Gradient Reweighting: Towards Imbalanced Class-Incremental Learning

Supplementary Material

Algorithm 1 Gradient Reweighting

```
Input: A new tasks: $\mathcal{T}^t$
Require: CIL model $\mathcal{M}^{t-1}(f_\theta, W)$, learning rate $\eta$
Initialize $\Phi^j, j \in \mathcal{Y}^{1:t}$    ▷ accumulated gradients
for i = 1, 2, ... do    ▷ iteration index
    $\Phi_i^j \leftarrow \Phi_i^j + ||\nabla_{\mathcal{L}_{ce}}(W_i^j)||$
    if $t = 1$ then    ▷ intra-phase gradient reweighting
        $\alpha_i^j \leftarrow \underset{m \in \mathcal{Y}^t}{min} \Phi_i^m / \Phi_i^j$    ▷ class balance ratio
        $W_{i+1}^j \leftarrow W_i^j - \eta \alpha_i^j \nabla_{\mathcal{L}_{ce}}(W_i^j)$    ▷ back prop
    else    ▷ inter-phase decoupled gradient reweightinge
        $\alpha_i^j \leftarrow \begin{cases} \underset{m \in \mathcal{Y}^{1:t-1}}{min} \Phi_i^m / \Phi_i^j & j \in \mathcal{Y}^{1:t-1} \\ \underset{m \in \mathcal{Y}^t}{min} \Phi_i^m / \Phi_i^j & j \in \mathcal{Y}^t \end{cases}$
        $r_{\Phi_i} \leftarrow \frac{\overline{\Phi}_i^{j \in \mathcal{Y}^{1:t-1}}}{\overline{\Phi}_i^{j \in \mathcal{Y}^t}}$    ▷ ratio of mean gradients $\overline{\Phi}$
        $r_i^j \leftarrow \begin{cases} min\{1, \frac{1}{r_{\Phi_i}}\} & j \in \mathcal{Y}^{1:t-1} \\ min\{1, r_{\Phi_i} \times exp(-\gamma \frac{|\mathcal{X}^{1:t-1}|}{|\mathcal{X}^{1:t}|})\} & j \in \mathcal{Y}^t \end{cases}$
    ▷ task balance ratio
        $\beta_i \leftarrow \frac{||\alpha_i r_i \nabla_{\mathcal{L}_{ce}}(W_i)||}{||\nabla_{\mathcal{L}_{dakd}}(W_i)}||$    ▷ loss balance ratio
        $W_{i+1}^j \leftarrow W_i^j - \eta(\alpha_i^j \nabla_{\mathcal{L}_{ce}}(W_i^j) + \beta_i \nabla_{\mathcal{L}_{kd}}(W_i^j))$
    end if
end for
$\mathcal{M}^{t-1} \rightarrow \mathcal{M}^t$    ▷ $\mathcal{T}^t$ finished
```

7. Extended Illustration of Methodology

The objective of our proposed gradient reweighting approach, as detailed in Section 4, is to mitigate bias in the fully connected (FC) layer during the CIL. This goal aligns with the motivations in previous works such as [49, 56]. However, our proposed strategy differs significantly where instead of implementing post-hoc corrections as in [49, 56], we propose to directly adjust the gradient updates during the learning phase, aiming to address the bias issue from the source. Furthermore, our approach demonstrates the flexibility in CIL by effectively addressing both intra-phase and inter-phase imbalances. This sets our method apart from existing strategies that predominantly target inter-phase issues, thus limiting the applicability in real-world scenarios characterized by non-uniform data distributions.

Algorithm 1 illustrates the entire procedure to learn a new task $\mathcal{T}^t$.

7.1. Regularized Softmax Cross-entropy

In Section 4.1, we introduced the regularized softmax cross-entropy to compensate for the side effect caused by gradient reweighting during the learning phase. In this part, we further provide a detailed illustration of where the issue comes from. The main goal of gradient re-weighting is to reduce the effect of imbalanced optimization between head and tail classes by down-weighting the weight updates of head classes. However, while this adjustment helps in emphasizing tail classes, it also inadvertently leads to increased loss values for the head classes, resulting in larger gradients. Specifically, the gradient of a head class input data $(\mathbf{x}_k, y_k)$ with respect to each output logit z_j can be calculated as

$$\frac{\partial \mathcal{L}_{ce}}{\partial z_j} = \begin{cases} p_j - 1 & j = y_k \\ p_j & j \neq y_k \end{cases}, p_j = \frac{exp(z_j)}{\sum_{m=1}^{|\mathcal{Y}^1|} exp(z_m)} \tag{13}$$

where z_j is the jth output logit and p_j is the corresponding softmax output. Attributing to the down-weight of gradients to the head class y_k, the output logit z_{y_k} and its softmax p_{y_k} decreases during the training process. Consequently, it results in an increase of the positive gradient norm $||1 - p_j||$, prompting the head class weight W^{y_k} to produce a higher output logit value. Concurrently, since the total sum of softmax outputs is constrained to 1, the decrease of p_{y_k} leads to the rise of the negative gradient norm $\sum_{j \neq y_k} ||p_j||$, driving the tail classes weight to output even lower scores. Our method utilizes regularized softmax as in Equation 5, which effectively mitigates this side effect by adding a per-class offset π_j to the output logit

$$\pi_j = \begin{cases} \frac{n_j}{\sum_m n_m} & t = 1 \\ \frac{min\{n_j, n_\varepsilon\}}{\sum_m min\{n_m, n_\varepsilon\}} & t > 1 \end{cases} \tag{14}$$

where t is the task index, n_j is the number of training data for class j, and n_ε denotes the exemplar budget per class. Thus, the instance-rich classes have larger π_j with an increase of softmax output p_j to compensate for the side effect of down-weighting the gradients during the training process.

7.2. Imbalanced Catastrophic Forgetting

As described in Section 1, we argue catastrophic forgetting could also be imbalanced. In this part, we provide further illustration and also present experimental results to visualize this issue. In CIL, the forgetting problem mainly comes from the unavailability of old classes' training data during the learning of new classes. However, as the training data is severely imbalanced, there is a significant variance in the amount of lost training data between head and tail classes due to a fixed memory budget. During task t, suppose we select n_e exemplars per class, thus we have $|\mathcal{X}_e^j| \leq n_e, \forall j \in \mathcal{Y}^{1:t-1}$ (note that some classes may contain less than n_e training data). Given the class imbalance

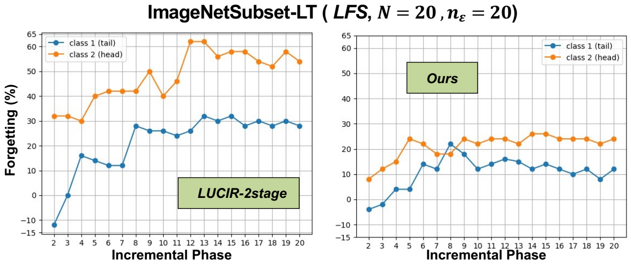

Figure 7. The forgetting rate (%) for one selected head class and one tail class by comparing the LUCIR-2stage [22, 28] and our method.

condition where $n_j \gg n_k$ for a head class j and a tail class k, it implies that a large volume of instances from head class j become unavailable in comparison to tail class k in the subsequent incremental learning phases $n_j - n_e \gg n_k - n_e$. (*e.g.* consider the case where a tail class k has training data $n_k < n_e$, then all of the training data for class k will be preserved for entire CIL.) This poses a unique challenge in CIL of imbalanced forgetting where head classes (with more training data lost) potentially suffer more performance degradation than tail classes (with less or even no training data lost).

To visualize the imbalance forgetting issue, we perform CIL on ImageNetSubset-LT following the implementation details as described in Section 5.1. Specifically, we select a tail class (class 1: 33 training images) and a head class (class 2: 1017 training images) from the initial task 1 and measure the forgetting rate $Forg = Acc_1 - Acc_i$ in each subsequent learning phase $i > 1$. Acc is the accuracy on test data belonging to that specific class (*i.e.* class 1 or class 2) where Acc_i denotes incremental phase i and Acc_1 is the accuracy in the initial phase when the class is firstly observed. The result is shown in Figure 7. We observe the head class suffers a higher performance degradation compared to the tail class in LUCIR-2stage [22, 28], showing the imbalanced catastrophic forgetting phenomenon. Our method, with gradient reweighting and Distribution-Aware Knowledge distillation (DAKD) loss as illustrated in Section 4, significantly reduces the forgetting for both tail and head classes while mitigating the imbalance issue. Specifically, the DAKD effectively addresses this problem by decoupling the original knowledge distillation loss [21] into a weighted sum of two components using a ratio σ measured by entropy on the lost training data distribution $\mathbf{s}$ with total classes c as

$$\sigma(\mathbf{s}) = \frac{-\sum_{j=1}^{c} v_j log(v_j)}{log(c)} \qquad v_j = \frac{s_j}{\sum_{m=1}^{c} s_m} \tag{15}$$

Therefore, when $\sigma = 1$ (balanced data loss), our DAKD works equivalently as original knowledge distillation loss. When σ decreases, it prioritizes knowledge retention from classes with greater data loss. Overall, the DAKD focuses more on preserving the performance of head classes with a greater loss of training data as well as allowing more plasticity for tail classes, enabling them to adapt more effectively to the current training data distribution.

8. Detailed Experimental Setup

8.1. Evaluation Metrics and Training Details

In this part, we first illustrate the evaluation metrics including the average accuracy (ACC) and forgetting rate as used in Section 5. Then we provide additional training details.

Evaluation Metrics: The average accuracy (ACC) [31] considers the performance of all incremental learning phases as

$$\bar{\mathcal{A}} = \frac{1}{N} \sum_{t=1}^{N} \mathcal{A}_t$$

where $\mathcal{A}_t$ is the top-1 classification accuracy after learning task $\mathcal{T}^t$ on all classes seen so far. The forgetting rate [31], also known as backward transfer (BWT), measures the performance drops during CIL as calculated in

$$\bar{\mathcal{F}} = \frac{-1}{N-1} \sum_{t=1}^{N-1} \mathcal{A}_N^t - \mathcal{A}_t^t$$

where $\mathcal{A}_N^t$ refers to the classification accuracy on task t after learning task N. In general, an expected CIL model should have higher average accuracy $\bar{\mathcal{A}} \uparrow$ as well as lower forgetting rate $\bar{\mathcal{F}} \downarrow$.

Training Details: Our method is implemented with PyTorch [38] based on the framework provided in [28, 33]. Each experiment is run on a single NVIDIA A40 GPU with 48G memory. The class order is generated and shuffled using the identical random seed (1993) as in [28, 40]. Only regular data augmentation technique is included such as random flip and crop. (No AutoAugment [13] as implemented in [11, 47]). To further ensure fair comparisons with existing work, instead of obtaining the results from the original publications, we reproduce the existing methods under the same setting three times for each experiment and report the average performance as illustrated in Section 5.1.

8.2. Exemplar Selection

In this part, we illustrate the exemplar setup used in Section 5. In conventional CIL, there are two widely used strategies for storing exemplars including (i) Fixed Memory (FM), and (ii) Growing Memory (GM). Specifically, the FM uses a fixed buffer size $|\mathcal{E}| = \mathcal{B}$ and split evenly for all classes seen so far thus each class contains $n_\varepsilon = \frac{\mathcal{B}}{c}$ exemplars. As more classes are encountered, n_ε will decrease. On the other hand, GM uses a fixed n_ε per class, thus the total exemplar size $|\mathcal{E}| = c \times n_\varepsilon$ is growing when more classes

Datasets	**CIFAR100-LT**				**ImageNetSubset-LT**			
Evaluation protocol	*LFS*		*LFH*		*LFS*		*LFH*	
Total tasks N	10	20	5	10	10	20	5	10
iCaRL [40]	38.71	34.66	30.13	29.98	50.10	43.22	45.28	43.98
IL2M [5]	44.42	40.54	39.83	37.87	47.53	40.02	46.95	44.34
BiC [49]	41.59	36.41	34.57	31.08	47.92	45.53	47.78	41.05
WA [56]	43.69	37.58	35.66	33.02	48.83	46.71	48.29	42.24
SSIL [1]	43.25	35.28	33.90	23.16	50.62	41.41	40.00	41.73
FOSTER [47]	43.68	36.70	38.43	35.19	49.72	42.68	47.31	46.89
MAFDRC [11]	44.27	37.82	42.10	41.94	50.83	44.20	48.69	47.11
EEIL-2stage [10, 28]	45.34	**41.03**	39.95	38.85	50.39	43.57	50.93	48.37
LUCIR-2stage [22, 28]	47.83	36.01	**45.10**	43.35	**53.47**	47.67	**54.88**	53.38
PODNet-2stage [15, 28]	**48.67**	34.17	44.42	43.54	52.00	44.55	54.75	**54.21**
FOSTER-2stage [28, 47]	46.35	38.93	43.21	**44.18**	52.64	**47.91**	54.26	53.87
Ours	**50.24**	**41.50**	**44.87**	**44.13**	**55.42**	**49.73**	**55.67**	**53.95**

Table 3. Results of average accuracy (%) for Ordered CIL on CIFAR100-LT, ImageNetSubset-LT with imbalance factor $\rho = 100$, memory budget $n_\varepsilon = 20$ evaluated under Learning From Scratch (*LFS*) and Learning From Half (*LFH*). **Best** and **Second Best** results are marked.

c are observed. Though FM is more practical and popular in conventional CIL with balanced training data distribution, it poses two non-trivial questions in imbalanced CIL including (a) how to allocate the fixed memory size $\mathcal{B}$ to class-imbalanced data distribution, and (b) how to update the memory buffer after observing new classes. Until now, the FM-based exemplar selection is still under-explored in imbalanced CIL (*i.e.* There lacks efficient exemplar selection strategies.)

In this work, we primarily follow the setup in [28] to use GM with fixed n_ε. Specifically, we select up to n_ε samples per class after each incremental learning phase by applying Herding algorithm [48] based on the class mean. Note that for tail classes with the number of training data less than n_ε, we store all of their training data in the exemplar set. Therefore, the exemplar set $\mathcal{E}$ could still exhibit the class-imbalanced issue.

Later in Section 9.1, we also explore a variant of FM setting by using dynamic n_ε in imbalanced CIL. Specifically, we set a fixed memory buffer size $\mathcal{B}$ and calculate the $n_\varepsilon = \frac{\mathcal{B}}{c}$ after learning each task. Note that this case is still different from the conventional FM setting as most classes in long-tailed distribution will have fewer training samples than n_ε. However, we can ensure the total buffer size is bounded with $|\mathcal{E}| \leq \mathcal{B}$.

8.3. The 2-Stage Implementation

In this part, we illustrate the implementation of the 2-stage module [28] to integrate with existing conventional CIL for experiments in Section 5. As proposed in [28], the 2-stage framework is structured as follows: stage-1 focuses on training the feature extractor and classifier using the entire dataset (aligned with conventional CIL training with no alternation). Subsequently, stage-2 involves the training of an additional Learnable Weight Scaling (LWS) layer using a class-balanced sampler.

Therefore, in our implementation of the 2-stage module, each incremental phase is conducted in two steps. Initially, we perform the original method with implementation details outlined in Section 5.1. Following this, an additional training phase is introduced specifically to learn the Learnable Weight Scaling (LWS). During this phase, we fix parameters in the feature extractor and the classifier corresponding to the previously learned classes. The implementation details of this training phase follow the [28], involving a fixed learning rate of 0.1 and 30 training epochs. Note that in Table 1, we did not include the results for the original LUCIR [22], EEIL [10], and PODNet [15] as their 2-stage versions have been proved to be more effective for imbalanced CIL in [28]. However, we further performed the 2-stage module on FOSTER [47] and compared it with the original results as shown in Table 1.

9. Additional Experimental Results

In this part, we first present additional experimental results under imbalanced CIL and then show the effectiveness of our proposed gradient reweighting method even in conventional CIL with balanced data distribution. All the experiment setting follows the same implementation setups in Section 5.1.

9.1. Results for Imbalanced CIL

Results on ImageNet-LT with 1,000 classes: We evaluate our method on large-scale datasets by constructing ImageNet-LT with 1,000 classes from ImageNet [42] using imbalance factor $\rho = 100$. The experimental results are shown in Figure 9. Notably, even in the context of

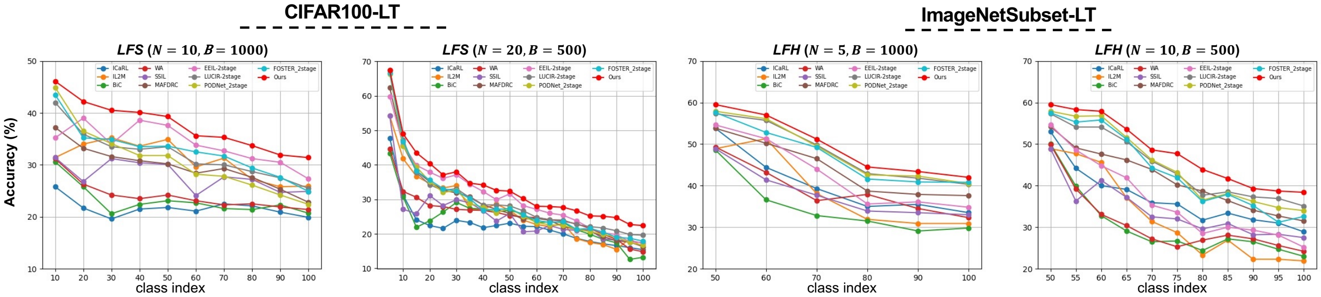

Figure 8. The classification accuracy (%) on test data belonging to all classes seen so far at each incremental step by using the fixed memory budget $\mathcal{B} \in \{500, 1000\}$ on CIFAR100-LT and ImageNetSubset-LT with imbalance factor $\rho = 100$.

Hyper-parameter γ	0	0.5	1.0	1.5	2.0	2.5	3.0	4.0	5.0	10.0
CIFAR100-LT (*LFH*, $N = 10$)	37.15	38.61	39.11	**39.54**	39.27	38.36	37.83	37.40	37.43	37.45
ImageNetSubset-LT (*LFS*, $N = 20$)	39.87	40.63	40.79	40.87	41.03	41.14	**41.23**	40.36	40.07	38.18

Table 4. Results of average accuracy of our proposed method by tuning hyper-parameter $\gamma \in [0, 10]$ with fixed $\lambda_b = 1$.

Hyper-parameter λ_b	0.1	0.5	1.0	2.0	3.0	4.0	5.0
CIFAR100-LT (*LFH*, $N = 10$)	37.54	**39.18**	39.11	32.79	27.59	25.25	21.82
ImageNetSubset-LT (*LFS*, $N = 20$)	40.37	**41.06**	40.79	37.12	35.00	32.21	28.47

Table 5. Results of average accuracy of our proposed method by tuning hyper-parameter $\lambda_b \in [0.1, 5]$ with fixed $\gamma = 1$.

Datasets	**CIFAR100**				**ImageNet-Subset**			
Evaluation protocol	*LFS*		*LFH*		*LFS*		*LFH*	
Total tasks N	20		10		20		10	
Exemplar Setup	GM	FM	GM	FM	GM	FM	GM	FM
iCaRL [40]	42.23	52.88	47.69	47.11	50.48	56.28	60.85	61.64
EEIL [10]	48.39	59.95	51.65	54.35	43.20	54.20	53.05	56.75
IL2M [5]	48.93	59.12	51.49	54.75	44.23	53.31	51.47	55.06
BiC [49]	52.12	57.47	33.56	48.55	51.68	61.27	58.57	62.32
WA [56]	51.28	56.73	35.62	49.13	49.87	62.94	56.95	61.82
SSIL [1]	50.84	57.66	43.52	47.58	48.74	58.29	59.30	59.35
LUCIR [22]	49.09	58.29	59.25	59.37	46.88	56.01	62.56	64.18
PODNet [15]	45.45	53.92	**60.50**	61.66	38.33	49.28	61.41	63.99
FOSTER [47]	51.90	63.37	**59.54**	**67.02**	**56.79**	69.42	**63.82**	**66.25**
MAFDRC [11]	**52.83**	**65.68**	58.44	66.21	54.63	**70.18**	62.13	65.47
Ours	**54.30**	**64.03**	59.31	**68.70**	**59.32**	**69.56**	**67.32**	**67.20**

Table 6. Results of average accuracy for conventional CIL on original CIFAR100 and ImageNet-Subset with fixed memory (FM) budget $\mathcal{B} = 2,000$ and growing memory (GM) budget $n_\varepsilon = 20$. **Best** and **Second Best** results are marked.

this extensive dataset, our method outperformed existing approaches at each incremental learning phase, demonstrating its efficacy in handling large-scale data in the real world.

Results with Ordered Long-Tailed CIL: In accordance with [28], we implemented Ordered Long-Tailed CIL where the learning process begins with the most frequent classes (with most training samples) and progresses towards the least frequent ones (with least training samples). This scenario is aligned with many realistic applications where learning typically starts with available common classes and gradually shifts to more challenging samples. The results on CIFAR100-LT and ImageNetSubset-LT with imbalance factor $\rho = 100$ are summarized in Table 3. We observed the results in ordered cases are typically better than the results in shuffled cases as shown in Table 1, which can be attributed to the significant reduction of intra-class imbal-

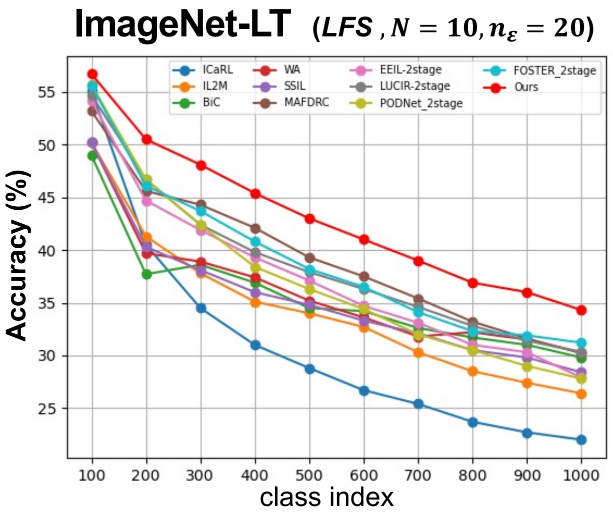

Figure 9. The classification accuracy (%) on test data belonging to all classes seen so far at each incremental step on ImageNet-LT with imbalance factor $\rho = 100$.

ance issue in this scenario. Despite this variation in learning conditions, our method consistently demonstrated promising results, outperforming existing approaches in both ordered and shuffled long-tailed CIL without requiring the additional training stage.

Results with Fixed Memory Budget: As described in Section 8.2, we consider a variant of fixed memory budget $\mathcal{B}$. The results on CIFAR100-LT and ImageNetSubset-LT with imbalance factor $\rho = 100$ and $\mathcal{B} \in \{500, 1000\}$ are visualized in Figure 8. Together with the results shown in Figure 4, we demonstrate the adaptability of our method under both exemplar setups to achieve the best performance at each incremental learning phase. However, as illustrated in Section 8.2, while the total buffer size $\mathcal{B}$ is bounded, the imbalanced CIL under a fixed memory budget usually introduces a more pronounced class imbalance issue within the exemplar set. Addressing this challenge remains a crucial area for future algorithm development.

Tuning Hyparameters: As illustrated in Section 4, we introduced two hyper-parameters in this work including (i) γ to control the magnitude of attenuation factor as in Equation 10, and (ii) λ_b to control the influence of knowledge distillation in the integrated objective as in Equation 9. As detailed in the experimental setup in Section 5.1, we use $\gamma = 1$ and $\lambda_b = 1$ for simplicity on all experiments to show the effectiveness of our method even without hyperparameter tuning. In this part, we demonstrate that tuning these two hyper-parameters can achieve better performance. The results by tuning γ and λ_b are summarized in Table 4 and Table 5, respectively. For γ, we observed that a moderate increment from $\gamma = 0$ gradually increases performance while a large γ results in performance degradation. This observation is aligned with our findings as in Section 4.2 where the model could under-fit on new classes without γ as the new classes receive less attention, and using large γ will conversely result in prediction bias towards new classes due to the inter-class imbalance issue in CIL. Similarly, as λ_b becomes larger, there is a sharp decrease in accuracy since the overly dominant knowledge distillation component in the integrated objective function can obstruct the effective learning of new classes. These observations highlight the adaptability of our method to achieve potential improvements to accommodate various applications in the real world.

9.2. Results for Conventional CIL

In this part, we highlight the effectiveness of our proposed method even under the Conventional CIL setting with class-balanced data distribution, where only the inter-phase imbalance issue is present. The results on CIFAR-100 and ImageNetSubset are summarized in Table 6 where we consider both growing memory (GM) with $n_\varepsilon = 20$ and fixed memory (FM) with $\mathcal{B} = 2,000$. Our method achieved promising performance across both datasets and memory setups. Notably, in the ImageNetSubset evaluations, our method significantly outperformed existing approaches under the GM setup, where the inter-phase imbalance issue presents a more substantial challenge. Additionally, we achieved promising performance under FM setup on CIFAR-100 and ImageNetSubset datasets under both evaluation protocols with varied incremental phases. These results further demonstrate the adaptability and effectiveness of our proposed method in a broader CIL context with various learning conditions.

9.3. Long-tailed Recognition

In this section, we evaluate our gredient reweighting for its efficacy in solving the imbalanced image classification beyond CIL by conducting comparative analyses with established methods in long-tailed recognition. We denote our regularized softmax output illustrated in Section 4.1 as RS. Specifically, we train a ResNet-32 to classify the 100 classes in CIFAR100-LT with various imbalance factor $\rho \in \{10, 50, 100\}$ following the training protocol as in [14]. The results are summarized in Table 7. Our method shows competitive performance even without the use of regularized softmax output. Upon integrating the regularized softmax output, which further balances the learning process, we consistently achieve improved classification accuracy and outperform existing methods.

9.4. Computation and Memory Efficiency

Our framework employs a compact vector, denoted as Φ, for the computation of accumulated gradients. To elucidate the

	CIFAR100-LT		
	$\rho = 100$	$\rho = 50$	$\rho = 10$
ROS [46]	36.32	41.28	55.12
Focal Loss [27]	38.91	43.26	55.08
LDAM [9]	40.82	45.68	57.32
CB Loss [14]	39.62	46.29	57.29
IB Loss [36]	43.62	46.80	58.01
BS Loss [41]	44.12	49.25	59.38
EQL v2 [45]	43.81	48.25	57.06
CMO [37]	43.54	47.92	58.97
Ours *w/o* RS	43.84	47.39	57.95
Ours	**45.27**	**49.02**	**60.71**

Table 7. Long-tailed recognition accuracy (%) on CIFAR100-LT with imbalance factor $\rho \in \{100, 50, 10\}$

storage usage, we consider the following detailed analysis. Assuming Φ comprises N elements, where each element is a floating-point number represented using 32 bits (or 4 bytes) of memory, the total storage requirement for Φ can be quantified using the equation:

$$\text{Storage}_{\Phi} = N \times 4(\text{bytes}) \tag{16}$$

For CIL, N refers to the number of classes seen so far. Consequently, for datasets with 100 classes such as CIFAR100 or ImageNetSubset, $N = 100$ at the last incremental learning phase, rendering the storage used by Φ as follows:

$$\text{Storage}_{\Phi} = 100 \times 4 = 400(\text{bytes}) \tag{17}$$

To provide a comparative perspective on this storage efficiency, we consider the storage required by a single RGB image from the CIFAR dataset. Each CIFAR image, with a resolution of 32×32 pixels and utilizing 8 bits (or 1 byte) per pixel for each of the three RGB color channels, necessitates the following storage

$$\text{Storage}_{\text{CIFAR}} = 32 \times 32 \times 3 = 3072(\text{bytes}) \tag{18}$$

The ratio of storage usage can be calculated as

$$\frac{\text{Storage}\Phi}{\text{StorageCIFAR}} = \frac{400}{3072} \approx 0.13 \tag{19}$$

Thus the total usage of storage for calculating Φ will be around $1/10$ of one single CIFAR image, while it brings significant performance improvements as shown in our experiments.

In terms of computational efficiency, our framework is designed for end-to-end training. This integrated approach sidesteps the substantial computational overhead typically associated with decoupled training phases. For instance, existing methods such as [28] require 30 additional epochs for balanced fine-tuning. Our experimental results underscore the substantial performance enhancements achieved through our methodology. These improvements are not merely computational but also extend to the accuracy and efficiency of the training process, showing great potential for facilitating high-effiency models in real-world applications.

References

[1] Hongjoon Ahn, Jihwan Kwak, Subin Lim, Hyeonsu Bang, Hyojun Kim, and Taesup Moon. Ss-il: Separated softmax for incremental learning. *Proceedings of the IEEE/CVF International conference on computer vision*, pages 844–853, 2021. 2, 5, 7, 3, 4

[2] Rahaf Aljundi, Eugene Belilovsky, Tinne Tuytelaars, Laurent Charlin, Massimo Caccia, Min Lin, and Lucas Page-Caccia. Online continual learning with maximal interfered retrieval. *Advances in Neural Information Processing Systems*, 32, 2019. 2

[3] Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. *Advances in Neural Information Processing Systems*, 32, 2019. 2

[4] Donghyeon Baek, Youngmin Oh, Sanghoon Lee, Junghyup Lee, and Bumsub Ham. Decomposed knowledge distillation for class-incremental semantic segmentation. *Advances in Neural Information Processing Systems*, 35:10380–10392, 2022. 8

[5] Eden Belouadah and Adrian Popescu. Il2m: Class incremental learning with dual memory. *Proceedings of the IEEE International Conference on Computer Vision*, pages 583–592, 2019. 2, 7, 3, 4

[6] Eden Belouadah, Adrian Popescu, Umang Aggarwal, and Léo Saci. Active class incremental learning for imbalanced datasets. *European Conference on Computer Vision*, pages 146–162, 2020. 2

[7] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. *Proceedings of the European Conference on Computer Vision*, 2014. 6

[8] Mateusz Buda, Atsuto Maki, and Maciej A Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural networks*, 106:249–259, 2018. 3

[9] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in neural information processing systems*, 32, 2019. 3, 6

[10] Francisco M. Castro, Manuel J. Marin-Jimenez, Nicolas Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. *Proceedings of the European Conference on Computer Vision*, 2018. 2, 7, 3, 4

[11] Xiuwei Chen and Xiaobin Chang. Dynamic residual classifier for class incremental learning. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18743–18752, 2023. 1, 2, 7, 3, 4

[12] Aristotelis Chrysakis and Marie-Francine Moens. Online continual learning from imbalanced data. *International Conference on Machine Learning*, pages 1952–1961, 2020. 2

[13] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 113–123, 2019. 2

[14] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9268–9277, 2019. 3, 6, 5

[15] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. *Proceedings of the European Conference on Computer Vision*, pages 86–102, 2020. 2, 6, 7, 3, 4

[16] Emanuele Francazi, Marco Baity-Jesi, and Aurelien Lucchi. A theoretical analysis of the learning dynamics under class imbalance. *International Conference on Machine Learning*, pages 10285–10322, 2023. 3

[17] Yiduo Guo, Bing Liu, and Dongyan Zhao. Dealing with cross-task class discrimination in online continual learning. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11878–11887, 2023. 1, 2, 3

[18] Tyler L. Hayes and Christopher Kanan. Online continual learning for embedded devices. *Conference on Lifelong Learning Agents*, 2022. 1

[19] Jiangpeng He and Fengqing Zhu. Online continual learning via candidates voting. *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 3154–3163, 2022. 2

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 6

[21] Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *Proceedings of the NIPS Deep Learning and Representation Learning Workshop*, 2015. 2, 5

[22] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 831–839, 2019. 2, 6, 7, 3, 4

[23] Ying Jin, Jiaqi Wang, and Dahua Lin. Multi-level logit distillation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24276–24285, 2023. 8

[24] Chris Dongjoo Kim, Jinseo Jeong, and Gunhee Kim. Imbalanced continual learning with partitioning reservoir sampling. *European Conference on Computer Vision*, pages 411–428, 2020. 2

[25] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Technical Report*, 2009. 6

[26] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, 2017. 2

[27] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 6

[28] Xialei Liu, Yu-Song Hu, Xu-Sheng Cao, Andrew D Bagdanov, Ke Li, and Ming-Ming Cheng. Long-tailed class incremental learning. *European Conference on Computer Vision*, pages 495–512, 2022. 2, 4, 6, 7, 3

[29] Yaoyao Liu, Yuting Su, An-An Liu, Bernt Schiele, and Qianru Sun. Mnemonics training: Multi-class incremental learning without forgetting. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12245–12254, 2020. 2

[30] Yaoyao Liu, Bernt Schiele, and Qianru Sun. Adaptive aggregation networks for class-incremental learning. *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 2544–2553, 2021. 2

[31] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, pages 6467–6476, 2017. 1, 2, 7, 8

[32] Davide Maltoni and Vincenzo Lomonaco. Continuous learning in single-incremental-task scenarios. *Neural Networks*, 116:56–73, 2019. 2

[33] Marc Masana, Xialei Liu, Bartłomiej Twardowski, Mikel Menta, Andrew D Bagdanov, and Joost Van De Weijer. Class-incremental learning: survey and performance evaluation on image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):5513–5533, 2022. 7, 2

[34] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. pages 109–165. Elsevier, 1989. 1

[35] Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. *International Conference on Learning Representations*, 2021. 3, 5

[36] Seulki Park, Jongin Lim, Younghan Jeon, and Jin Young Choi. Influence-balanced loss for imbalanced visual classification. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 735–744, 2021. 3, 6

[37] Seulki Park, Youngkyu Hong, Byeongho Heo, Sangdoo Yun, and Jin Young Choi. The majority can help the minority: Context-rich minority oversampling for long-tailed classification. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6887–6896, 2022. 3, 6

[38] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. *Proceedings of the Advances Neural Information Processing Systems Workshop*, 2017. 2

[39] Ameya Prabhu, Philip HS Torr, and Puneet K Dokania. Gdumb: A simple approach that questions our progress in continual learning. *Proceedings of the European Conference on Computer Vision*, pages 524–540, 2020. 2

[40] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. iCaRL: Incremental classifier and representation learning. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 1, 2, 6, 7, 3, 4

[41] Jiawei Ren, Cunjun Yu, Xiao Ma, Haiyu Zhao, Shuai Yi, et al. Balanced meta-softmax for long-tailed visual recognition. *Advances in neural information processing systems*, 33:4175–4186, 2020. 3, 6

[42] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3): 211–252, 2015. 6, 3

[43] Christian Simon, Piotr Koniusz, and Mehrtash Harandi. On learning the geodesic path for incremental learning. *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 1591–1600, 2021. 2

[44] Jingru Tan, Changbao Wang, Buyu Li, Quanquan Li, Wanli Ouyang, Changqing Yin, and Junjie Yan. Equalization loss for long-tailed object recognition. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11662–11671, 2020. 3

[45] Jingru Tan, Xin Lu, Gang Zhang, Changqing Yin, and Quanquan Li. Equalization loss v2: A new gradient balance approach for long-tailed object detection. pages 1685–1694, 2021. 3, 6

[46] Jason Van Hulse, Taghi M Khoshgoftaar, and Amri Napolitano. Experimental perspectives on learning from imbalanced data. *Proceedings of the 24th international conference on Machine learning*, pages 935–942, 2007. 3, 6

[47] Fu-Yun Wang, Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan. Foster: Feature boosting and compression for class-incremental learning. *Proceedings of the European Conference on Computer Vision*, pages 398–414, 2022. 2, 7, 3, 4

[48] Max Welling. Herding dynamical weights to learn. *Proceedings of the International Conference on Machine Learning*, pages 1121–1128, 2009. 2, 3

[49] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 1, 2, 5, 7, 8, 3, 4

[50] Shipeng Yan, Jiangwei Xie, and Xuming He. Der: Dynamically expandable representation for class incremental learning. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3014–3023, 2021. 2

[51] Lu Yang, He Jiang, Qing Song, and Jun Guo. A survey on long-tailed visual recognition. *International Journal of Computer Vision*, 130(7):1837–1872, 2022. 3

[52] Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. *International Conference on Learning Representations*, 2018. 2

[53] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. *International Conference on Machine Learning*, pages 3987–3995, 2017. 2

[54] Shaoyu Zhang, Chen Chen, Xiyuan Hu, and Silong Peng. Balanced knowledge distillation for long-tailed learning. *Neurocomputing*, 527:36–46, 2023. 8

[55] Yifan Zhang, Bingyi Kang, Bryan Hooi, Shuicheng Yan, and Jiashi Feng. Deep long-tailed learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 3

[56] Bowen Zhao, Xi Xiao, Guojun Gan, Bin Zhang, and Shu-Tao Xia. Maintaining discrimination and fairness in class incremental learning. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 13208–13217, 2020. 1, 2, 5, 6, 7, 8, 3, 4

[57] Borui Zhao, Quan Cui, Renjie Song, Yiyu Qiu, and Jiajun Liang. Decoupled knowledge distillation. *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 11953–11962, 2022. 5, 8