FedNPC: Stochastic Noise-driven Post-hoc Classifier Calibration Method for Federated Long-tailed Learning

Supplementary Material

6. Proof of Proposition 1

Proof. We compute the gradient of $\mathcal{L}_{\text{NPC}}$ with respect to $\boldsymbol{w}_c$. Let $S_j = \sum_{c=1}^{C} e^{\boldsymbol{w}_c^\top \boldsymbol{z}_j}$. Therefore:

$$\frac{\partial \mathcal{L}_{\text{NPC}}}{\partial \boldsymbol{w}_c} = -\frac{1}{M}\sum_{j=1}^{M}\left[\frac{\partial}{\partial \boldsymbol{w}_c}(\boldsymbol{w}_{y_j}^\top \boldsymbol{z}_j) - \frac{\partial}{\partial \boldsymbol{w}_c}\log(S_j)\right].$$

Case 1. When $c = y_j$:

$$\begin{aligned}
\frac{\partial \mathcal{L}_{\text{NPC}}}{\partial \boldsymbol{w}_c} &= -\frac{1}{M}\sum_{j=1}^{M}\left[\boldsymbol{z}_j - \frac{\partial}{\partial \boldsymbol{w}_c}\log(S_j)\right] \\
&= -\frac{1}{M}\sum_{j=1}^{M}\left[\boldsymbol{z}_j - \frac{1}{S_j}\cdot\frac{\partial S_j}{\partial \boldsymbol{w}_c}\right] \\
&= -\frac{1}{M}\sum_{j=1}^{M}\left[\boldsymbol{z}_j - \frac{e^{\boldsymbol{w}_c^\top \boldsymbol{z}_j}}{\sum_{c'=1}^{C} e^{\boldsymbol{w}_{c'}^\top \boldsymbol{z}_j}}\boldsymbol{z}_j\right] \\
&= -\frac{1}{M}\sum_{j=1}^{M}\left[\boldsymbol{z}_j - \boldsymbol{p}(c|\boldsymbol{z}_j)\boldsymbol{z}_j\right] \\
&= -\frac{1}{M}\sum_{j=1}^{M}\left[(1-\boldsymbol{p}(c|\boldsymbol{z}_j))\boldsymbol{z}_j\right]. \qquad (4)
\end{aligned}$$

Case 2. When $c \neq y_j$:

$$\begin{aligned}
\frac{\partial \mathcal{L}_{\text{NPC}}}{\partial \boldsymbol{w}_c} &= -\frac{1}{M}\sum_{j=1}^{M}\left[-\frac{\partial}{\partial \boldsymbol{w}_c}\log(S_j)\right] \\
&= -\frac{1}{M}\sum_{j=1}^{M}\left[-\frac{e^{\boldsymbol{w}_c^\top \boldsymbol{z}_j}}{\sum_{c'=1}^{C} e^{\boldsymbol{w}_{c'}^\top \boldsymbol{z}_j}}\boldsymbol{z}_j\right] \\
&= \frac{1}{M}\sum_{j=1}^{M}\boldsymbol{p}(c|\boldsymbol{z}_j)\boldsymbol{z}_j. \qquad (5)
\end{aligned}$$

Combined results:

$$\begin{aligned}
\nabla_{\boldsymbol{w}_c}\mathcal{L}_{\text{NPC}} &= -\frac{1}{M}\sum_{j=1}^{M}\Big[\mathbb{I}(c=y_j)(1-\boldsymbol{p}(c|\boldsymbol{z}_j)) \\
&\qquad - \mathbb{I}(c\neq y_j)\boldsymbol{p}(c|\boldsymbol{z}_j)\Big]\boldsymbol{z}_j \\
&= -\frac{1}{M}\sum_{j=1}^{M}\left[\mathbb{I}(c=y_j) - \boldsymbol{p}(c|\boldsymbol{z}_j)\right]\boldsymbol{z}_j \\
&= \frac{1}{M}\sum_{j=1}^{M}\left[\boldsymbol{p}(c|\boldsymbol{z}_j) - \mathbb{I}(c=y_j)\right]\boldsymbol{z}_j, \qquad (6)
\end{aligned}$$

where $\mathbb{I}(\cdot)$ is the indicator function.

We compute the inner expectation $\mathbb{E}_y[\cdot|\boldsymbol{z}]$, holding $\boldsymbol{z}$ constant:

$$\begin{aligned}
\mathbb{E}_{y\sim\mathcal{U}(C)}[\nabla_{\boldsymbol{w}_c}\mathcal{L}|\boldsymbol{z}] &= \mathbb{E}_y[p(c|\boldsymbol{z})\boldsymbol{z}] - \mathbb{E}_y[\mathbb{I}(c=y)\boldsymbol{z}], \qquad (7) \\
&= p(c|\boldsymbol{z})\boldsymbol{z} - \boldsymbol{z}\cdot\mathbb{E}_y[\mathbb{I}(c=y)]. \qquad (8)
\end{aligned}$$

By linearity of expectation and noting that $P(y=c) = 1/C$:

$$\mathbb{E}_y[\nabla_{\boldsymbol{w}_c}\mathcal{L}|\boldsymbol{z}] = \left(p(c|\boldsymbol{z}) - \frac{1}{C}\right)\boldsymbol{z}. \qquad (9)$$

□

7. Proof of Theorem 1

Proof. Following the Neural Collapse (NC) phenomenon, we analyze the terminal phase of training where the model exhibits specific geometric properties on the balanced dataset. Let $\{\boldsymbol{\mu}_c\}_{c=1}^{C}$ denote the class-mean features and $\boldsymbol{\mu}_G = \frac{1}{C}\sum_{c=1}^{C}\boldsymbol{\mu}_c$ be the global mean feature vector.

The normalized class means $\tilde{\boldsymbol{\mu}}_c = \frac{\boldsymbol{\mu}_c - \boldsymbol{\mu}_G}{\|\boldsymbol{\mu}_c - \boldsymbol{\mu}_G\|}$ converges to form a simplex equiangular tight frame (ETF) with the following properties:

$$\|\boldsymbol{\mu}_c - \boldsymbol{\mu}_G\| - \|\boldsymbol{\mu}_{c'} - \boldsymbol{\mu}_G\| \to 0 \quad \forall c \neq c', \qquad (10)$$

$$\langle \tilde{\boldsymbol{\mu}}_c, \tilde{\boldsymbol{\mu}}_{c'} \rangle = -\frac{1}{C-1} \quad \forall c \neq c'. \qquad (11)$$

The classifier weight matrix $\boldsymbol{W} = [\boldsymbol{W}_1, \ldots, \boldsymbol{W}_C]$ aligns with the normalized class means $\tilde{\mathbf{M}} = [\tilde{\boldsymbol{\mu}}_1, \ldots, \tilde{\boldsymbol{\mu}}_C]$ with $\|\tilde{\boldsymbol{\mu}}_c\|_2 = 1$ through the relation:

$$\left\| \frac{\boldsymbol{W}^\top}{\|\boldsymbol{W}\|_F} - \frac{\widetilde{\mathbf{M}}}{\|\widetilde{\mathbf{M}}\|_F} \right\|_F \to 0, \qquad (12)$$

where $\|\boldsymbol{W}\|_F = \sqrt{\sum_{c=1}^{C}\|\boldsymbol{W}_c\|_2^2}$ is the Frobenius norm of the weight matrix and $\|\tilde{\mathbf{M}}\|_F = \sqrt{C}$. This alignment implies each weight vector can be expressed as:

$$\boldsymbol{W}_c = \alpha\tilde{\boldsymbol{\mu}}_c, \qquad (13)$$

where $\alpha = \frac{\|\boldsymbol{W}\|_F}{\sqrt{C}}$. From the self-duality condition and the ETF properties, we derive the norm equality:

$$\|\boldsymbol{W}_c\|_2 = \|\alpha\tilde{\boldsymbol{\mu}}_c\|_2 = \alpha \quad \forall c \in \{1, \ldots, C\}. \qquad (14)$$

The scaling factor α is class-independent, proving that all classifier weights $\{\boldsymbol{W}_c\}_{c=1}^{C}$ have equal norms. Therefore, we can proof the norm equality:

$$\|\boldsymbol{w}_c\|_2 = \|\boldsymbol{w}_{c'}\|_2 \quad \forall c \neq c'. \tag{15}$$

□

8. More experiments

Hyper-parameter discussion. Figure 7 presents the hyper-parameter analysis of FedNPC based on multiple benchmarks across varying sample sizes M, training epochs E, and learning rates η on CIFAR-LT datasets. It indicates reduced sensitivity to these critical hyper-parameters, which is a valuable property for practical federated learning deployments.

Weight norm evolution. Figure 8 visualizes how our method guides classifier weight norms from an imbalanced to a balanced distribution when integrated with different federated learning baselines. The heatmaps demonstrate that all four combinations—FedAvg, FedProx, FedDisco, and FedLWS—with our approach exhibit a clear transition over training epochs. Initially, the weight norms display high variance across classes, reflecting the underlying data imbalance. As training progresses, the norms gradually converge toward a more uniform distribution across class indices. This consistent re-balancing effect underscores our method's capability to steer diverse federated optimizers toward improved equilibrium in the classifier, irrespective of their inherent bias characteristics.

Effects of crucial parameters in FL. As shown in Fig. 9, we tune three crucial parameters of baselines in FL: the number of clients $K \in \{10, 30, 50\}$, participation ratio $R \in \{0.1, 0.3, 0.5\}$, the number of local epoch $E \in \{1, 5, 15\}$, and Non-IIDness $\alpha \in \{0.1, 1.0, 10.0\}$. The proposed method consistently enhances the performance of baseline algorithms across all experimental settings. The experiments reveal that our proposed method brings performance improvement across different FL settings.

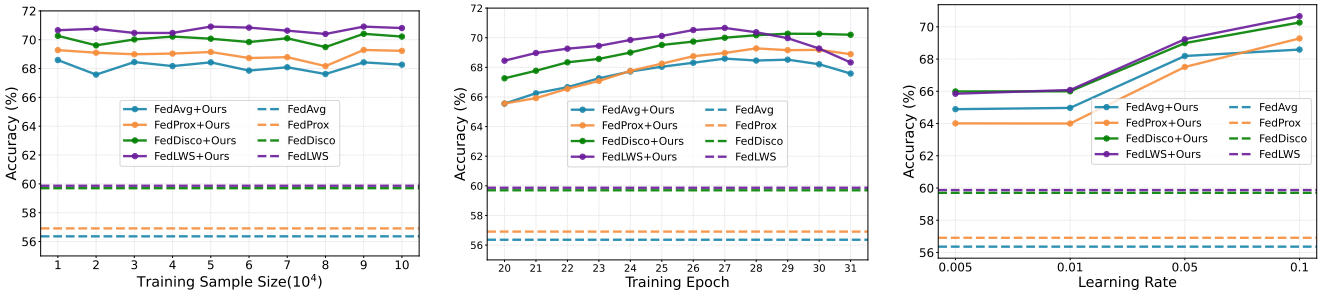
De-bias comparison on FT-LT. Table 5 evaluates debias classifier methods for federated learning on CIFAR-LT datasets under varying imbalance factors. Results show server-side methods generally outperform client-side approaches. The proposed FedAvg+FedNPC method demonstrates substantial improvements. The optimal performance is achieved by combining BALMS with FedNPC, indicating that client-side and server-side de-biasing strategies are complementary for federated long-tailed learning. Performance gains are most significant under severe data imbalance, highlighting the need for effective de-biasing in challenging scenarios.

Computation efficiency. In Table 6, we show computational costs per-training epoch (in seconds) of different federated learning methods on Fed-LT. The methods include CReFF, FedAvg, and FedAvg+Ours. Our method adds only a minimal time increment of 1.85–2.19 seconds per epoch to FedAvg, resulting in a total time that is far lower than CReFF (3–10 times faster than CReFF across all scenarios). The small and stable time overhead of ours demonstrates its high efficiency, ensuring practicality in federated learning without significantly extending the training time.

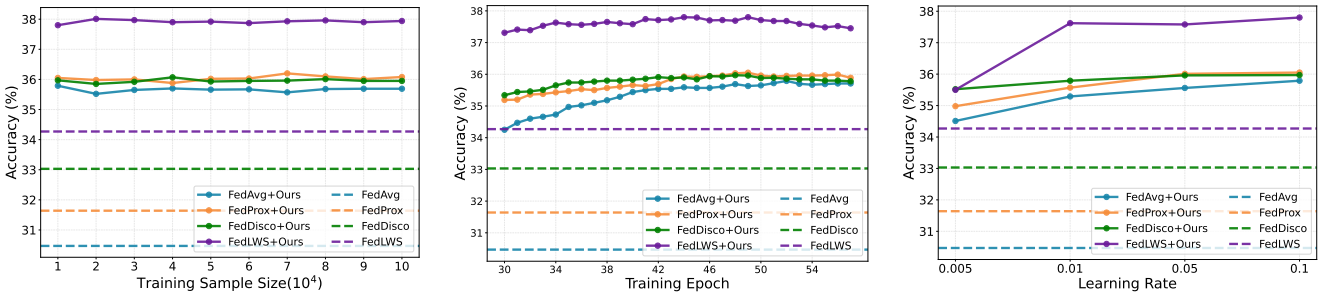
Long-tailed. To further assess the robustness of our approach, we also examine its performance in conventional long-tailed classification tasks. For a fair comparison, we evaluate data-free post-hoc methods, specifically τ-normalized and Post-hoc logit adjustment, alongside three common baselines: ERM, ERM-DRW, and BAMLS. Table 7 shows that our method outperforms both τ-normalized and Post-hoc logit adjustment under different baselines with different imbalanced factors, especially our method achieves a clear performance boost on ERM. It further validates the effectiveness of our approach.

Imbalanced norm postulate. Figures 10 and 11 investigate the relationship between class sample size, classifier weight norm, and test accuracy under Fed-LT. The results validate Assumption 1 (Imbalanced Norm Postulate) by demonstrating strong positive correlations between these three variables across both CIFAR-10-LT and CIFAR-100-LT datasets. Classes with larger sample sizes consistently develop larger classifier weight norms during Fed-LT training. These elevated weight norms correspond directly to higher test accuracy for corresponding classes. This consistent pattern across different data imbalance configurations confirms that classifier norm serves as a reliable indicator of model performance bias in federated learning scenarios. The findings substantiate the critical link between data distribution, learned representation magnitude, and ultimate predictive performance in imbalanced decentralized learning environments.

Existing calibration limitations. We revisit federated-oriented and long-tailed (LT)-oriented classifier calibration method, with key limitations of existing approaches summarized in Table 8. Federated-oriented methods face critical privacy and communication issues: CCVR [27] leaks local data distribution via uploaded feature statistics, while CReFF [35] and CLIP2FL [37] risk privacy

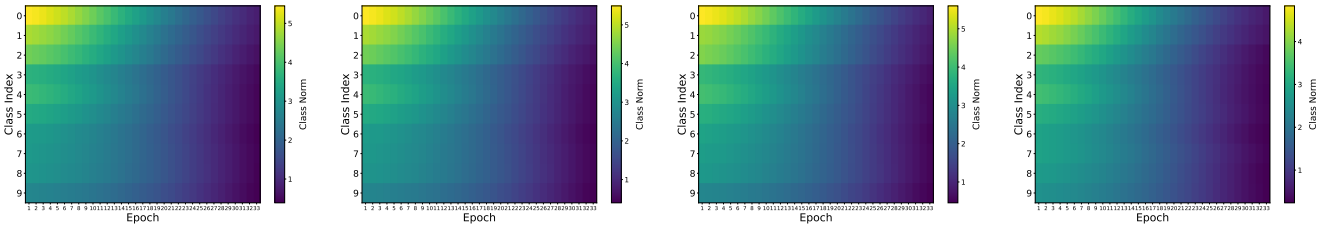


(a) Training sample size M (CIFAR-10). (b) Training epoch E (CIFAR-10). (c) Learning rate η (CIFAR-10).



(d) Training sample size M (CIFAR-100) (e) Training epoch E (CIFAR-100). (f) Learning rate η (CIFAR-100)

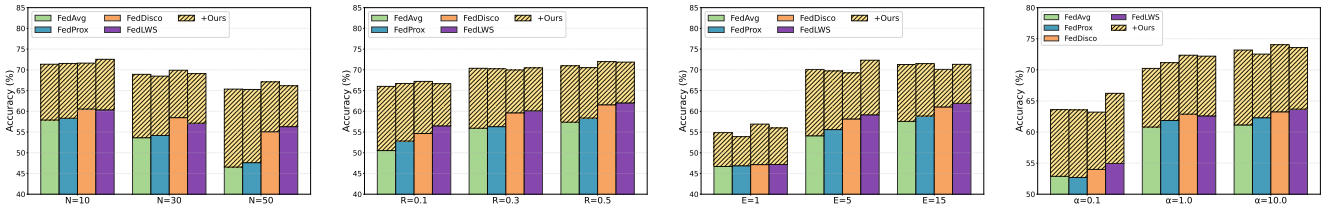
Figure 7. Hyper-parameter discussion under various federated methods. All experiments are conducted on CIFAR-10-LT and CIFAR-100-LT datasets, analyzing the impact of training sample size (M), epoch (E), and learning rate (η) on model performance.



(a) FedAvg+Ours. (b) FedProx+Ours. (c) FedDisco+Ours. (d) FedLWS+Ours.

Figure 8. Classifier weight norm distribution across training epochs under different federated learning methods with ours.

breaches through gradient uploads, all increasing communication overhead. LT-oriented methods suffer from infeasibility or inflexibility: cRT [15], BALMS [32], and LOS [24] depend on unavailable global data, while τ-normalized [15] and post-hoc logit adjustment [29] lack adaptive tuning, relying on static adjustments or fixed offsets tied to global statistics. Additionally, methods like FEDIC [34], MLLM-LLaVa-LF [44], and LWS [15] require auxiliary datasets or multi-modal supervision, conflicting with data privacy and raising computational costs. In contrast, our method avoids all these limitations, satisfying all five evaluated properties for robust and practical Fed-LT calibration.



(a) Client number. (b) Participation ratio. (c) Local Epoch. (d) Non-IIDness α.

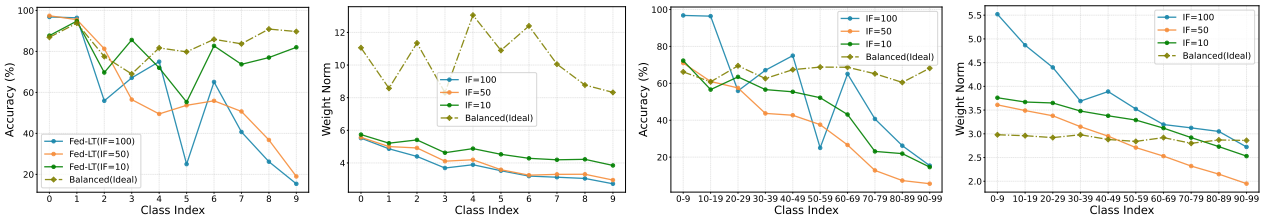
Figure 9. Ablation studies of federated learning methods under different client numbers, participation ratios, local epochs, and Non-IIDness on CIFAR-10-LT.

Table 5. Comparison results of de-bias classifier methods designed for long-tailed tasks on federated long-tailed datasets with $\alpha = 0.5$.

	Method	CIFAR-10-LT			CIFAR-100-LT			Average
		IF=100	IF=50	IF=10	IF=100	IF=50	IF=10	
Client	FedAvg [28]	56.36	59.63	78.05	30.47	36.58	45.93	51.17
	+MaxNorm [1]	47.87	51.18	72.01	25.73	35.32	43.54	46.28$^{-4.89}$
	+τ-normalized [15]	49.95	51.41	72.08	26.22	33.71	43.65	46.17$^{-5.00}$
	+cRT [15]	54.45	55.61	76.60	31.67	36.76	47.01	50.35$^{-0.82}$
	+Logit Adj. [29]	58.88	62.66	75.20	32.24	36.98	46.65	52.10$^{+0.93}$
	+LOS [24]	63.81	69.66	77.53	34.00	38.10	47.92	55.17$^{+4.00}$
	+BALMS [32]	**66.18**	**72.65**	**79.88**	**35.96**	**40.98**	**50.22**	**57.65**$^{+6.48}$
Server	FedAvg [28]	56.36	59.63	78.05	30.47	36.58	45.93	51.17
	+Logit Adj. [29]	58.36	61.59	78.33	31.07	37.22	46.51	52.18$^{+1.01}$
	+τ-normalized [15]	63.42	67.99	78.19	34.38	39.84	48.67	55.41$^{+4.24}$
	+ FedNPC (Ours)	**68.52**	**71.33**	**78.91**	**35.65**	**40.64**	**48.86**	**57.32**$^{+6.15}$
	FedAvg-BALMS [32]	66.18	72.65	79.88	35.96	40.98	50.22	57.65
	+ FedNPC (Ours)	**69.95**	**73.63**	**80.17**	**36.26**	**41.24**	**50.52**	**58.63**$^{+0.98}$

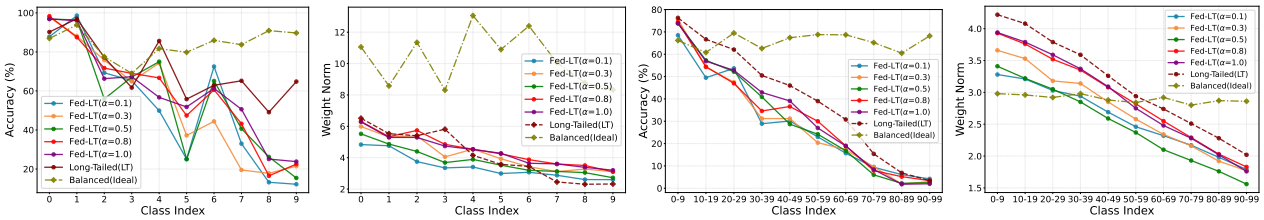
Table 6. Computational costs (sec) per training epoch on Fed-LT.

Method	CIFAR-10-LT		CIFAR-100-LT		Average
	$\alpha = 0.1$	$\alpha = 0.5$	$\alpha = 0.1$	$\alpha = 0.5$	
CReFF	31.62	31.65	117.21	121.73	75.55
FedAvg	10.81	11.60	10.13	12.15	11.17
+Ours	**+1.85**	**+1.88**	**+2.13**	**+2.19**	**+2.01**



(a) Accuracy on CIFAR-10-LT. (b) Norm on CIFAR-10-LT. (c) Accuracy on CIFAR-100-LT. (d) Norm on CIFAR-100-LT.

Figure 10. The relationship between class sample size, classifier weight norm, and test accuracy under Fed-LT on CIFAR-LT under different IFs.



(a) Accuracy on CIFAR-10-LT. (b) Norm on CIFAR-10-LT. (c) Accuracy on CIFAR-100-LT. (d) Norm on CIFAR-100-LT.

Figure 11. The relationship between class sample size, classifier weight norm, and test accuracy under Fed-LT on CIFAR-LT under different α.

Table 7. Comparison results of data-free post-hoc methods on long-tailed datasets with ResNet-32.

Method	CIFAR-10-LT				CIFAR-100-LT				Average
	200	100	50	10	200	100	50	10	
ERM	65.76	70.95	77.69	86.83	34.54	38.65	43.67	56.73	61.80
+τ-normalized	70.22	72.74	79.66	86.83	37.18	41.88	46.24	57.21	64.00
+Post-hoc adj.	72.93	73.90	81.14	87.18	37.83	42.17	46.97	57.44	64.95
+Ours	**73.72**	**74.28**	**81.83**	**88.00**	**38.05**	**42.36**	**46.80**	**57.95**	**65.38**
ERM-DRW	69.83	76.68	81.14	88.19	37.66	41.92	46.36	58.67	65.06
+τ-normalized	72.52	78.12	81.78	88.19	37.66	41.92	46.36	58.67	65.65
+Post-hoc adj.	73.29	79.21	82.14	88.19	38.53	42.52	46.88	58.67	66.18
+Ours	**74.83**	**79.86**	**83.17**	**88.50**	**38.78**	**43.20**	**47.45**	**58.81**	**66.83**
BS	73.31	78.33	82.10	88.60	39.12	41.73	47.20	58.29	66.09
+τ-normalized	76.24	79.37	82.37	88.73	39.12	41.73	47.20	58.29	66.61
+Post-hoc adj.	76.22	80.02	82.17	88.60	39.12	41.73	47.20	58.29	66.67
+Ours	**76.53**	**80.01**	**82.84**	**88.83**	**39.26**	**41.81**	**47.42**	**58.45**	**66.90**

Table 8. Summary of federated-oriented and long-tailed (LT)-oriented classifier calibration methods for the Fed-LT scenario. We evaluate multiple key properties: auxiliary datasets, multi-modal supervision, gradient, statistical, and adaptive tuning. ✓ denotes satisfied properties and ✗ denotes that it is not.

Task	Method	Year	No Auxiliary Dataset	No Multi-modal Supervision	No Gradient Upload	No Statistical Unload	Adaptive Tuning
Federated oriented	CCVR [27]	2021	✓	✓	✓	✗	✓
	CReFF [35]	2022	✓	✓	✗	✓	✓
	FEDIC [34]	2022	✗	✓	✓	✓	✓
	CLIP2FL [37]	2024	✓	✗	✗	✓	✓
	MLLM-LLaVa-LF [44]	2025	✗	✗	✓	✓	✓
Long-tailed oriented	cRT [15]	2019	✗	✓	✓	✓	✓
	τ-normalized [15]	2019	✓	✓	✓	✓	✗
	LWS [15]	2019	✗	✓	✓	✓	✓
	BALMS [32]	2020	✗	✓	✓	✗	✓
	Post-hoc Logit Adj. [29]	2021	✓	✓	✓	✗	✗
	MaxNorm [1]	2022	✗	✓	✓	✓	✓
	LOS [24]	2025	✗	✓	✓	✓	✓
Fed-LT & LT	**Ours**	**2025**	✓	✓	✓	✓	✓