The Appendix includes the detailed hyper-parameter settings, client-side data distribution examples, additional results on VGG11 and ResNet20 for different data rates between high-end and low-end groups, degrees of inter-client label skew and client-side models, and results when there are three groups of clients, as listed below:

- **Section A.1**: Hyper-parameter settings.
- **Section A.2**: Data distribution in clients for CIFAR10 with non-IID data corresponding to $\lambda = 0.05$ and $\lambda = 0.3$.
- **Section B.1**: Training-time and accuracy performance of HeteroSFL for VGG11 on CIFAR10 with client-side data having $\lambda = 0.3$ non-IID and IID data distributions.
- **Section B.2**: Accuracy performance of HeteroSFL with different data rates between high-end and low-end groups for VGG11 on CIFAR10.
- **Section B.3**: Training-time and accuracy performance of HeteroSFL for ResNet20 on CIFAR100 with client-side data having $\lambda = 0.05$ and $\lambda = 0.3$ non-IID data distributions.
- **Section B.4**: Accuracy performance of HeteroSFL with and without BDKS for ResNet20 on CIFAR100.
- **Section B.5**: Accuracy performance of HeteroSFL with and without W2N knowledge sharing for VGG11 on CIFAR10 and ResNet20 on CIFAR100.
- **Section B.6**: Training-time and accuracy performance of HeteroSFL for VGG11 and ResNet20 for different client-side models.
- **Section B.7**: Extending BDKS to multiple groups - low-end, middle-end, and high-end.

A Details of Experimental Settings

A.1 Hyper-parameters

We use NVIDIA GeForce RTX 3090 to run the experiments. The detailed experimental settings are as follows:

- **Epoch**: Every experiment is run for 200 epochs.
- **Batch size**: For a 10-client SFL system, the batch size is 13 samples per client so $10 \times 13 = 130$ samples per client in server; For 50-client SFL system, the batch size is 3 samples per client and so $50 \times 3 = 150$ samples per client in server.
- **Learning rate**: The learning rate for server-side model is 0.01 for VGG11 and 0.1 for ResNet20. The learning rate for client-side model is 0.02 for both VGG11 and ResNet20. The learning rate is reduced during training using the *CosineAnnealingLR* scheduler in Pytorch.
- **Optimizer**: SGD is used as the optimizer with momentum=0.9, weight_decay=5e-4.
- τ **in Logit Calibration Zhang et al. (2022)**: τ is the hyperparameter for logit calibration and is set as 5.
- θ **in Section 4.2**: We define θ to be the percentage of samples in an underrepresented class compared to that in the global data set. $\theta = 10\%$ in our experiments.
- α **in Eq.8:** α controls the magnitude of N2W knowledge sharing and is a function of the model and dataset. Specifically, the best value of α changes when the client-side model and the number of BL channel changes; but does not change with the number of clients, and the non-IID degree of data distribution. We list the α values for the different configurations used in the experiments below:
  - **VGG11 with different client-side models**
    - **2 convolution layers**: α is set as 4.0 when using 1-channel narrow BL and 16-channel wide BL.
    - **3 convolution layers**: α is set as 3.2, 2.8 and 4.0 when using 4-, 2-, and 1-channel narrow BL, respectively, for 16-channel wide BL. α is set as 3 when using 8- and 4-channel wide BL for 1-channel narrow BL.
    - **4 convolution layers**: α is set as 3.5 when using 2-channel narrow BL and 16-channel wide BL.
    - **Multiple groups**: α is set as 2.6 and 0.5 when clients are separated into three and four groups, respectively.
  - **ResNet20 with different client-side models**
    - **3 residual blocks**: α is set as 1.5 when using 1-channel narrow BL for 16-channel wide BL.
    - **4 residual blocks**: α is set as 2.0 when using 2-channel narrow BL for 16-channel wide BL.
    - **5 residual blocks**: α is set as 1.4, 1.3 and 0.5 when using 4-, 2-, and 1-channel narrow BL, respectively, for 16-channel wide BL.

A.2 DATA DISTRIBUTION IN DIFFERENT CLIENTS

We set $\lambda = 0.05$ and $\lambda = 0.3$ for non-IID distributed data in clients, where λ is the Dirichlet distribution parameter used to control the non-IID degrees Hsu et al. (2019). Lower λ indicates more significant non-IID degree. The following figures show the corresponding data distributions in different clients for CIFAR10 in the 10-client SFL system when $\lambda = 0.05$ and $\lambda = 0.3$. When $\lambda = 0.05$, the samples in every client are mostly from 1 to 2 classes and when $\lambda = 0.3$, every client contains samples from 4 to 5 classes.

(a) $\lambda = 0.05$ (b) $\lambda = 0.3$

Figure 4: CIFAR10 data distribution in clients for a 10-client system

B ADDITIONAL RESULTS

B.1 TRAINING-TIME AND ACCURACY PERFORMANCE OF HETEROSFL FOR VGG11 ON CIFAR10

In Section 5.2, we provided the accuracy performance of different competing methods as a function of training time reduction under $\lambda = 0.05$ non-IID data for VGG11 on CIFAR10 (Table 1). To demonstrate the effectiveness of HeteroSFL for other data distribution, we present results for client-level non-IID data with $\lambda = 0.3$ in Table 4 and client-level IID data in Table 5. We compare the performance of HeteroSFL with other competing methods, namely, SampleReduce, Top-k sparsity, and IdBL, for different training time reduction ratios. We study the effect when the number of high-end clients changes from 30% to 80%. Compared to baseline SFL, all competing methods achieve significant training time reduction with some accuracy loss, with HeteroSFL achieving the lowest accuracy loss for the same training time reduction. Specifically, compared with baseline SFL, HeteroSFL reduces training time by $16\times$ to $128\times$ with 0.28% to 3.54% accuracy loss for $\lambda = 0.3$ non-IID data and 1.46% to 4.14% accuracy loss for IID data. For the same training time reduction, HeteroSFL outperforms SampleReduce, Top-k sparsity, and IdBL by up to 35.59%, 14.35% and 6.98%, for $\lambda = 0.3$ non-IID data, and 25.63%, 5.93% and 9.90% for IID data.

Table 4: Accuracy of SampleReduce (SR), Top-k Sparsity, identical-sized BLs (IdBL), and HeteroSFL with 30% and 80% of high-end clients under $\lambda = 0.3$ non-IID data for VGG11 on CIFAR10. The baseline SFL accuracy is 84.62%.

Training-time reduction	30% high-end clients				80% high-end clients			
	SR	Top-k Sparsity	IdBL	HeteroSFL	SR	Top-k Sparsity	IdBL	HeteroSFL
$16\times$	62.94	82.77	84.34	**84.34**	62.94	82.77	84.34	**84.34**
$64\times$	47.76	72.85	**83.50**	83.35	58.05	81.88	83.50	**84.40**
$128\times$	47.47	66.73	80.73	**81.08**	59.27	80.54	80.73	**83.62**
$256\times$	46.51	65.06	75.73	**76.37**	57.95	80.33	75.73	**82.71**

Table 5: Accuracy of SampleReduce (SR), Top-k Sparsity, identical-sized BLs (IdBL), and HeteroSFL with 30% and 80% of high-end clients under IID data distribution for VGG11 on CIFAR10. The baseline SFL accuracy is 89.61%.

Training-time reduction	30% high-end clients				80% high-end clients			
	SR	Top-k Sparsity	IdBL	HeteroSFL	SR	Top-k Sparsity	IdBL	HeteroSFL
$16\times$	70.37	87.81	88.15	**88.15**	70.37	87.81	88.15	**88.15**
$64\times$	61.62	84.09	86.50	**87.25**	68.20	86.59	86.50	**87.97**
$128\times$	60.81	81.51	83.42	**85.47**	68.02	86.02	83.42	**87.80**
$256\times$	60.81	78.02	77.88	**83.95**	68.06	85.32	77.88	**87.78**

B.2 ACCURACY PERFORMANCE OF HETEROSFL WITH DIFFERENT DATA RATES BETWEEN HIGH-END AND LOW-END GROUPS FOR VGG11 ON CIFAR10

Table.6 shows the performance of HeteroSFL when the data rates between high-end and low-end group change from $4:1$ to $16:1$ for $\lambda = 0.05$ data with 60% clients in the high-end group. Given $256\times$ training time reduction, HeteroSFL outperforms SampleReduce, Top-k sparsity, HetBL and IdBL by up to 49.47%, 12.46%, 2.60% and 4.07%, for $4:1$ data rate ratio, and 34.98%, 4.70%, 4.15% and 5.03% for $16:1$ data rate ratio. The improvement of HeteroSFL over IdBL becomes higher when the data rate difference is larger since the additional information sent by the high-end group increases.

Table 6: Accuracy of SampleReduce (SR), Top-k Sparsity, identical-sized BLs (IdBL), HetBL and HeteroSFL when the data rates between high-end and low-end group change with 60% clients in high-end groups under $\lambda = 0.05$ non-IID data for VGG11 on CIFAR10. The training time reduction is $256\times$.

Data rate ratios	**4:1**	**8:1**	**16:1**
SampleReduce	14.67	22.89	30.12
Top-k Sparsity	51.68	53.70	59.40
IdBL	60.07	60.07	60.07
HetBL	61.54	62.04	60.95
HeteroSFL	**64.14**	**66.22**	**65.10**

B.3 TRAINING-TIME AND ACCURACY PERFORMANCE OF HETEROSFL FOR RESNET20 ON CIFAR100

In Section 5.2, we provided the accuracy performance of different competitive methods as a function of training time reduction under $\lambda = 0.05$ non-IID data for VGG11 on CIFAR10 (Table 1). To demonstrate the effectiveness of HeteroSFL for other models, we compare the performance of HeteroSFL with other competitive methods and different training time reduction ratios for ResNet20 on CIFAR100. We study the performance for non-IID data distributions with $\lambda = 0.05$ (Table 7) and $\lambda = 0.3$ (Table 8) for 30% and 80% high-end clients.

We find that for ResNet20, all competing methods achieve significant training time reduction with some accuracy loss, with HeteroSFL achieving the lowest accuracy loss for the same training time reduction. Compared with baseline SFL, HeteroSFL reduces training time by $2\times$ to $16\times$ with no more than 1.47% and 4.07% accuracy loss for $\lambda = 0.05$ and 0.3 non-IID data, respectively. For the same training time reduction, HeteroSFL outperforms SampleReduce, Top-k sparsity, and IdBL by up to 10.48%, 5.69% and 7.48% for $\lambda = 0.05$ non-IID data, and 9.53%, 7.71% and 7.58% for $\lambda = 0.3$ non-IID data.

Table 7: Accuracy of SampleReduce (SR), identical-sized BLs (IdBL), Top-k Sparsity, and HeteroSFL with 30% and 80% of high-end clients under $\lambda = 0.05$ data distribution for ResNet20 on CIFAR100. The accuracy of basic SFL is 58.11%.

Training-time reduction	30% high-end clients				80% high-end clients			
	SR	Top-k Sparsity	IdBL	HeteroSFL	SR	Top-k Sparsity	IdBL	HeteroSFL
$2\times$	55.89	58.94	61.39	**61.39**	55.89	58.94	61.39	**61.39**
$8\times$	49.70	53.58	**59.83**	59.27	53.57	56.81	59.83	**61.01**
$16\times$	46.16	52.38	56.20	**56.64**	53.45	56.70	56.20	**58.08**
$32\times$	45.86	47.29	50.44	**52.96**	52.63	53.98	50.44	**57.92**

Table 8: Accuracy of SampleReduce (SR), identical-sized BLs (IdBL), Top-k Sparsity, and HeteroSFL with 30% and 80% of high-end clients under $\lambda = 0.3$ data distribution for ResNet20 on CIFAR100. The accuracy of basic SFL is 64.12%.

Training-time reduction	30% high-end clients				80% high-end clients			
	SR	Top-k Sparsity	IdBL	HeteroSFL	SR	Top-k Sparsity	IdBL	HeteroSFL
$2\times$	60.54	62.75	64.90	**64.90**	60.54	62.75	64.90	**64.90**
$8\times$	52.21	57.05	62.84	**63.12**	60.03	60.55	62.84	**63.68**
$16\times$	51.82	54.02	59.51	**60.05**	59.83	60.21	59.51	**61.62**
$32\times$	49.37	48.99	55.25	**56.70**	59.66	59.64	55.25	**62.83**

B.4 EFFECTIVENESS OF BDKS FOR RESNET20 ON CIFAR100

In Section 5.3, we showed the accuracy of using HeteroSFL with and without BDKS for VGG11 on CIFAR10 for $256\times$ training time reduction (Table 1 and Table 3). To further illustrate the importance of BDKS in other models, we evaluate the performance of BDKS for ResNet20 in a 10-client system with $32\times$ training time reduction and present it in Table 9. Similar to the VGG11 case, the accuracy of HeteroSFL without BDKS is lower than with BDKS, especially when the number of high-end clients is small. We see that BDKS helps improve accuracy (maximum 6.30%) when the number of high-end clients is less than 60%. When the number of high-end clients is higher than 60%, the high-end set of classes has sufficient number of samples from most classes, making the accuracies of HeteroSFL with and without BDKS quite close.

Table 9: Accuracy performance of HeteroSFL with and without BDKS under $\lambda = 0.05$ for ResNet20 on CIFAR10 in 10-client SFL. The training time reduction is $32\times$.

# of clients	High-end prop.	HetBL	HeteroSFL
10-clients	30%	46.66	**52.96**
	40%	50.78	**53.43**
	60%	54.23	**54.54**
	80%	57.90	**57.92**

B.5 WIDE-TO-NARROW (W2N) KNOWLEDGE SHARING

In Section 4, we observed that the gradients of narrow and wide BLs interfere with each other when training. To further investigate this phenomenon, we replace W2N knowledge sharing in BDKS with logit calibration, referred to as HeteroSFL w/o W2N; the corresponding results are shown in Table 10. For ResNet20, the accuracy of HeteroSFL w/o W2N is significantly worse than HeteroSFL, especially with 80% high-end clients. Here the accuracy of using logit calibration is 3.82% lower than using W2N due to the interference between narrow and wide BLs. This illustrates that W2N knowledge sharing is necessary to train the narrow BLs in high-end group for higher accuracy.

Table 10: Accuracy performance of HeteroSFL without W2N (w/o W2N) or W2N knowledge distillation to train narrow BL under $\lambda = 0.05$ for VGG11 on CIFAR10 and ResNet20 on CIFAR100. The training time reduction is $256\times$ for VGG11 and $32\times$ for ResNet20.

Model	High-end prop.	w/o W2N	W2N
VGG11	40%	**62.82**	62.39
	60%	63.63	**65.10**
	80%	**67.04**	66.82
ResNet20	40%	52.63	**53.43**
	60%	53.22	**54.54**
	80%	54.10	**57.92**

B.6 TRAINING-TIME AND ACCURACY PERFORMANCE USING HETEROSFL OF DIFFERENT CLIENT-SIDE MODELS

In the main body of the paper, the model for VGG11 is split such that the client-side model has 3 convolution layers, and the model for ResNet20 is split such that the client-side model has 5 residual blocks. To further demonstrate the effectiveness of HeteroSFL, we use different client-side models. Specifically, for VGG11, the client-side models have 2, 3, or 4 convolution layers, and for ResNet20, the client-side model has 3, 4, and 5 residual blocks. Fig.5 and Fig.6 show the accuracy comparisons of HeteroSFL and other training time reduction methods for different client-side models. For $128\times$ training time reduction, our proposed HeteroSFL still achieves the best accuracy among all the methods.

For VGG11, compared with SampleReduce, when the number of client-side convolution layers is 2, 3 and 4, HeteroSFL achieves up to 37.05%, 52.16%, and 53.79% accuracy improvement; compared with Top-k Sparsity, HeteroSFL achieves up to 9.44%, 9.61%, and 11.08% accuracy improvement; and compared with identical-sized BL, HeteroSFL achieves up to 9.48%, 3.31%, and 4.86% average accuracy improvement, respectively.

Similarly, for ResNet20, compared with SampleReduce, when the number of the client-side residual blocks is 3, 4 and 5, HeteroSFL achieves up to 19.28%, 8.32%, and 10.08% accuracy improvement; compared with Top-k Sparsity, HeteroSFL achieves up to 4.85%, 3.43%, and 3.86% accuracy improvement; and compared with identical-sized BL, HeteroSFL achieves up to 1.94%, 1.98%, and 1.88% accuracy improvement, respectively.

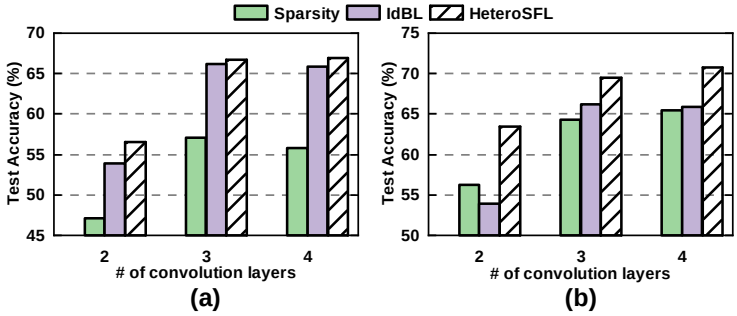

Figure 5: Accuracy performance of SFL using Identical-sized BL (IdBL), Top-k Sparsity, and HeteroSFL with different client-side models under $\lambda = 0.05$ for VGG11 on CIFAR10. The training time reduction is $128\times$. The data rate ratio between high-end and low-end groups is $16 : 1$. The SampleReduce method is always lower than 40% thus is not shown here. (a) 30% high-end clients. (b) 80% high-end clients.

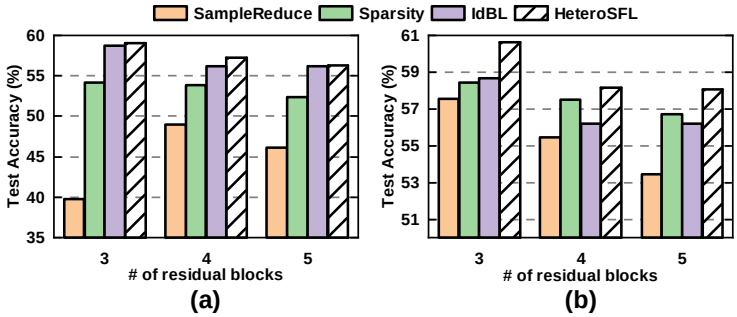

Figure 6: Accuracy performance of SFL using SampleReduce, Identical-sized BL (IdBL), Top-k Sparsity, and HeteroSFL with different client-side models under $\lambda = 0.05$ for ResNet20 on CIFAR100. The training time reduction is $16\times$. The data rate ratio between high-end and low-end groups is $16 : 1$. (a) 30% high-end clients. (b) 80% high-end clients.

B.7 EXTENDING BDKS TO MULTIPLE GROUPS

In the main body of the paper, we assumed that there are two groups of clients, low-end and high-end, where the low-end group has $4 \times -16\times$ lower communication data rates compared to the high-end clients. We proposed bi-directional knowledge sharing (BDKS) in Section 4.2 to address the overfitting of wide BL model caused by the underrepresented classes in high-end clients.

What if the clients subscribe to more than two different communication protocols with more than two different data rates? Here we extend BDKS to cases where clients are separated into three sets, low-end, middle-end, and high-end. The three different sets of clients use narrow BL, medium BL, or wide BL to achieve different levels of data compression. Here the narrow BL is a subnetwork of medium BL, and medium BL is a subnetwork of wide BL. With client-level non-IID data, the underrepresented classes are present in both middle-end and high-end clients. The loss function of BDKS for three sets of clients is as follows:

- **Low-end clients (Eq.9)**: The narrow BL is trained by the logit calibration loss function.
- **Middle-end clients (Eq.10)**: The medium BL is trained by logit calibration and knowledge sharing from the narrow BL model to the medium BL model (N2M); the narrow BL in middle-end clients is trained by knowledge sharing from the medium BL model to the narrow BL model (M2N) to eliminate the interference of gradients from different-sized BL.
- **High-end clients (Eq.11)**: The wide BL is trained by logit calibration and knowledge sharing from the medium BL model to the wide BL model (M2W); the medium BL is trained by knowledge sharing from the wide BL model to the medium BL model (W2M) and knowledge sharing from the narrow BL model to the medium BL model (N2M) to mitigate the overfitting due to underrepresented classes in middle-end clients; the narrow BL is trained by knowledge sharing from the medium BL model to the narrow BL model (M2N).

$$\mathcal{L}_{k,lowend}^{BDKS} = \underbrace{\mathcal{L}_k^{cal}(f_k^{p_N}(x), y)}_{\text{training narrow BL in low-end clients}} \tag{9}$$

$$\begin{aligned}\mathcal{L}_{k,middleend}^{BDKS} = &\underbrace{\mathcal{L}_k^{cal}(f_k^{p_M}(x), y) + \alpha \cdot \mathcal{L}_k^{N2M}(f_k^{p_M}(x), f_k^{p_N}(x))}_{\text{training medium BL in middle-end clients}} \\ &+ \underbrace{\mathcal{L}_k^{M2N}(f_k^{p_N}(x), f_k^{p_M}(x))}_{\text{training narrow BL in middle-end clients}}\end{aligned} \tag{10}$$

$$\begin{aligned}\mathcal{L}_{k,highend}^{BDKS} = &\underbrace{\mathcal{L}_k^{cal}(f_k^{p_W}(x), y) + \alpha \cdot \mathcal{L}_k^{M2W}(f_k^{p_W}(x), f_k^{p_M}(x))}_{\text{training wide BL in high-end clients}} \\ &+ \underbrace{\mathcal{L}_k^{W2M}(f_k^{p_M}(x), f_k^{p_W}(x)) + \alpha \cdot \mathcal{L}_k^{N2M}(f_k^{p_M}(x), f_k^{p_N}(x))}_{\text{training medium BL in high-end clients}} \\ &+ \underbrace{\mathcal{L}_k^{M2N}(f_k^{p_N}(x), f_k^{p_M}(x))}_{\text{training narrow BL in high-end clients}}\end{aligned} \tag{11}$$

where $f_k^{p_W}(x)$, $f_k^{p_M}(x)$ and $f_k^{p_N}(x)$ represent the logits generated by wide, medium and narrow BL model.