Supplementary Material for NeurIPS Submission

Anonymous NeurIPS Submission

Paper ID 3191

1 Outline

As the supplementary material for our paper titled “Fed-GraB: Federated Long-tailed Learning with Self-Adjusting Gradient Balancer", we provide further details about the **settings of Fed-LT**, **implementation detail**, **additional experiments** as well as the **further analysis**, organized into following sections:

- Sec. 2 introduces detailed settings of Fed-LT, with regard to local/global imbalance levels, categorization methods and the data partition approaches.
    - Sec. 2.1 provides a definition of IF_G, a parameter for long-tailed degree control
    - Sec. 2.2 explained the motivations and the implementation details of the categorization methods based on cumulative frequency.
    - Sec. 2.3 gives the details of dataset partition with addressing the non-IID and long-tailed data distributions.
- Sec. 3 gives detailed implementation of `Fed-GraB`.
- Additional Experiments
    - Sec. 4.1 shows the performance when DPA is combined with other centralized LT-oriented re-weighting methods, and further shows the outperformance of our proposed SGB.
    - Sec. 4.2 explains the effects of the inner parameters of PID algorithm and demonstrates its superior dynamic performance.
    - Sec. 4.3 provides additional ablation study of parameters K_P, K_I, K_D in the self-adjusting controller.
    - Sec. 4.4 gives additional ablation of hyper-parameters ϕ, γ, ζ in mapping function.
    - Sec. 4.5 provides additional results to demonstrates the effectiveness of `Fed-GraB`, e.g., IID CIFAR-10-LT, class-wise performance.
- Additional Analysis
    - Sec. 5.1 discusses the motivation of using static $\Delta_j(t)$.
    - Sec. 5.2 demonstrates that SGB has the capability to be flexibly mounted on a customized tail class for performance boosting.
    - Sec. 5.3 shows the effectiveness of DPA in full-scale category sections.
    - Sec. 5.4 evaluates the performance of different mounting strategies in `Fed-GraB`.
    - Sec. 5.5 demonstrates the the tracking performance of diverse clients.
    - Sec. 5.6 provides the T-SNE visualization results to further demonstrate the effectiveness of `Fed-GraB`.

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

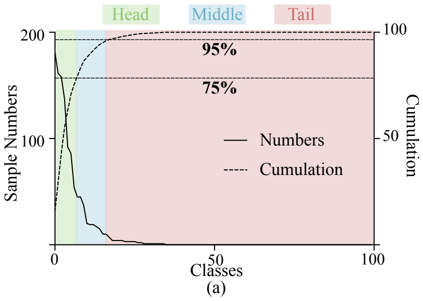


Figure 1: A demonstration of categorization in a cumulative manner. All data are from one client sampled randomly on CIFAR-10-LT, with $\text{IF}_\text{F} = 100$ and $\alpha = 0.1$

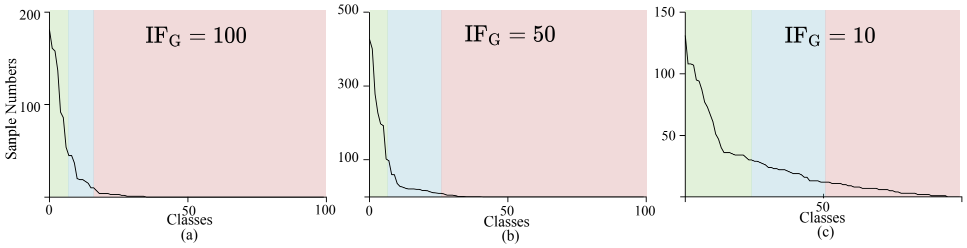


Figure 2: A depiction of our categorization in different imbalanced factors $100, 50$ and 10 on CIFAR-10-LT.

2 Detailed Settings in Fed-LT

2.1 Long-tailed distribution in FL

In this part, we give the definitions of $\text{IF}_\text{L}^{(k)}$ and IF_G to facilitate the formulation of the imbalance level from the global and local perspectives in Fed-LT.

$\mathcal{D}_k$ denotes the local dataset of client k with size $n_k = |\mathcal{D}_k| = \sum_{i=1}^{M} n_k^{(i)}$, where $n_k^{(i)}$ is the number of data samples of class i in client k. To characterize the long-tailed data distribution in Fed-LT, we use the common assumption in long-tailed learning, in which each class is sorted according to the number of images in decreasing order. We shall use the sorted class to obtain class categories (i.e., head, middle, and tail classes). And we measure the imbalance factor in both local and global perspectives, given by

$$\text{IF}_\text{L}^{(k)} = \frac{\max_j\{n_k^{(j)}\}}{\min_s\{n_k^{(s)}\}}, \text{IF}_\text{G} = \frac{\max_j\{\sum_{i=1}^{N} n_i^{(j)}\}}{\min_s\{\sum_{i=1}^{N} n_i^{(s)}\}}, \tag{1}$$

in which $\text{IF}_\text{L}^{(k)}$ represents the imbalance factor at client k and IF_G is global imbalance factor.

In this work, we mainly investigate the scenarios with diverse IF_G and heterogeneous data.

2.2 Class categorization via cumulative frequency in Fed-LT

In FL, the data heterogeneity would result in discrepancies in local data statistics among $\{\mathcal{D}_k\}_{k=1}^{N}$, especially the diverse dataset sizes, across different clients, which brings challenges to categorizing the classes via the number of data samples in each class. Therefore, we adopt the cumulative frequency in class partition criterion to categorize head, middle and tail classes for each client.

The categorization process is as follows. Firstly, all the classes are sorted according to the number of data samples from largest to smallest and then we can obtain the frequency information sequentially after normalization. After the above operations, each class is labeled with its frequency value. Then we set up two thresholds (e.g., 75%, 95%) to partition the classes into head class $j \in \mathcal{J}$, middle class $m \in \mathcal{M}$ and tail class $r \in \mathcal{R}$. We visualize our categorization in Fig. 1.

Conventional class categorization methods in CL are mainly based on the number of data samples in each class. However, in Fed-LT, the head/tail class categorization could be different due to the discrepancy in local data statistics, among $\mathcal{D}_i$ and $\mathcal{D}_j$ (for $i \neq j$). And the imbalanced local dataset sizes and the diverse local imbalance factor $\mathrm{IF}_{\mathrm{L}}^k$ would make the number-based categorization approach inappropriate. For example, some clients may not have head classes in some cases where the maximum number is smaller than the head class threshold. In contrast, the cumulative frequency-based categorization approach is not sensitive, as the categorization is based on the frequency or ratio, rather than an absolute criterion on numbers. The given examples are shown in Fig. 2. The distributions come from randomly sampled clients in our experiments of CIFAR-100-LT, when IF_{G} is 100, 50, and 10 respectively. The pink shade stands for tail classes. The categorization results demonstrate the adopted cumulative frequency-based categorization approach is agnostic to diverse imbalance factors.

2.3 Details of dataset partition

To emulate distributions of Fed-LT (long-tailed at the global level and heterogeneous among clients), we first obtain global long-tailed datasets and then divide the long-tailed dataset into $\mathcal{D} = \{\mathcal{D}_k\}_{k=1}^N$ via dataset partition/sampling (e.g., non-IID or IID data sampling), where $\mathcal{D}_k$ denotes the local dataset of client k with size $|\mathcal{D}_k|$.

For CIFAR-10/100-LT, to shape the original CIFAR-10/100 dataset into long-tailed distribution, the dataset is truncated followed by an exponential distribution class-wisely, and the imbalanced level is controlled by the *global* imbalanced factor IF_{G}. After obtaining the truncated long-tailed dataset, we then conduct sampling to divide $\mathcal{D}$ into $\{\mathcal{D}_k\}_{k=1}^N$. For non-IID data partition, we use Dirichlet distribution-based sampling approach [1], and the identicalness could be controlled by the parameter α, where a smaller α would lead to a more non-IID data distributions. Therefore, followed by this two-step data partition approach, we can obtain the distributions with global long-tailed distribution with data heterogeneity across clients.

For ImageNet-LT, long-tail naturally exists via configuration [2]. Based on the long-tail partition with the number of images per class ranging from 5 to 1280, we further divide the data by Dirichlet Distribution and assign the data samples to clients.

For iNaturalist, the existing iNaturalist-User-120k[3] is designed for federated learning contains balanced samples, and it is not appropriate to be directly used in our experiments.Therefore, a new version is made, namely iNatualist-120k, with 160k examples of 1,023 species classes and partitioned based on iNaturalist-2017 [4]. We randomly sample 1023 items from class-100 to class-2300 sorted from head to tail, from which 30 samples per class are drawn to form the test set, and others belong to the training set with the number of images per class ranging from 15 to 782.

3 Details of implementations

In this section, we provide the details of the controller in SGB and the class activation approach.

The implementation of SGB. The expression of SGB is:

$$u_j(t) = K_p e_j(t) + K_I \int_0^t e_j(t)\mathrm{d}t + K_D \frac{\mathrm{d}e_j(t)}{\mathrm{d}t}, \tag{2}$$

However, in Fed-LT, we compute the re-weighting coefficient $\beta_j(t)$ for each mini-batch, so it is infeasible to execute SGB in this discrete AI system. The model is discretized for the convenience of implementation:

$$u_j(t) = K_P e_j(t) + K_I \sum_{j=0}^{t} e_j(t) + K_D(e_j(t) - e_j(t-1)), \tag{3}$$

In Fed-GraB, we further differentiate Eq. (3):

$$\begin{aligned} \Delta u_j(t) = {} & K_P(e_j(t) - e_j(t-1)) + K_I e_j(t) \\ & + K_D(e_j(t) - 2e_j(t-1) + e_j(t-2)), \end{aligned} \tag{4}$$

The increment of SGB output $\Delta u_j(t)$ is added to last $\Delta u_j(t-1)$ calculate a value in current iteration:

$$u_j(t) = u_j(t-1) + \Delta u_j(t) \tag{5}$$

Then we utilize $u_j(t)$ to generate the re-weighting coefficient β. Note that all experiments are conducted by Eq. (4) for convenience, although Eq. (3) can work as well.

Class Activation Considering an exceptional case, in terms of *debt classifier* (classifier without positive sample), it will receive no positive sample from the training set. The continued influx of negative gradients makes any balance strategy invalid to reach gradient equilibrium. To overcome it, we proposed a Class Activation strategy. SGB does not start to work until it perceives a passing positive gradient. We visualize the trends of *debt classifier* in Fig. 8 (See clients 6, 8, 10 for example.).

Evaluation metrics In our experiments, we compare test accuracy and the required number of communication rounds to measure the generalization and communication efficiency performance, respectively. For the test accuracy, we report the best overall test accuracy for all classes, then we provide the corresponding test accuracy for the head (many-shot), middle (medium-shot), and tail (few-shot) classes separately. Here, we define communication efficiency as the *required communication rounds to reach a target accuracy*. Informally, a smaller number of required rounds indicates better communication efficiency.

Implementation details All experiments were conducted on the PyTorch platform utilizing the NVIDIA GeForce RTX 3090. To introduce diverse levels of data heterogeneity and imbalanced global data, we employed different settings for the parameters α and IF_G. Specifically, in the CIFAR-10-LT dataset, we used $\alpha = 1$ and IF_G values of $10, 50, 100$. In the CIFAR-100-LT dataset, we used $\alpha = 0.5$ and a fixed IF_G value of 100. For the ImageNet-LT and iNaturalist datasets, we used $\alpha = 0.5$, 0.1, and 0.5, respectively. Regarding the local training process, we set the local epoch to 5 for CIFAR-10/100-LT and 1 for ImageNet-LT/iNaturalist-160k. The batch size was set to 10 for CIFAR-10/100-LT and 128 for ImageNet-LT/iNaturalist-160k. We employed the SGD optimizer with a momentum of 0.5, while the learning rate was set to 0.03 for all datasets. The client size varied depending on the dataset, with 40 clients for CIFAR-10-LT and 20 clients for the other datasets. For local training on CIFAR-10/100-LT, we utilized ResNet18 and ResNet34, while ResNet50 was employed for the real-world datasets.

We derived the prior vector by DPA after running 80 rounds with FedAvg and then update the prior vector via DPA before starting local training. We report the test accuracies for all classes and the head/middle/tail classes in our experiments. For the hyper-parameters of Fed-GraB, we use $\gamma = 10, \delta = 9, \zeta = 0.5$ in mapping function, and $K_p = 10, K_I = 0.01, K_D = 0.1$ in all datasets. We train the generic model with $T = 500$ communication rounds via applying SGD optimizer for local training in all experiments unless otherwise stated.

4 Additional Experiments

4.1 Apply DPA to other re-weighting method in Fed-LT

To demonstrate the effectiveness of DPA, we combine it with other distribution-based re-weighting methods. Note that `Focal Loss` and `EQL v2` conduct re-weighting by internal mechanisms, e.g., independent loss function and inner gradients. The global versions of these methods involve collecting clients' information. We introduce `ETF`[5] as a new baseline not covered in the main paper. When training with DPA, the SGB is applied to all classes and performed using a prior derived from DPA. In contrast, local DPA uses the local real distribution as a prior. The results are shown in Tab. 1. To further investigate the scenarios with diverse heterogeneous data, `LDAM` utilizes the global distribution as prior knowledge. In this part, we evaluate the performance of `LDAM+DPA` on CIFAR-10-LT with fixed $\text{IF}_\text{F} = 100$. The results are shown in Tab. 2. These experiments indicate global prior derived by DPA generally improves performance. Nevertheless, the global version of other baselines could not eliminate compensating divergence, leading to worse performance than `Fed-GraB`. In summary, we further demonstrate the effectiveness of DPA and the outperformance of SGB in our proposed `Fed-GraB`.

Method	DPA		Many	Med	Few	All
	w/o	w/				
τ-norm	✓		0.965	0.688	0.577	$0.713_{\pm 0.012}$
		✓	0.962	0.726	0.611	$0.731_{\pm 0.011}$
Eqlv2	✓		0.967	0.731	0.567	$0.714_{\pm 0.011}$
		✓	0.898	0.717	0.586	$0.721_{\pm 0.006}$
GCL	✓		0.954	0.730	0.623	$0.713_{\pm 0.016}$
		✓	0.958	0.722	0.618	$0.730_{\pm 0.003}$
ETF	-	-	0.499	0.624	0.703	$0.631_{\pm 0.001}$
FedIR	-	-	0.976	0.726	0.562	$0.715_{\pm 0.005}$
Fed-GraB	✓		0.953	0.753	0.620	$0.743_{\pm 0.005}$
		✓	0.910	0.698	0.713	$0.761_{\pm 0.008}$

Table 1: Additional results on CIFAR-10 (IF_G=100, $\alpha = 0.5$).

Setting	Method	Many	Med	Few	All
α=0.5	LDAM	0.877	0.583	0.490	0.634
	LDAM+DPA	0.955	0.701	0.602	0.713
	Fed-GraB	0.887	0.675	0.714	0.754
α=1	LDAM	0.891	0.638	0.495	0.657
	LDAM+DPA	0.971	0.730	0.616	0.733
	Fed-GraB	0.910	0.698	0.713	0.768

Table 2: A comparison of test accuracies for LDAM and DPA on CIFAR-10 with imbalance factor 100.

4.2 Explanation and ablation study for PID

The main objective of Fed-GraB is to drive $\Delta(t)$ towards zero, a strategy that effectively counteracts imbalanced training. To achieve this, we utilize a gradient controller, specifically a Proportional-Integral-Derivative (PID) to adjust $\Delta(t)$.

(K_p, K_I, K_D), parameters of PID, are designed for fast settling, steady state error elimination, and oscillation suppression in the control process, which affects the *peak time* (t_p), *steady state error* ($ess(\infty)$), and *overshoot* ($\sigma(\%)$) (**the smaller, the better**), respectively. *Peak time* (t_p) refers to the time it takes for the system response to reach its maximum (or peak) value for the first time. *Steady state error* ($ess(\infty)$) is the difference between the desired final output zero and the actual output of the system $\Delta(t)$ as the time approaches infinity. *Overshoot* ($\sigma(\%)$) is the extent to which the system's response exceeds its final steady-state value. An increase in K_p shortens t_p with overshoot and oscillation, which can be relieved by the item K_D, and K_I is used to decrease the steady-state error. The specific values of these parameters are generally tuned empirically, ensuring that $\Delta(t)$ is rapidly, stably, and precisely confined around zero. The experiments in Tab. 3 show the effect of K_p, K_I, K_D individually via corresponding three indicators.

Indeed, the hyper-parameters of SGB require extra tuning. But once they are set with good performance, the algorithm shall have stable performance across diverse tasks without additional adjustment. Such an insensitive effect could be observed in Tab. 1, 2 in the main paper and Tab. 3, 4 in $\mathcal{SM}$.

Notably, our design of the PID controller focuses on minimizing $\Delta(t)$ to the greatest extent possible, while the overall accuracy of the neural system demonstrates a high level of robustness even in the presence of slight variations in parameters. Consequently, alternative controllers can be employed as substitutes for the PID controller.

4.3 Additional ablation of K_P, K_I, K_D

K_P, K_I, K_D are core coefficients in SGB, which can affect the tracking of $\Delta_j(t)$. We have visualized the trends of parameter sets. To further verify the robustness of SGB, we carry out experiments on CIFAR-10-LT. We fix $\alpha = 1$ and changed IF_G from 10 to 100, and train for 120 rounds. The results are reported in Tab. 4. The sets of $\{K_P, K_I, K_D\}$ are $\mathbf{k}_0 : \{10, 0.01, 0.1\}$, $\mathbf{k}_1 : \{10, 0, 0.3\}$,

K_P	K_I	K_D	t_p	$ess(\infty)$	$\sigma\%$	Accuracy
0	0	0	-	-	-	$0.722_{\pm 0.005}$
3	0	0	∞	2.3363	-	$0.733_{\pm 0.009}$
10	0	0	334	0.6012	-	$0.767_{\pm 0.005}$
10	0	0.1	461	0.6439	-	$0.760_{\pm 0.006}$
10	0.01	0	276	0.2793	1.1954	$0.778_{\pm 0.005}$
10	0.01	0.1	317	0.2651	0.2696	$0.764_{\pm 0.007}$

Table 3: Ablation study of PID on CIFAR-10 (IF$_\text{G}$=100, $\alpha = 0.5$).

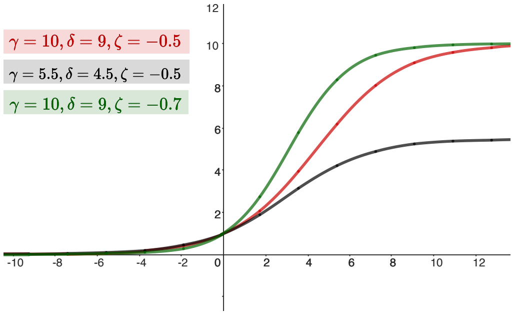


Figure 3: The mapping functions of `SGB` with different parameters.

$\mathbf{k}_2 : \{10, 0.5, 0.1\}$ and $\mathbf{k}_3 : \{5, 0.02, 0.3\}$. The experiment result shows that SGB is parameter-insensitive in Fed-LT for the parameters K_P, K_I, K_D.

4.4 Additional ablation of γ, δ, ζ

The $u_j(t)$ of SGB is supposed to be enlarged and smoothed out by non-linear function $\phi(x) = \frac{\gamma}{1+\delta e^{-\zeta x}}$. After the mapping function, the output is utilized to re-weight gradients. To verify that our model is not sensitive to the parameters of the mapping function, we design 3 sets of γ, δ, ζ: namely $\phi_0 : \{10, 9, -0.5\}$, $\phi_1 : \{5.5, 4.5, -0.5\}$ and $\phi_2 : \{10, 9, -0.7\}$. The mapping curves are shown in Fig. 3. We give the results for the above parameter sets in Tab. 5. Mapping functions can be designed as any other forms (e.g. linear function) in the condition of $\phi(0) = 1$.

Paras	IF$_\text{G}$=100				IF$_\text{G}$=50				IF$_\text{G}$=10			
	Many	Med	Few	All	Many	Med	Few	All	Many	Med	Few	All
$\mathbf{k}_0$	0.969	0.726	0.755	0.786	0.963	0.771	0.823	0.830	0.937	0.787	0.920	0.870
$\mathbf{k}_1$	0.967	0.739	0.753	0.790	0.968	0.772	0.818	0.830	0.940	0.777	0.925	0.869
$\mathbf{k}_2$	0.967	0.719	0.752	0.782	0.968	0.769	0.810	0.826	0.937	0.794	0.923	0.875
$\mathbf{k}_3$	0.967	0.705	0.773	0.785	0.966	0.773	0.815	0.829	0.940	0.777	0.927	0.870

Table 4: Performance comparison of `Fed-GraB` with different K_P, K_I, K_D.

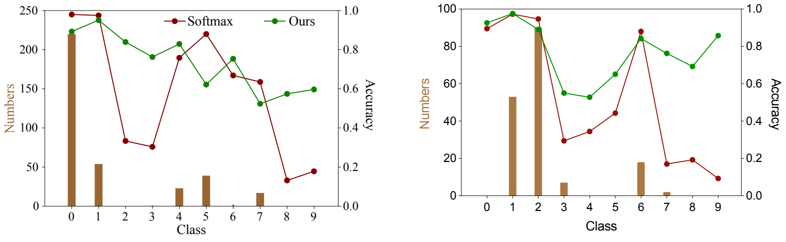
4.5 Further evaluations of the results in the main paper

As reported in the main paper, our experiments are all implemented with non-IID data. To better demonstrate the effectiveness of our proposed `Fed-GraB` on diverse settings. In Tab. 6, we provide comparison results on CIFAR-10-LT with IID setting, where the LT dataset $\mathcal{D}$ will be uniformly sampled into N clients at random. We use the same configuration as the main paper reported.

As supplements to the main paper, we append the class-wise performance before and after local training, and the results are shown in Fig. 4. `Fed-GraB` can effectively eliminate local tailed performance degradation. We additionally compare Fed-GraB (Global-SGB) to the Local-SGB where each client mounts SGB according to local distribution rather than global prior vector derived from DPA with different IF$_\text{G}$. The results on CIFAR-10-LT in Fig. 5 show that Global-SGB outperforms Local-SGB.

Setting	Paras	$\mathrm{IF_G}$=100				$\mathrm{IF_G}$=50			
		Many	Med	Few	All	Many	Med	Few	All
α=1	ϕ_0	0.963	0.685	0.758	0.770	0.946	0.708	0.828	0.804
	ϕ_1	0.938	0.780	0.692	0.777	0.942	0.798	0.743	0.805
	ϕ_2	0.948	0.778	0.676	0.771	0.939	0.794	0.738	0.800
α=0.5	ϕ_0	0.963	0.673	0.739	0.758	0.948	0.709	0.832	0.806
	ϕ_1	0.940	0.767	0.677	0.766	0.934	0.784	0.774	0.810
	ϕ_2	0.936	0.758	0.685	0.765	0.931	0.780	0.759	0.802

Table 5: Performance comparison of `Fed-GraB` with different γ, δ, ζ.



(a)

(b)

Figure 4: Performance comparison on class-wise accuracy before and after local training on CIFAR-10-LT. Each one stands for a randomly selected client.

Setting	Models	$\mathrm{IF_G}$=50				$\mathrm{IF_G}$=100			
		Many	Med	Few	All	Many	Med	Few	All
IID	FedAvg	0.894	0.764	0.661	0.785	0.929	0.720	0.595	0.733
	FedProx	0.899	0.767	0.660	0.788	0.934	0.729	0.602	0.740
	Creff	0.908	0.794	0.668	0.802	0.938	0.734	0.592	0.738
	FedNova	0.904	0.786	0.685	0.803	0.927	0.736	0.625	0.749
	τ-norm	0.899	0.779	0.739	0.815	0.916	0.734	0.652	0.756
	Eqvl2-FL	0.899	0.762	0.669	0.789	0.932	0.734	0.595	0.738
	LDAM-FL	0.887	0.759	0.654	0.779	0.925	0.722	0.555	0.716
	Focal-FL	0.879	0.747	0.610	0.759	0.929	0.710	0.573	0.721
	Fed-GBA	0.883	0.791	0.773	**0.823**	0.921	0.714	0.695	**0.768**

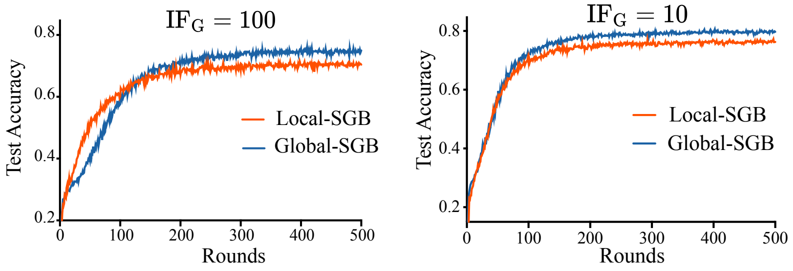
Table 6: Test accuracies of various methods on IID partitioned CIFAR-10-LT with diverse imbalance factors. Both the overall test accuracies and category-wise accuracy are included.

5 Further Analysis

5.1 Static and dynamic $\Delta_j(t)$

In the main paper, we show that the value of static $\Delta_j(t)$ does not matter, as the model will reach to similar performance with a difference $\Delta_j(t)$. Here we raise a question: *whether is there an optimal dynamic $\Delta_j(t)$ in long-tailed learning?* One of the conjectures is that it does contain a natural optimal trend of $\Delta_j(t)$, termed as "best re-weighting", but it has not been understood yet.

However, in the Fed-LT setting, $\Delta_j(t)$ trends of best re-weighting are highly related to numerous systematic parameters, e.g., the degree of long-tail and heterogeneity, class numbers, sampling strategy, client numbers, and size of the dataset. It is unrealistic to design or compute the best trend for each client with diverse data distribution. Even though it is possible to obtain the potential best re-weighting strategy, it is still hard to directly deploy the centralized long-tail re-weighting method to local training due to well-known discrepancies or inconsistencies across local and global sides in Fed-LT. Such challenges motivate us to adopt static $\Delta_j(t)$ for all clients in our proposed `Fed-GraB`.



(a) (b)

Figure 5: A comparison of the test accuracies of two gradient re-balancing methods: Local-SGB and Global-SGB, on CIFAR-10-LT with different imbalanced factors.

5.2 Plug-and-play class-wise performance boosting with SGB

In `Fed-GraB`, SGB is mounted on all classes and performs with the combination of the prior vector instead of mounting on specifically designed tailed classes without global prior. Here, we show that SGB can be flexibly mounted on a customized tail class (i.e., plug-and-play) without global prior for performance boosting. As shown in Fig. 6, though `Fed-GraB` could effectively improve the accuracy on all middle and tailed classes and maintain the performance of head classes. The scheme "Mounted on Class 6" could significantly improve the performance of the tail Class 6.

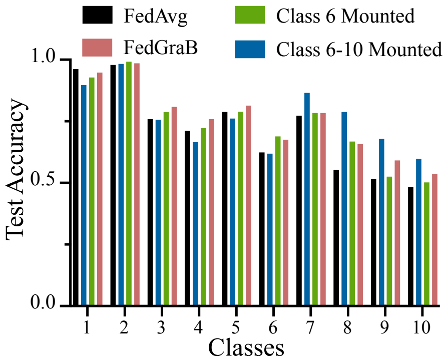


Figure 6: Plug-and-play performance boosting with SGB.

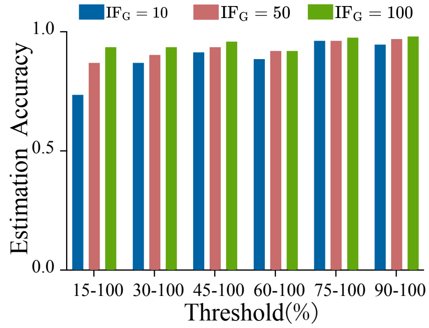


Figure 7: Tail estimation performance with DPA when tail part shifts by a different threshold, the value of which ranges from 15 to 90.

5.3 Full-scale accuracy of DPA

To further demonstrate the effectiveness of DPA, we shift the threshold of **tail categorization** to a full-scale level. We set $\mathrm{IF_F} = 100/50/10$ and $\alpha = 0.5$. After training 80 rounds with `FedAvg`. The weights of global classifiers are used to generate the prior vector via DPA. Then we sort the distribution from large to small as a long-tailed curve. The thresholds are selected in $15\% - 90\%$, as illustrated in Fig. 7. It shows that DPA can achieve good overall performance in all class sections, and reach slightly higher estimation accuracy for extremely long-tailed data.

5.4 The effects of mounting strategy

In the experiments, masking top classes from mounting SGB is necessary for backbone training, because it could learn good representations. Hence, `Fed-GraB` remains top class gradients unchanged for better feature extractor and re-weights the tailed part for better classifiers.

To show the effectiveness of the used "mounting on the tail classes" strategy, we compare mounting SGB on the tail (Fed-GraB-95) and the middle&tail (Fed-GraB-75) in Tab. 7. Based on the pre-trained model by `FedAvg`, we further train for 300 rounds. The results show that the two strategies have similar performance and "mounting on the tail classes" enjoys lower computational costs.

Setting	Method	IF_G=50				IF_G=100			
		Many	Med	Few	All	Many	Med	Few	All
α=1	Fed-GraB-95	0.957	0.809	0.756	0.818	0.960	0.789	0.683	0.781
	Fed-GraB-75	0.962	0.772	0.819	0.829	0.963	0.723	0.758	0.785
α=0.5	Fed-GraB-95	0.946	0.795	0.796	0.826	0.956	0.771	0.680	0.772
	Fed-GraB-75	0.957	0.750	0.835	0.826	0.968	0.700	0.744	0.771

Table 7: Performance evaluation of different SGB mounting strategies

5.5 Tracking of all clients

SGB is an effective gradient balancer with generalization capability, which is not limited by class types or local distributions. We visualize the trend of $\Delta_j(t)$ for head, middle, and tail classes in Fig. 8. It shows that SGB can work well with clients with diverse distributions. Note that *debt classifiers* will be out of control caused by no positive gradients passing by, and we propose a Class Activation strategy (in Sec. 3) to address this issue.

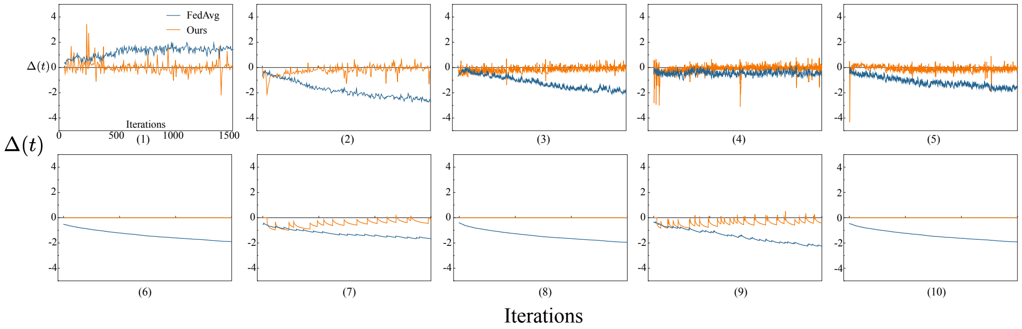


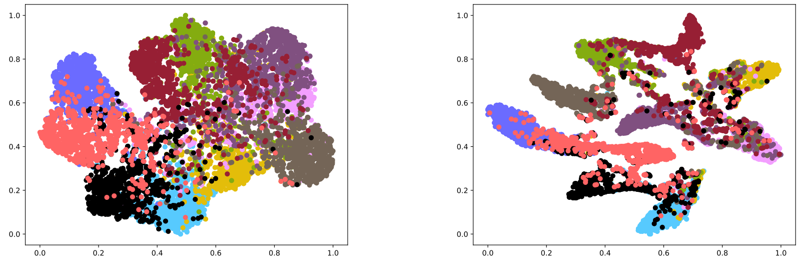
Figure 8: Tracking visualization of 10 selected clients on CIFAR-100-LT, with $\text{IF}_\text{G} = 50$ and $\alpha = 0.5$.

5.6 T-SNE visualization

We additionally visualize model outputs via t-SNE ([6]); See Fig. 9. The mapping points of the uniform category are relatively concentrated. Category clusters are further apart so that it is well distinguished and there is less internal noise (from other categories). It demonstrated that our model achieves high discrimination.

References

[1] T.-M. H. Hsu, H. Qi, and M. Brown, "Measuring the effects of non-identical data distribution for federated visual classification," *arXiv preprint arXiv:1909.06335*, 2019. 3



(a) (b)

Figure 9: T-SNE visualization on CIFAR-10-LT. **Left:** For FedAvg, the samples from different classes are mixed, especially in the central part. **Right:** For Fed-GraB, both intersection area and individual class area decrease.

[2] Z. Liu, Z. Miao, X. Zhan, J. Wang, B. Gong, and S. X. Yu, “Large-scale long-tailed recognition in an open world,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2537–2546, 2019. 3

[3] T.-M. H. Hsu, H. Qi, and M. Brown, “Federated visual classification with real-world data distribution,” in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X*, pp. 76–92, 2020. 3

[4] G. Van Horn, O. Mac Aodha, Y. Song, Y. Cui, C. Sun, A. Shepard, H. Adam, P. Perona, and S. Belongie, “The inaturalist species classification and detection dataset,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8769–8778, 2018. 3

[5] Y. Yang, S. Chen, X. Li, L. Xie, Z. Lin, and D. Tao, “Inducing neural collapse in imbalanced learning: Do we really need a learnable classifier at the end of deep neural network?,” 2022. 4

[6] L. Van der Maaten and G. Hinton, “Visualizing data using t-sne.,” *Journal of machine learning research*, vol. 9, no. 11, 2008. 9