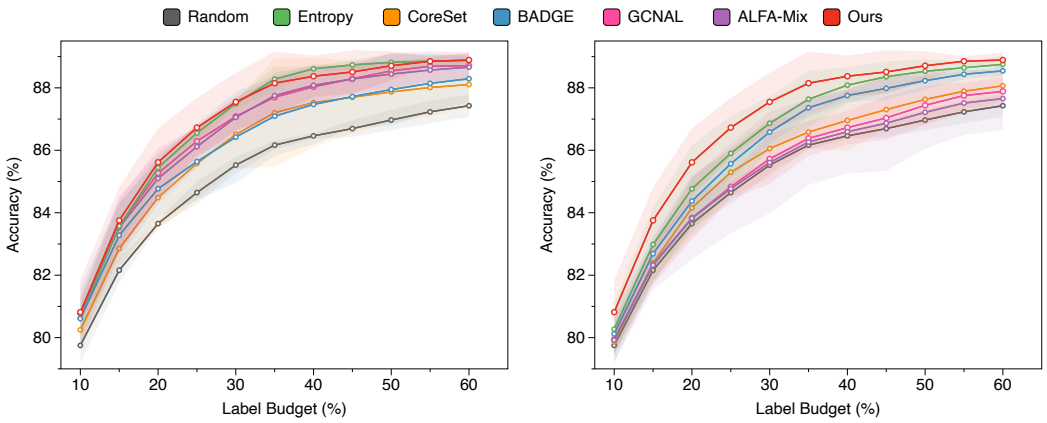
(a) Global Model (b) Local-only Model

Figure 21. Test accuracy on OrganAMNIST, four layers of CNN, $\alpha = 0.1$, medium budget size (5%), and random initialization.



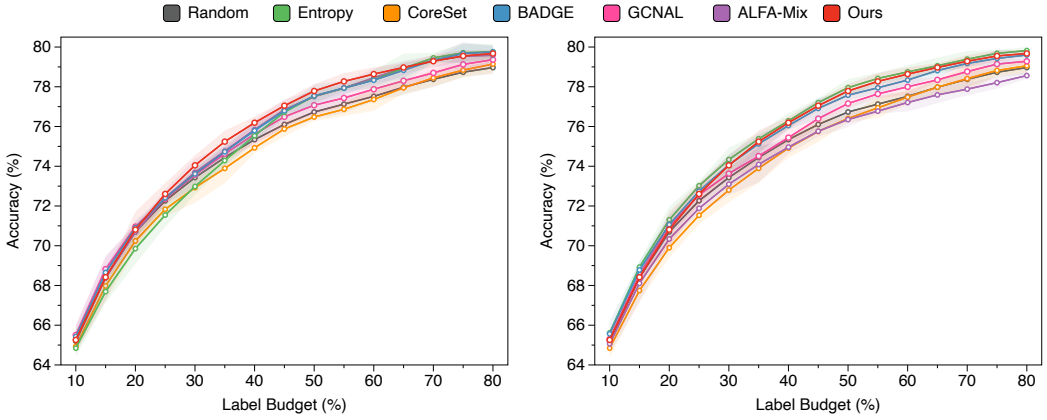
(a) Global Model (b) Local-only Model

Figure 22. Test accuracy on CIFAR-10, four layers of CNN, $\alpha = 1.0$, medium budget size (5%), and random initialization.



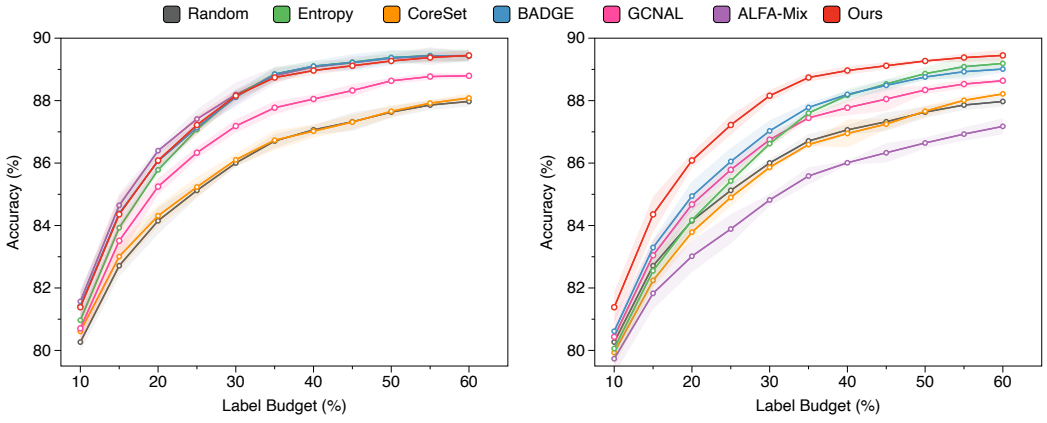
(a) Global Model (b) Local-only Model

Figure 23. Test accuracy on SVHN, four layers of CNN, $\alpha = 1.0$, medium budget size (5%), and random initialization.



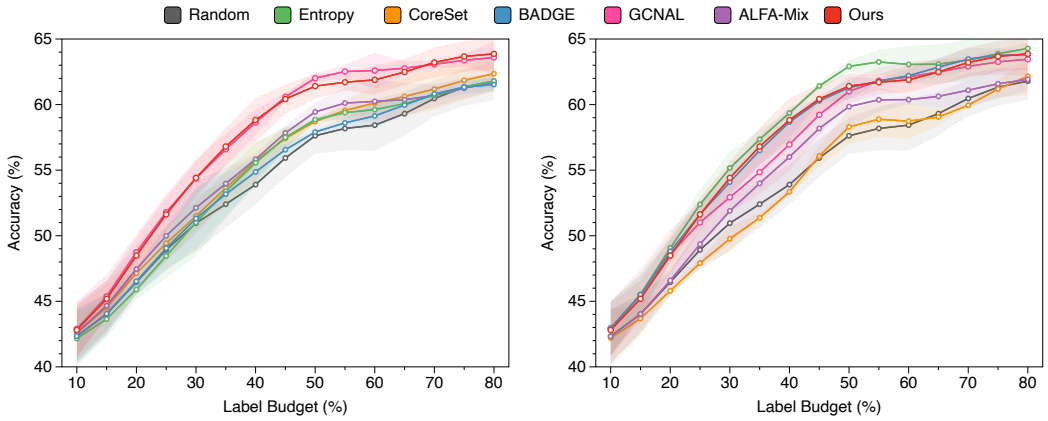
(a) Global Model (b) Local-only Model

Figure 24. Test accuracy on CIFAR-10, four layers of CNN, $\alpha = \infty$, medium budget size (5%), and random initialization.



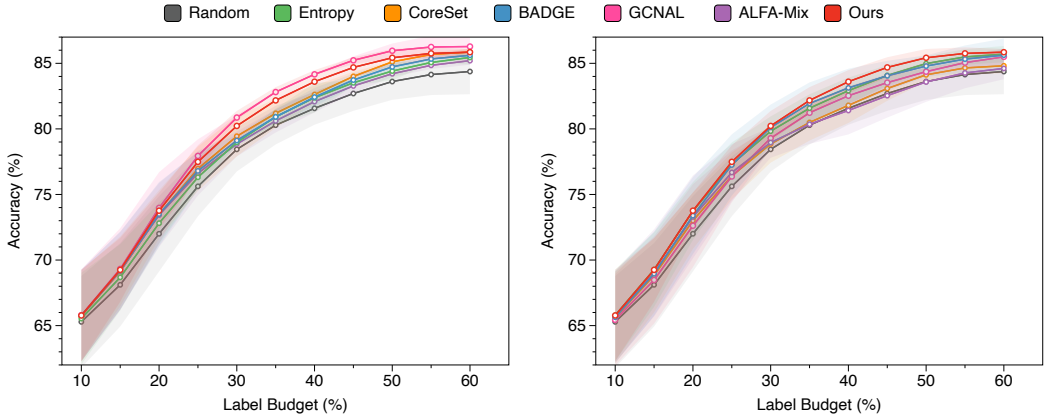
(a) Global Model (b) Local-only Model

Figure 25. Test accuracy on SVHN, four layers of CNN, $\alpha = \infty$, medium budget size (5%), and random initialization.



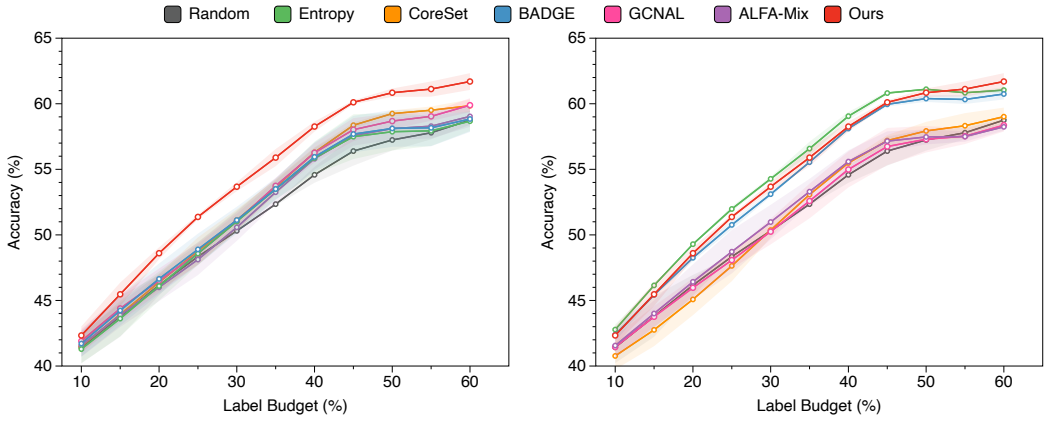
(a) Global Model (b) Local-only Model

Figure 26. Test accuracy on CIFAR-10, MobileNet, $\alpha = 0.1$, medium budget size (5%), and random initialization.



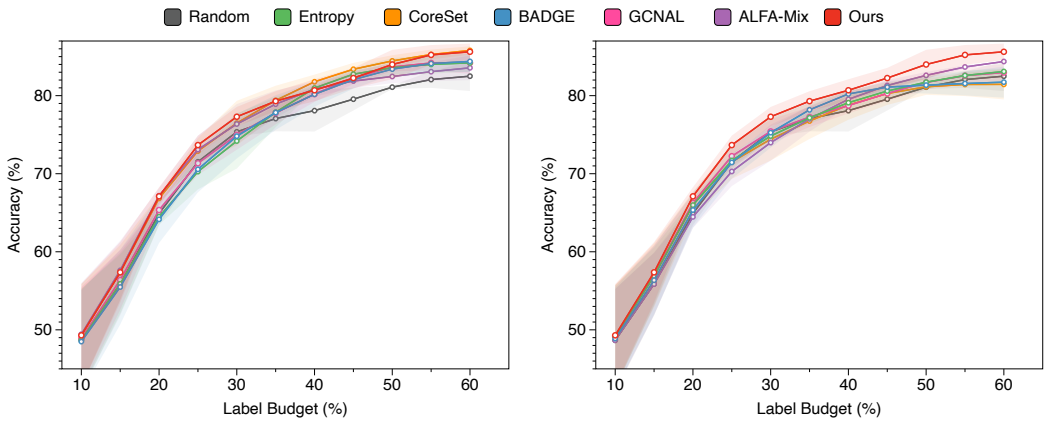
(a) Global Model (b) Local-only Model

Figure 27. Test accuracy on SVHN, MobileNet, $\alpha = 0.1$, medium budget size (5%), and random initialization.



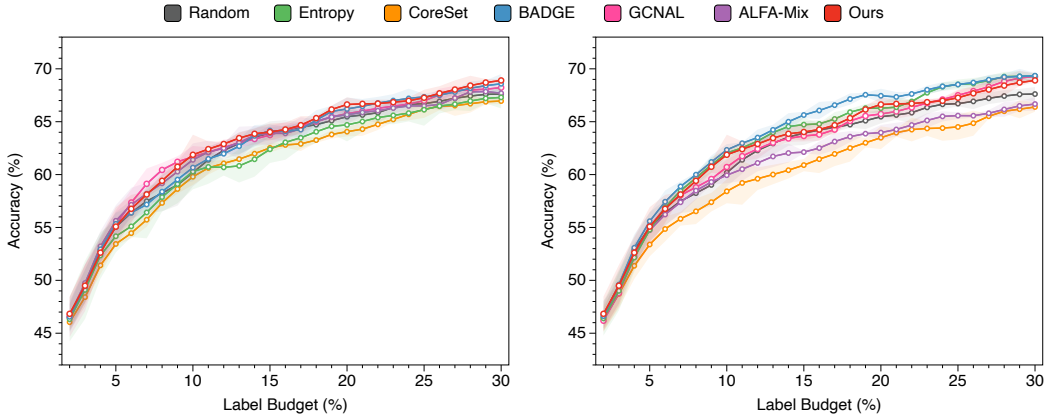
(a) Global Model (b) Local-only Model

Figure 28. Test accuracy on CIFAR-10, ResNet-18, $\alpha = 0.1$, medium budget size (5%), and random initialization.



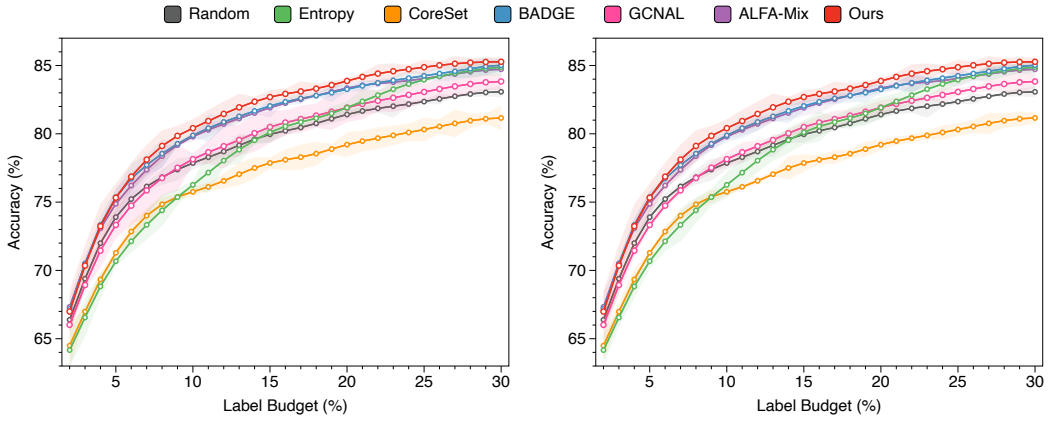
(a) Global Model (b) Local-only Model

Figure 29. Test accuracy on SVHN, ResNet-18, $\alpha = 0.1$, medium budget size (5%), and random initialization.



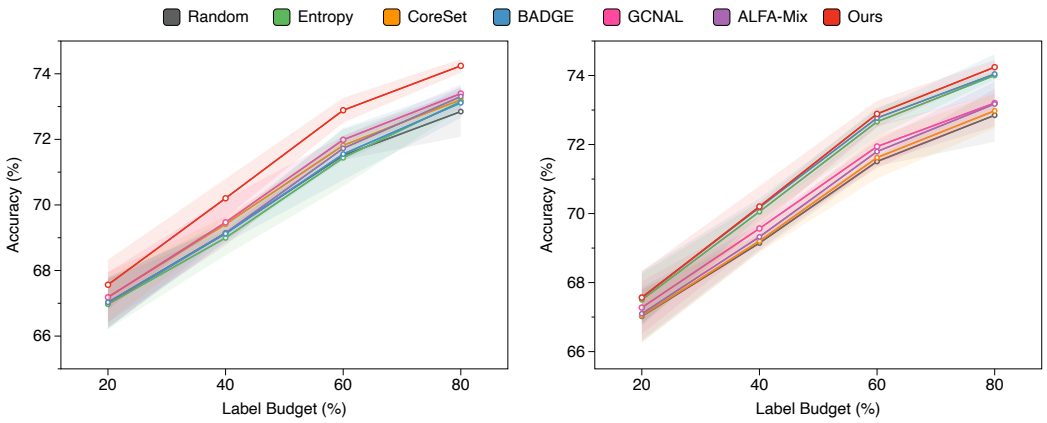
(a) Global Model (b) Local-only Model

Figure 30. Test accuracy on CIFAR-10, four layers of CNN, $\alpha = 0.1$, small budget size (1%), and random initialization.



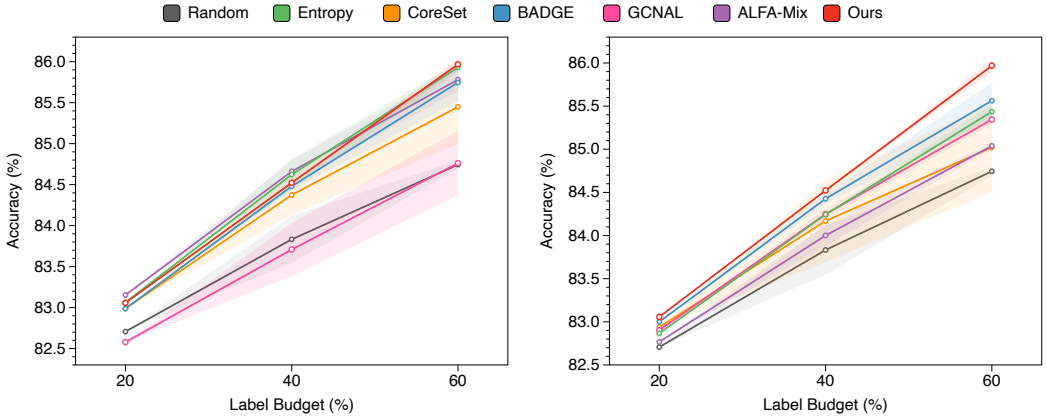
(a) Global Model (b) Local-only Model

Figure 31. Test accuracy on SVHN, four layers of CNN, $\alpha = 0.1$, small budget size (1%), and random initialization.



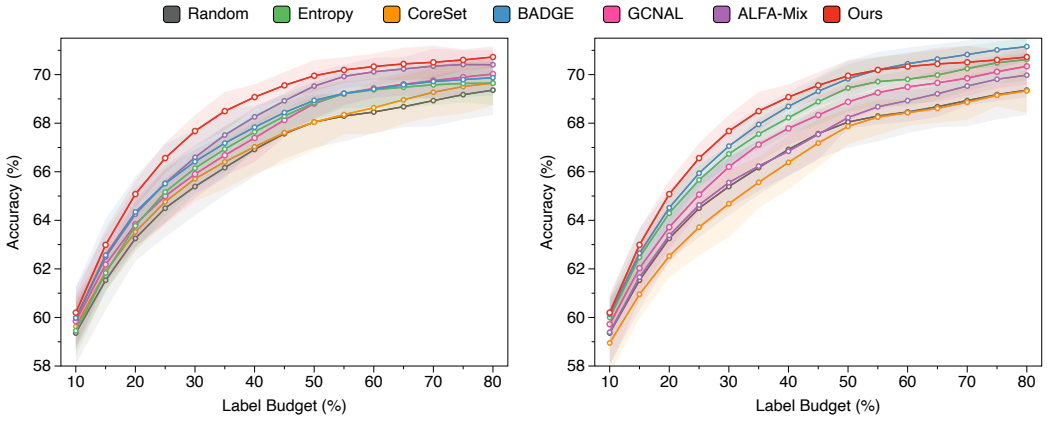
(a) Global Model (b) Local-only Model

Figure 32. Test accuracy on CIFAR-10, four layers of CNN, $\alpha = 0.1$, large budget size (20%), and random initialization.



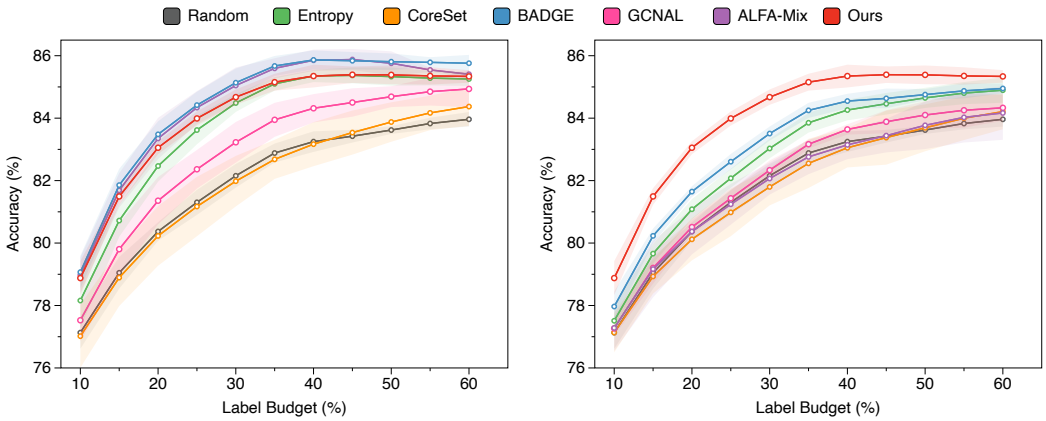
(a) Global Model (b) Local-only Model

Figure 33. Test accuracy on SVHN, four layers of CNN, $\alpha = 0.1$, large budget size (20%), and random initialization.



(a) Global Model (b) Local-only Model

Figure 34. Test accuracy on CIFAR-10, four layers of CNN, $\alpha = 0.1$, medium budget size (5%), and continue initialization.



(a) Global Model (b) Local-only Model

Figure 35. Test accuracy on SVHN, four layers of CNN, $\alpha = 0.1$, medium budget size (5%), and continue initialization.