Appendix

This appendix provides additional details. Specifically, the following contents are included:

- Hyperparameter settings
- Performance evaluation under high sparsity
- Additional comparisons
- Extensive ablation studies

A. Hyperparameter Settings

Hyperparameter	**Symbol**	**LeNet**	**ResNet-18**
Learning rate	η	0.01	0.01
Batch size	B	32	32
Target Sparsity	s	0.8	$\{0.9, 0.95, 0.98, 0.99\}$
PFFDST Sparsity 1	s_1	0.9	$\frac{1-s}{2}$
PFFDST Sparsity 2	s_2	0.8	s
Differential Sparsity	f_1, f_2	0.05	$\frac{1-s}{4}$
Clients	C	400	200
Clients per Round	K	20	20
Unbalance rate for Dirichlet distribution	β	0.1	0.1
Communication rounds	R	$200, \ldots$	$200, \ldots$
Mask Readjustment rounds	R_{end}	$\frac{1}{4}R$	$\frac{1}{4}R$
Period of mask adjustment	ΔR	10	10
Optimizer		SGD	SGD
Momentum (if applicable)		0.9	0.9
Weight Decay (if applicable)		1×10^{-4}	1×10^{-4}
L2 regularization strength		0.001	0.001

Table 1: Hyperparameter Settings

Communication Cost (GB)	FedDST (R)	PFFDST (R)
4	200	350
8	400	700
12	600	1050
16	800	1400

Table 2: Communication Rounds (R) for LeNet under Different Bandwidth Constraints with Sparsity $s = 0.8$

Communication Cost (GB)	FedDST (R)	PFFDST (R)
34	200	350
68	400	700
102	600	1050
136	800	1400

Table 3: Communication Rounds (R) for ResNet under Different Bandwidth Constraints with Sparsity $s = 0.9$

Table 1 summarizes the hyperparameter settings used for both LeNet and ResNet-18 models, including learning rate, batch size, target sparsity levels for PFFDST, client participation, and optimization parameters, etc.

Bandwidth constraints determine the number of communication rounds in federated learning. PFFDST adopts a sparser communication strategy than FedDST by synchronizing fewer parameters per round, allowing it to compensate with a higher total number of communication rounds. Additionally, the introduction of a differential rate (Rend) incurs extra communication costs. To maintain the same overall communication cost, we set the total number of communication rounds for PFFDST to be 7/4 that of FedDST. Specifically, in a scenario where 20 clients are selected per round out of 400 total clients, Tables 2 and Table 3 present the required number of communication rounds (R) for implementing FedDST and PFFDST using the LeNet and ResNet architectures, respectively.

B. Performance under High Sparsity

This section explores the performance of PFFDST under high-sparsity constraints, demonstrating its ability to maintain high accuracy even when the network's trainable parameters are drastically reduced.

CIFAR-10 with High Sparsity. Table 4 provides a comparative analysis of FedDST and PFFDST on the CIFAR-10 dataset using a ResNet-18 network with high sparsity ratios ($s = 0.98, 0.99$). Notably, despite the aggressive parameter reduction, PFFDST consistently outperforms FedDST, achieving an average accuracy improvement of approximately 4%. These results highlight the robustness and efficiency of PFFDST in scenarios with severely limited communication bandwidth

CIFAR-100 with High Sparsity. To further validate the capabilities of PFFDST under high sparsity conditions, we conduct experiments on the more challenging CIFAR-100 dataset, which features a larger number of classes. Table 5 summarizes the results for the same high-sparsity ratios ($s = 0.98, 0.99$). Consistent with the CIFAR-10 findings, PFFDST demonstrates its superiority, achieving an average accuracy improvement of 6% compared to FedDST. These results highlight the effectiveness and generalizability of our approach across diverse datasets and demanding sparsity levels.

C. Additional Comparisons

In this section, we present additional experiments to comprehensively evaluate the performance of our proposed method. Specifically, we compare it with state-of-the-art (SoTA) methods on image classification and natural language processing (NLP) tasks, as well as analyze its efficiency under varying communication constraints.

Comparison with SoTA Methods. We compare the performance of our method with FedTiny (Huang et al. 2023), FedMef (Huang et al. 2024), and FLASH (Babakniya et al. 2023). The results for FedTiny and FedMef are directly obtain from their original papers, while for FLASH, we retrain the model. For PFFDST, we integrat parameter freezing method into FLASH. All methods were evaluated under the same experimental settings, following those used in FedMef, including the CIFAR-10 dataset, a ResNet-18 model, a sparsity level of 0.95, a Dirichlet parameter $\alpha = 0.5$, 100 clients, and a communication constraint of 39GB. (as shown in Table 6).

Commu. Cost (GB)	8	16	24	32	4	8	12	16
Train. FLOPs ($\times 10^{13}$)	1.6	3.2	4.8	6.4	0.8	1.6	2.4	3.2
FedAvgM (dense)	15.09±1.7	20.99±0.7	25.11±1.6	28.15±2.7	11.34±1.2	15.09±1.7	19.04±2	20.99±0.7
FedProx (dense)	17.81±1.8	21.01±2.2	27.19±1.6	30.91±2.5	14.71±3.2	17.81±1.8	20.98±2.3	21.01±2.2
	s = 0.98				s = 0.99			
FedDST	50.01±1.8	51.86±1.8	54.03±0.3	54.60±0.9	47.25±1.9	49.80±1.3	50.62±1.0	52.21±0.6
FedDST+FedProx	49.06±2.2	50.48±2.4	53.16±1.2	53.75±1.7	47.78±1.6	49.51±1.1	50.95±1.5	50.98±1.2
PFFDST (ours)	**54.34**±1.2	**55.50**±1.0	**56.69**±1.6	**58.20**±1.5	**51.73**±0.9	**52.50**±1.6	**54.70**±0.3	**55.34**±0.7

Table 4: Experiment results of PFFDST compared with other dense and sparse methods given cumulative bandwidth limits, on non-iid CIFAR-10 using ResNet-18. We fix $s = 0.98, 0.99$.

Commu. Cost (GB)	8	16	24	32	4	8	12	16
Train. FLOPs ($\times 10^{13}$)	1.6	3.2	4.8	6.4	0.8	1.6	2.4	3.2
FedAvgM (dense)	7.05±0.3	12.97±0.7	16.35±0.9	19.55±0.7	3.50±0.3	7.05±0.3	10.08±0.7	12.97±0.7
FedProx (dense)	6.89±0.6	12.90±0.8	16.12±0.4	20.07±0.3	3.61±0.2	6.89±0.6	10.60±0.5	12.90±0.8
	s = 0.98				s = 0.99			
FedDST	28.95±0.1	30.51±0.5	31.12±0.5	31.42±0.4	26.93±0.4	27.91±0.2	28.68±0.5	29.33±0.6
FedDST+FedProx	29.50±0.6	30.17±0.4	30.93±0.4	31.26±0.2	26.67±0.7	28.01±0.4	28.50±0.3	28.88±0.3
PFFDST (ours)	**35.14**±0.9	**35.25**±0.3	**35.98**±0.5	**37.11**±0.6	**31.01**±1.6	**34.22**±1.4	**34.31**±0.2	**34.80**±0.4

Table 5: Experiment results of PFFDST compared with other dense and sparse methods given cumulative bandwidth limits, on non-iid CIFAR-100 using ResNet-18. We fix $s = 0.98, 0.99$.

Methods	FedDST	FedTiny	FedMef	FLASH	**PFFDST**
Accuracy(%)	72.8	71.8	73.6	76.9	**78.6**

Table 6: Comparison of PFFDST with FedTiny, FedMef, and FLASH on CIFAR-10 using ResNet-18 under 0.95 sparsity.

	Sparsity = 0.9			Sparsity = 0.95		
Commu. Cost (GB)	17	34	51	8	16	24
FLASH	45.12	61.73	72.57	43.91	59.73	69.57
PFFDST	**59.30**	**72.22**	**77.29**	**58.06**	**71.66**	**74.82**

Table 7: Comparison of PFFDST and FLASH on CIFAR-10 using ResNet-18 under various communication constraints.

Performance with Limited Communication. Furthermore, we evaluate the performance of PFFDST and FLASH under various communication constraints on the CIFAR-10 dataset using ResNet-18. For these experiments, we use 100 clients, set $\alpha = 0.1$, and test the models under two sparsity levels (0.9 and 0.95). As shown in Table 7, PFFDST consistently outperformed FLASH across three different communication cost scenarios. Notably, the improvements are more significant under lower communication budgets, highlighting the superior efficiency of PFFDST in resource-constrained settings.

Comm. Cost (GB)	6.4	8.8	19.2
FedDST (%)	68.66	77.41	79.03
FLASH (%)	80.27	81.65	83.02
PFFDST (%)	**81.30**	**82.33**	**83.83**

Table 8: Results for fine-tuning SST dataset on BERT-base model.

Evaluation on NLP Tasks. We evaluate the performance of PFFDST and FLASH on NLP tasks. To ensure a fair comparison with FLASH, we follow its experimental setup and integrate our PFFDST method into their framework. Specifically, we fine-tune the BERT-base (Devlin 2018) model on the SST-2 (Wang 2018) task. As shown in Table 8, our method consistently outperforms both FedDST and FLASH across all communication cost levels. At a communication cost of 6.4 GB, our approach achieves an accuracy of 81.30%, surpassing FLASH (80.27%) and significantly outperforming FedDST (68.66%). This trend continues at 8.8 GB, where our method achieves 82.33% accuracy, outperforming FLASH (81.65%) and FedDST (77.41%). At

the highest communication cost of 19.2 GB, our method achieves 83.83% accuracy, exceeding FLASH (83.02%) and FedDST (79.03%).

D. Extensive Ablation Study

Readjustment Ratio (%)	10	25	50	75
Accuracy (%)	60.93	61.20	61.59	60.95

Table 9: Accuracy under different readjustment ratios.

We conduct ablation experiments to evaluate the impact of the readjustment ratio. Following the experimental settings described in our paper, we perform these experiments on the CIFAR-10 dataset using ResNet-18 under the same communication limit and FLOPS, as shown in Table 9. The results indicate that our method performs well when the readjustment ratio is set between 0.25 and 0.5. A readjustment ratio that is too low hinders mask optimization, while an excessively high ratio increases communication costs.

References

Babakniya, S.; Kundu, S.; Prakash, S.; Niu, Y.; and Avestimehr, S. 2023. Revisiting Sparsity Hunting in Federated Learning: Why does Sparsity Consensus Matter? *Transactions on Machine Learning Research*.

Devlin, J. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Huang, H.; Zhang, L.; Sun, C.; Fang, R.; Yuan, X.; and Wu, D. 2023. Distributed pruning towards tiny neural networks in federated learning. In *2023 IEEE 43rd International Conference on Distributed Computing Systems (ICDCS)*, 190–201. IEEE.

Huang, H.; Zhuang, W.; Chen, C.; and Lyu, L. 2024. FedMef: Towards Memory-efficient Federated Dynamic Pruning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 27548–27557.

Wang, A. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.