TaE: Task-aware Expandable Representation for Long Tail Class Incremental Learning

Linjie Li[1], Zhenyu Wu[1⋆], Jiaming Liu[2], and Yang Ji[1]

[1] School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing, 100876, China
[2] School of Computer Science, Peking University, Beijing, 100871, China
shower0512@bupt.edu.cn

Abstract. Class-incremental learning is dedicated to the development of deep learning models that are capable of acquiring new knowledge while retaining previously learned information. Most methods focus on balanced data distribution for each task, overlooking real-world long-tailed distributions. Therefore, Long-Tailed Class-Incremental Learning has been introduced, which trains on data where head classes have more samples than tail classes. Existing methods mainly focus on preserving representative samples from previous classes to combat catastrophic forgetting. Recently, dynamic network algorithms freeze old network structures and expand new ones, achieving significant performance. However, with the introduction of the long-tail problem, merely extending Determined blocks can lead to miscalibrated predictions, while expanding the entire backbone results in an explosion of memory size. To address these issues, we introduce a novel **T**ask-**a**ware **E**xpandable (**TaE**) framework, dynamically allocating and updating task-specific trainable parameters to learn diverse representations from each incremental task while resisting forgetting through the majority of frozen model parameters. To further encourage the class-specific feature representation, we develop a **C**entroid-**E**nhance**d** (**CEd**) method to guide the update of these task-aware parameters. This approach is designed to adaptively allocate feature space for every class by adjusting the distance between intra- and inter-class features, which can extend to all "training from sketch" algorithms. Extensive experiments demonstrate that TaE achieves state-of-the-art performance.

Keywords: Long-tailed Class-incremental Learning · Data imbalanced Learning · Continual Learning · Lifelong Mashine Learning.

1 Introduction

Class Incremental Learning (CIL) aims to establish a unified classifier across all known classes, and the most challenging problem of CIL is catastrophic forgetting [4,10,28]. Conventional CIL makes the assumption that a balanced

⋆ Corresponding author

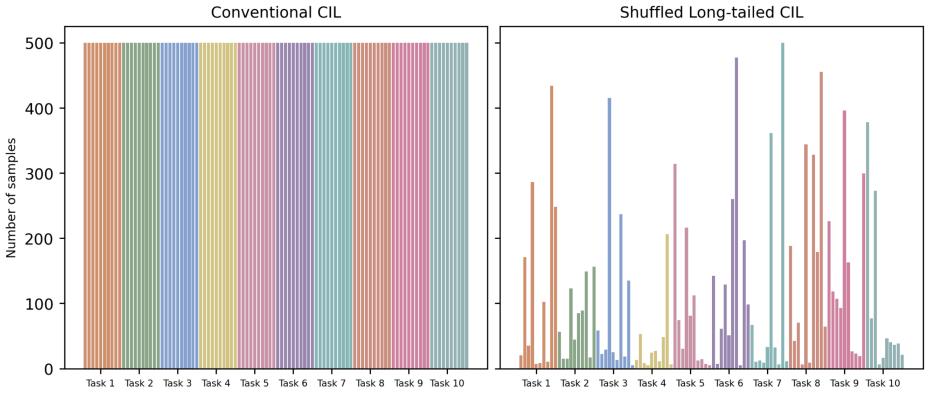


Fig. 1: Illustration of Conventional and Shuffled distribution.

distribution forms the training set. On the other hand, real-world data distributions are typically long-tailed [37,18], meaning that there are significantly more head-class samples than tail-class samples. Proposals have recently been made for Long-tailed Class-incremental learning (LT-CIL)[16]. Shuffled LT-CIL biases the model toward learning head classes while neglecting tail classes during new task training, as illustrated in 1. This leads to a deficiency in the learning of the tail class features. Similarly, when retaining old knowledge, the model experiences more severe forgetting due to the blurring of tail class features. As such, the performance of the CIL approach is decreased by this long-tailed specialized catastrophic forgetting[16].

Currently, dynamic networks exhibit commendable performance in both CIL and LT-CIL [39,35,40,26,34,16]. In LT-CIL tasks, the dynamic network's ability to expand its structure enhances the model's representative capacity, allowing it to better adapt to the data distribution of new tasks. As a result, the model demonstrates stronger learning capabilities in handling long-tailed distributions within new tasks. Dynamic network approaches can be roughly divided into two categories: expanding a part of the network structure, like deep blocks[40], or fully expanding the backbone[35,27]. However, due to the long-tailed distribution, shallow networks' ability to represent is not as general as a balanced distribution. Therefore, only expanding the deep blocks in every task and freezing shallow blocks might lead to miscalibrated predictions. Conversely, fully expanding the entire backbone could result in an explosion of memory size.

In this paper, we introduce a novel **T**ask-**a**ware **E**xpandable (TaE) framework, dynamically allocating and updating task-specific trainable parameters to learn diverse representations from each incremental task, while resisting forgetting through the majority of frozen model parameters. Specifically, before training on a new task, we conduct forward propagation for each sample and accumulate the entire parameter gradients to identify the top-ρ%(eg,.5%, 10%, 20%, 30%) most sensitive parameters. These parameters are then exclusively updated during training. To further enhance class-specific feature representa-

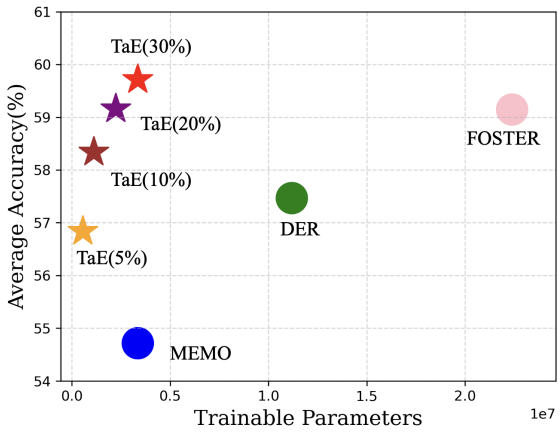


Fig. 2: Parameter-performance comparison of different dynamic network methods on ImageNet100-LT B0-10steps. TaE only expands a few training parameters to exceed the SOTA CIL method.

tion, we develop a **C**entroid-**E**nhance**d** (CEd) method to guide the update of task-aware parameters. We acquire a centroid for each class and maintain a set of centroids. These centroids are treated as learnable parameters, dynamically updated throughout the model training process to adapt to changes in the feature space resulting from the introduction of new data. This approach aims to adaptively minimize intra-class feature distances while maximizing inter-class feature distances among all observed classes. The CEd method enhances the class-specific representation of task-aware parameters, which decreases the overfitting of head classes and underfitting of tail classes.

We conducted experiments on two commonly used benchmark datasets, including CIFAR100 and ImageNet100. We set both datasets with a long-tail configuration and shuffled their order. We set up different ratios of long-tail and different ways of incrementation. Extensive experiments and ablation studies prove the effectiveness of our method. Especially in representative experiments on the Imagenet100 dataset, an extension by just 5% results in surpassing MEMO's performance by 0.64% in final accuracy and 2.12% in average accuracy. Furthermore, an extension by 10% leads to surpassing DER with improvements of 1.20% and 0.87% in final accuracy and average accuracy, respectively, as depicted in 2. The main contributions can be summarized in three points:

- We introduce a novel Task-aware Expandable (TaE) framework to address shuffled LT-CIL challenges, allowing adaptive parameter updates while significantly reducing the expandable model memory size.
- To further encourage class-specific feature representation, we propose a Centroid Enhanced (CEd) method to guide the update of these task-aware parameters and jointly address the challenge of low discriminability of tail class features in LT-CIL.

– The method we proposed achieves state-of-the-art performance on all 12 benchmarks. Notably, our method achieves an average accuracy surpassing the *SOTA* by 3.69% on CIFAR-100 and 2.25% on ImageNet100 under the B0-10 steps with $\rho = 0.1$.

2 Related Work

2.1 Class-incremental Learning

Recent CIL algorithms can be roughly divided into model-centric and algorithm-centric methods [39]. Within the model-centric methods, dynamic network algorithms have demonstrated commendable performances. Yan et al extend a new network backbone when faced with a new task, subsequently aggregating it at the feature level with a larger classifier [35]. Wang et al. contend that not all features are necessarily effective and, building on the foundation set by DER, employ knowledge distillation for model reduction [27]. Zhou et al. decouples the intermediate layers of the network. Extending the deep network, reduces memory overhead, addressing the issue of excessive memory budget cost by dynamic network expansion [40]. On the other hand, within algorithm-centric methods, knowledge distillation methods stand out in terms of efficacy. Li et al were the first to employ knowledge distillation in CIL, establishing a regularization term via knowledge distillation to counteract forgetting [13]. Rebuffi et al. enhance LWF by utilizing an exemplar set and introducing a novel selection method called *herding* [23]. Wang et al. [29] proposed a novel Semantic knowledge-guided ciL framework to address the inter-class confusion problem in incremental learning. This framework utilizes semantic knowledge extracted from class labels to construct two inter-class relation graphs encompassing all encountered old and new classes.

2.2 Long-tailed Learning

The phenomenon of long-tail data distribution is pervasive in the real world, and the challenges of long-tail learning have been studied extensively. Most existing works can be categorized into class rebalancing, information augmentation, and model improvement. Class rebalancing focuses on techniques such as resampling and reweighting. Lin et al. investigated the difficulty of class prediction as a method for re-weighting [14]. Cao et al. impose variable margin factors for distinct classes, determined by their training label occurrences, prompting tail classes to possess broader margins [1]. Beyond basic image manipulations such as rotation and cropping, transfer learning has also emerged as a focal point in recent research on data augmentation. Wang et al introduced a MetaNetwork that transforms few-shot model parameters to many-shot ones [31]. Liu et al. focus on head-to-tail transfer at the classifier tier [15]. Yin et al. leverage head-class variance insights to bolster feature augmentation for tail classes, aiming for greater intra-class variance in tail-class features [36]. In the model improvement section, the most popular research focuses on training by decoupling the

feature extractor and the classifier. Kang et al. first introduced the concept of a two-stage decoupled training scheme [9]. Chu et al. introduced an innovative classifier re-training method using tail-class feature augmentation [2].

2.3 Long-tail Class-incremental Learning

Recently, Liu et al. [16] incorporated long-tail distributions into class incremental learning (LT-CIL), drawing inspiration from the learnable weight scaling (LWS) approach introduced by the decoupling strategy [9], and demonstrated that the long-tail distribution of data not only deteriorates the performance of class incremental learning algorithms but also compromises their robustness. "Based on Liu et al.'s work, Kalla et al. [8] achieve classifier alignment by leveraging global variance as an informative measure and class prototypes in the second stage. While this approach enhances the robustness of tail class features, aligning all classes using the feature space of the largest class may result in an inadequate distribution of the overall feature space, thereby affecting the model's plasticity. Therefore, the CEd proposed in this paper can adaptively adjust the feature space size for each class during the training process. Wang et al.[30] constructed two parallel spaces sub-prototype and reminiscence spaces—to learn robust representations and mitigate forgetting in LT-CIL.

3 Method

3.1 Preliminary

To formalize, we denote the sequence of T training tasks as $\left\{\mathcal{D}^1, \mathcal{D}^2, \cdots, \mathcal{D}^T\right\}$, where each task $\mathcal{D}^t$ is characterized by $\{(\mathbf{x}_i^t, y_i^t)\}_{i=1}^{n_t}$. Here, $\mathbf{x}_i^t \in \mathbb{R}^D$ represents an instance belonging to class $y_i \in Y_t$, and Y_t denotes the label space of task t. Importantly, the label spaces for different tasks are mutually exclusive. During the training of task t, only data from $\mathcal{D}^t$ is accessible. The model's performance is evaluated across all observed class labels, defined as $\mathcal{Y}_t = Y_1 \cup \cdots Y_t$ [39].

The Exemplar Set is an auxiliary collection of instances from previous tasks, denoted as $\mathcal{E} = \{(\mathbf{x}_j, y_j)\}_{j=1}^{M}$, where $y_j \in \mathcal{Y}_{t-1}$. This set, combined with the current dataset, i.e., $\mathcal{E} \cup \mathcal{D}^t$, facilitates model updates within each task [40]. One prominent method for choosing exemplars that is widely used is herding, as suggested in [23].

The cross-entropy loss $\mathcal{L}_{ce}$ at task t is define as:

$$\mathcal{L}_{ce,t}(\mathbf{x}, y) = -\frac{1}{|\mathcal{D}^t| + |\mathcal{E}|} \sum_{(\mathbf{x},y)\in\mathcal{D}^t\cup\mathcal{E}} \mathbf{y} \cdot \log(p_{1:t}(\mathbf{x})) \tag{1}$$

where $\mathbf{y}$ is a one-hot vector in which the position of the ground-truth label is 1, and $p_{1:t}(\mathbf{x})$ is a vector with the probability predictions for image $\mathbf{x}$ overall seen classes.

An overview of our approach is illustrated in Fig.3.

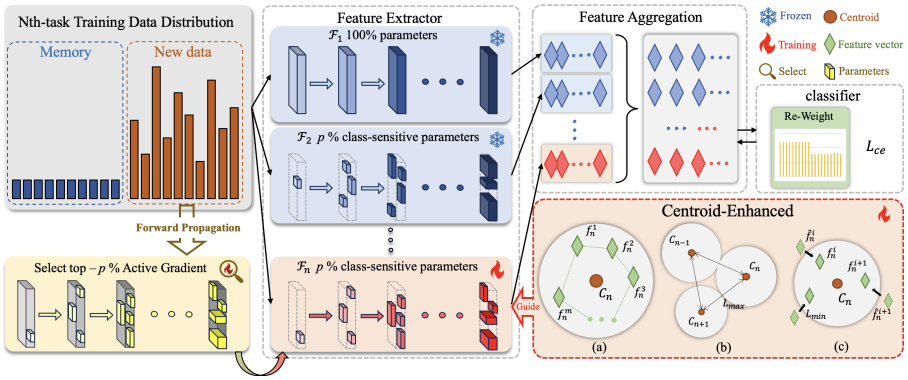


Fig. 3: The overview of TaE. During training for the n task, the training set undergoes multiple forward passes on the model from the previous round. It selects the most sensitive $p\%$ of gradients associated with these parameters, expands these selected parameters, and freezes the others. The learning process is guided by the Centroid-Enhanced (CEd) method: (a) each class learns a centroid updated throughout training; (b) centroids between different classes remain distant; (c) features within the same class converge towards the centroid. This process is governed by the $\mathcal{L}_{max-min}$ loss. The Re-weight strategy trains classifier learning.

3.2 Extraction and Expansion of Task-Aware Parameters

Recent research has observed that the pre-trained backbone exhibits distinct feature patterns at different locations [22,19]. When confronted with downstream tasks, it is unnecessary to completely retrain all parameters. Instead, adjusting the domain-specific parameters based on the features of different task data proves more effective than a full retraining approach [5]. Inspired by this, the domains between new and old tasks are closely related (e.g., both involving image classification tasks) in CIL, and leveraging the characteristics of dynamic network methods, we treat the model learned in the t-1-th task as the "pre-trained network" for the t_{th} task. Consequently, we select task-aware parameters for expansion and freeze the majority of parameters to mitigate forgetting.

Formally, given the training dataset D^t for the t_{th} task, and the $t-1_{th}$ task model $\mathcal{F}_{t-1}$. We initially pass D^t through the $\mathcal{F}_{t-1}$. Through multiple iterations involving forward propagation, we compute the gradient of each parameter and accumulate the changes in gradients over several loops, and the parameters are sorted based on their magnitudes. Finally, the top $p\%$ (e.g., 5%, 10%, 20%, 30%) of the most sensitive parameters are selected. The algorithmic process is illustrated in Algorithm.1.

Algorithm 1 Select Top-p% Most Sensitive Parameters

Input: t^{th} task data $\mathcal{D}^t$, Top percentage p, The number of forward iteration Z
Require: θ // parameter of $\mathcal{F}_{t-1}$,
Output: Top p% sensitive parameters
for $i = 1$ to n **do**
  for $(\mathbf{x}_i^t, y_i^t) \in \mathcal{D}^t$ **do**
    Perform forward pass: $\mathcal{F}_{t-1}(\mathbf{x}_i^t) \rightarrow \hat{y}_i$
    Compute $\mathcal{L}_{ce}$ by Eq 1
    Compute gradients: $\nabla\theta_i = \frac{\partial \mathcal{L}_{ce}}{\partial \theta}$
    Accumulate gradients: $\mathrm{G}_\theta = \mathrm{G}_\theta + \nabla\theta_i$
  end for
end for
Average accumulated gradients: $\bar{\mathrm{G}}_\theta = \frac{\mathrm{G}_\theta}{Z}$
Sort parameters based on gradient magnitude
$\theta_{select} = \mathrm{Top}(\mathrm{G}_\theta, p)$
return θ_{select}

3.3 Centroid-Enhanced Method

Due to the issue of long-tailed distribution, the model often exhibits a bias towards the head classes, leading to a lack of learning from tail class samples. This results in the features of the tail class being less discriminative. This problem with tail classes becomes more pronounced in CIL. To tackle this issue, we introduce a Feature Isolation Strategy designed to steer the model towards learning class-specific features. In detail, we set a learnable centroid for each class of the new task and introduced a centroid bank to store centroids of all seen classes. During training, we freeze centroids of old classes in the centroid bank, learn centroids of new classes, and employ $\mathcal{L}_{min-max}$ to isolate feature spaces among different classes. We dynamically adjust the size of class feature spaces, where $\mathcal{L}_{min}$ encourages intra-class feature cohesion, and $\mathcal{L}_{max}$ promotes inter-class feature separation [17,25,21].

For computing the similarity between feature vectors, we opt to use cosine similarity to measure distance [33], which is thus: $\cos(u, v) = \frac{u \cdot v}{\|u\|\|v\|}$ where u and v are two vectors, $\cdot$ represents the dot product, and $\|\cdot\|$ is the L_2 norm.

The *min* loss ensures that features from samples within the same class gravitate towards their corresponding class centroid. It can be mathematically represented as:

$$\mathcal{L}_{min,t} = -\frac{1}{|\mathcal{D}^t|} \sum_{i=1}^{|\mathcal{D}^t|} \cos(f(\mathbf{x}_i^t), c_{y_i^t}) \tag{2}$$

where $f(\mathbf{x}_i^t)$ is the features of sample $\mathbf{x}_i^t$ fitted by the model, $c_{y_i^t}$ is the centroid of label y_i^t.

The max loss ensures that centroids of different classes are distinct and separated in the feature space. The loss is defined as:

$$\mathcal{L}_{max,t} = \frac{1}{\mathcal{K}_t(\mathcal{K}_t - 1)} \sum_{j=1}^{\mathcal{K}_t} \sum_{k \neq j}^{\mathcal{K}_t} \cos(c_j^t, c_k^t) \tag{3}$$

where $\mathcal{K}_t$ is the number of classes in task t.

The overall loss function at task t is defined as:

$$\mathcal{L}_{all,t} = \gamma_1 \mathcal{L}_{ce,t} + \gamma_2 \mathcal{L}_{min,t} + \gamma_3 \mathcal{L}_{max,t} \tag{4}$$

where γ_1, γ_2, and γ_3 are the hyper-parameter to control the effort of these losses. The details of setting different hyperparameters and comparing experimental results are provided in Sec.4.7.

3.4 Reweight strategy

For the training set of a single task, $\mathcal{E} \cup \mathcal{D}^t$, with a total of $\mathcal{K}_t$ classes and i-th class having a quantity of k_i^t samples. The effective weight for each class is defined as:

$$w_i = \frac{1 - \beta}{1 - \beta^{k_i^t}} \tag{5}$$

where β is a hyperparameter that controls the weights of different classes, and w_i represents the weight for the i-th class. Based on our experiments, we found that setting β to 0.95 yields the best results.

4 EXPERIMENT

4.1 Experiment Setup and Implementation Details

Datasets. CIFAR-100 [12] is composed of 32x32 pixel color images classified into 100 classes. The dataset comprises 50,000 training images, with 500 images per class, and 10,000 evaluation images, featuring 100 images per class. ImageNet-100 [24] is curated by selecting 100 classes from the ImageNet-1000 dataset, which includes 100,000 training images, with 1,000 images per class, and 5,000 evaluation images with 50 images per class.

Datasets Protocol.: We follow the protocol introduced in [23], which involves training all 100 classes through multiple splits, encompassing 5, 10, and 20 incremental steps while maintaining a fixed memory size of 2,000 exemplars across batches. *B0* indicates that the model is trained from sketch rather than pre-trained on an amount of data before incremental learning.

Imbalanced Ratio Protocol. We adopt three protocols for the head-to-tail sample count ratio in the long-tailed distribution: $\rho = 0.1$, $\rho = 0.05$, and $\rho = 0.01$. In Tab.1, the details of three protocols combined with two datasets are presented, including the sample counts for head and tail classes, as well as the sample quantities for each category stored in the memory.

Table 1: Imbalanced Ratio Protocol, H-T-M means the number of head, tail, and memory, k means 1000.

	$\rho = 0.1$	$\rho = 0.05$	$\rho = 0.01$
Dataset	H-T-M	H-T-M	H-T-M
CIFAR100-LT	500-50-20	500-25-20	500-5-5
ImageNet100-LT	1k-100-20	1k-50-20	1k-10-10

Metrics. In CIL methods, we use standard metrics to evaluate performance: "Last" refers to the accuracy on the most recent task, and "Avg" refers to the Average Accuracy, which calculates the accuracy across all observed classes[28]. Let $a_{i,j}$ be the accuracy of the model on the test set of task j after training from task 1 to task i. The Average Accuracy can be calculated as follows:

$$Avg(A_T) = \frac{1}{T}\sum_{j=1}^{T} a_{T,j} \tag{6}$$

Comparison Methods. We compare our method with existing CIL methods: EWC[11], LwF[13], iCaRL[23], Foster[27], MEMO[40], DER[35], and LT-CIL methods: UCIR[7] and PodNet[3] combined with LWS[16] and GVAlign[8]. It is important to note that, where MEMO only extends the backbone by 30%, we set the proportion of TaE extension to $p = 30\%$ when compared with other methods.

Implementation Details. We implement our method using PyTorch [20] and conduct our experiments on two NVIDIA GeForce RTX 3090 GPUs. We applied a long-tailed transformation to both CIFAR-100 and ImageNet100-LT. The experiments leveraged the public implementations of existing CIL methods within the PyCIL framework [38]. For CIFAR-100, we employed ResNet32 [6], with an initial learning rate set at 0.1, which was reduced by a factor of 10 at epochs 80, 120, and 150 (out of a total of 170 epochs). For ImageNet100-LT, we utilized ResNet18 with the same learning rate adjustments. In these experiments, we choose exemplars for memory based on the herding selection strategy [32], in line with previous studies [23].

4.2 Evaluation on CIFAR100-LT

Quantitative Results Tab.2 and Tab.4 summarize the results of the CIFAR-100 benchmark tests. The experimental results demonstrate that, under similar parameter expansion, TaE's task-aware parameter expansion is more suitable for LT-CIL scenarios. The experimental results indicate that our approach outperforms other methods across all six settings in the CIFAR100-LT dataset, which substantiates the strong robustness of our method.

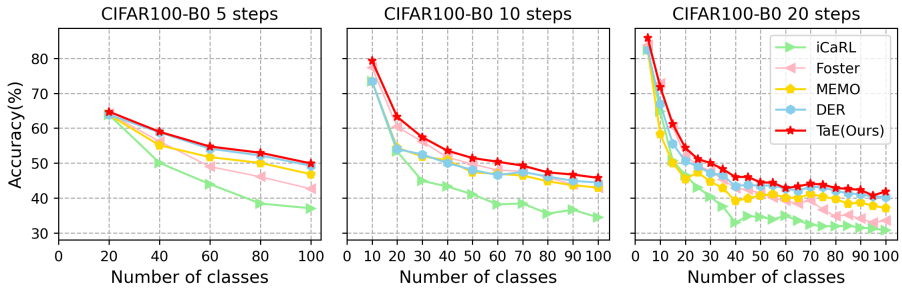


Fig. 4: The performance for each step. The model is trained on CIFAR100-LT ρ = 0.1 with three Datasets Protocols.

Table 2: Performance comparison of typical CIL algorithms on CIFAR100-LT B0-10steps benchmarks, evaluated across three Imbalanced Ratio Protocols.

	$\rho = 0.1$		$\rho = 0.05$		$\rho = 0.01$	
Method	Last	Avg	Last	Avg	Last	Avg
Finetune(baseline)	7.8	22.42	7.57	20.98	5.83	15.36
EWC[11]	8.88	22.37	8.45	21.65	5.63	14.92
LwF[13]	16.82	33.51	16.94	28.45	11.56	20.39
iCaRL[23]	34.43	44.82	33.63	41.64	14.32	22.68
Foster[27]	42.89	53.82	39.86	47.28	15.17	21.72
MEMO[40]	43.02	50.94	40.08	46.31	21.0	26.39
DER[35]	44.43	50.74	39.24	44.37	22.89	28.44
TaE(p=30%)	**46.84**	**55.04**	**40.87**	**48.29**	**23.54**	**29.69**

4.3 Evaluation in ImageNet100-LT

Quantitative Results Fig.5 illustrates the accuracy of different methods at each incremental stage, revealing our approach consistently outperforms others. Tab.3 presents the performance of different methods under three distinct long-tail ratio settings, with TaE consistently demonstrating the best performance across all scenarios.

4.4 Comparison with LT-CIL Methods

Tab.4 presents a performance comparison of our method and three existing LT-CIL methods (LWS [16] and GVAlign [8]) under the same experimental settings. As the table shows, TaE consistently demonstrates superior efficacy across varied experimental conditions.

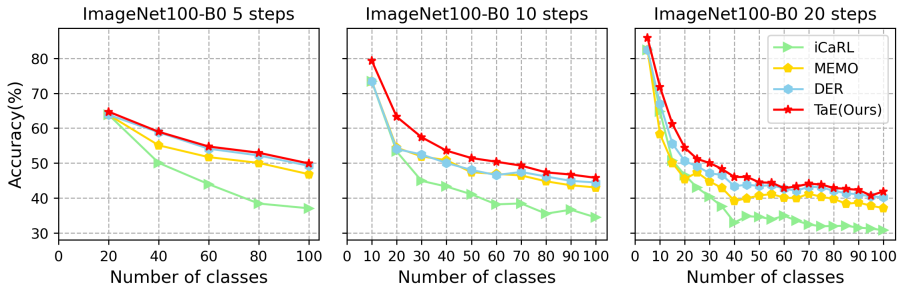


Fig. 5: The performance for each step. The model is trained on ImageNet100-LT B0-10steps $\rho = 0.1$.

Table 3: Performance comparison of typical CIL algorithms on ImageNet100-LT B0-10steps benchmarks, evaluated across three Imbalanced Ratio Protocols.

	$\rho = 0.1$		$\rho = 0.05$		$\rho = 0.01$	
Method	Last	Avg	Last	Avg	Last	Avg
Finetune(baseline)	8.9	23.62	9.12	21.24	6.10	15.77
EWC[11]	10.83	24.74	9.32	23.02	7.06	17.17
LwF[13]	19.62	38.76	18.04	34.67	14.18	25.11
iCaRL[23]	33.72	44.40	32.98	42.51	15.78	23.49
Foster[27]	49.08	59.15	44.07	50.40	18.10	23.12
MEMO[40]	48.78	54.73	44.10	50.40	25.72	30.47
DER[35]	50.32	57.48	41.22	50.69	26.82	31.22
TaE(p=30%)	**52.86**	**59.73**	**44.69**	**55.28**	**29.98**	**33.75**

4.5 Ablation Study

To validate the effectiveness of each component in TaE, we conducted an ablation study on CIFAR100-LT B10-10steps with $\rho = 0.1$. As shown in Tab.5, both the average and final accuracy progressively improved as we added more components. We observe that, building upon TaE(30%), the inclusion of Re-weight results in an average accuracy improvement of 1.16%, and further incorporating $\mathcal{L}_{max-min}$ leads to an additional average accuracy enhancement of 1.35%.

4.6 Discussion on the Selection of Parameter Ratios

We conducted experiments on the ImageNet100-LT and CIFAR100-LT datasets using the B0-10steps benchmark with $p = 0.1$, comparing the performance of TaE $(5\%, 10\%, 20\%, 30\%)$ with MEMO and DER. When training the model on ImageNet100-LT, as illustrated in Fig.6b, the performance of TaE increases with higher values of p. TaE surpasses MEMO at $p = 5\%$ and exceeds DER at $p = 10\%$. TaE(30%) outperforms MEMO in terms of both final accuracy

Table 4: Performance comparison of LTCIL algorithms on CIFAR100-LT and ImageNet100-LT B0-5steps and 10steps with $\rho = 0.1$ benchmarks.

		CIFAR100-LT		ImageNet100-LT	
Method	Conference	B0-10steps	B0-5steps	B0-10steps	B0-5steps
UCIR(+LWS)[16]	ECCV2022	39.40	39.00	49.42	47.96
PodNet(+LWS)[16]	ECCV2022	36.37	37.03	49.75	49.51
UCIR(+GVAlign)[8]	WACV2024	42.80	41.64	50.69	47.58
PodNet(+GVAlign)[8]	WACV2024	42.72	41.61	52.01	50.81
TaE(p=30%)	ACCV2024	**54.97**	**55.04**	**59.73**	**64.48**

Table 5: Ablations of key components in TaE(30%). We report the average and last accuracy on CIFAR100-LT B0-10 steps. RW means Re-Weight strategy.

RW	$\mathcal{L}_{max-min}$	Last	Avg
✗	✗	45.03	51.92
✓	✗	45.79	53.08
✓	✓	46.84	55.04

and average accuracy, achieving superiority of 4.08% and 5.00%, respectively. Moreover, compared to DER, TaE(30%) exhibits a superiority of 2.54% in final accuracy and 2.25% in average accuracy.

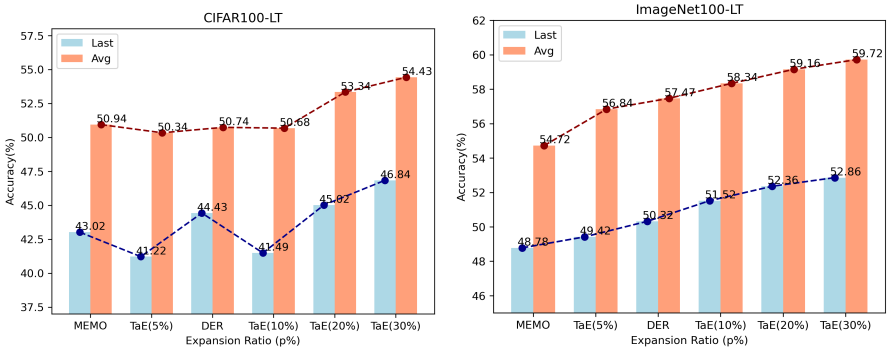
However, as illustrated in Fig.6a, when training the model on CIFAR100, TaE only surpasses DER and MEMO at $p = 20\%$. TaE(30%) surpasses MEMO in both final accuracy and average accuracy, demonstrating advantages of 3.82% and 3.39%, respectively. Additionally, in comparison to DER, TaE(30%) demonstrates a superiority of 2.41% in final accuracy and 3.71% in average accuracy.

4.7 Discussion on Hyper-parameter

We conducted a sensitivity analysis on the hyperparameters in the Eq.4. For simplicity, we set γ_1 to 1 and explored various combinations of γ_2 and γ_3 to assess the importance of $\mathcal{L}_{min}$ and $\mathcal{L}_{max}$. From the experimental results, it is evident that the performance of all hyperparameter combinations surpasses that of the State-of-the-Art method, confirming the robustness of TaE. The best performance is achieved when the values of γ_2 and γ_3 are close, and the model slightly leans towards emphasizing inter-class distances. The results show a final accuracy of 47.19% and an average accuracy of 54.72%, surpassing the SOTA by 2.76% and 3.98%, respectively.

4.8 Effectiveness of Centroid-Enhanced Method

The centroid-enhanced(**CEd**) method can be applied to most "training from sketch" CIL methods. In this section, we will validate the effectiveness of the



(a) (b)

Fig. 6: The Last and Average Accuracy. We compare our model(p = 5%, 10%, 20%, 30%) with *SOTA* CIL methods (DER [35] and MEMO [40]), and the model trained on the experiment CIFAR100-LT and ImageNet100-LT B0-10steps with $\rho = 0.1$.

Table 6: Performance comparison of four CIL algorithms combined with CEd on the CIFAR100 $\rho = 0.1$ benchmarks, evaluated across three Datasets Protocols.

	B0-5 steps		B0-10 steps		B0-20 steps	
Method	Last	Avg	Last	Avg	Last	Avg
LwF[13]	8.15	23.12	17.56	32.87	27.02	40.09
w CEd	**8.86**	**24.23**	**17.84**	**33.79**	**30.99**	**41.22**
iCarL[23]	37.05	46.51	34.13	43.47	30.74	39.59
w CEd	**43.76**	**52.35**	**41.24**	**51.95**	**33.35**	**43.51**
DER[35]	46.26	52.91	44.43	50.74	39.70	46.31
w CEd	**48.51**	**56.50**	**45.07**	**53.23**	**40.05**	**47.52**
Foster[27]	41.82	51.46	40.44	51.23	35.03	45.51
w CEd	**42.08**	**51.73**	**40.98**	**52.67**	**36.97**	**48.12**

centroid set in various experimental settings. In Tab.6, we present the experimental results on the CIFAR100 with $\rho = 0.1$. We applied CEd to four representative CIL methods. Our experiments were conducted under three protocols: B0-5steps, B0-10steps, and B0-20steps.

The experimental results reveal that the inclusion of the centroid set enhances the performance of four representative CIL methods. We can observe improvements across various benchmarks for all four CIL methods after the incorporation of the CEd approach. Particularly, in the experimental setup of iCaRL with B0-10 steps, the final accuracy and average accuracy in the last stage increased by 7.11% and 8.48%, respectively. The average accuracy even surpasses that of DER

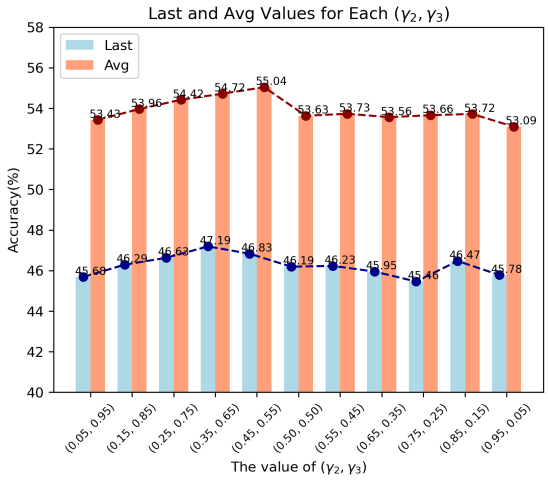


Fig. 7: The Last and Average Accuracy. We compare our model($p = 30\%$) across various combinations of $\mathcal{L}_{max}$ and $\mathcal{L}_{min}$ and the model trained on the experiment CIFAR100-LT B0-10steps with $\rho = 0.1$.

and Foster under the same experimental conditions. In other benchmarks, the improvement of iCaRL is notably pronounced. The above experiments demonstrate that our proposed centroid method effectively strengthens the ability of CIL to handle LT-CIL.

5 Conclusions

In our study, we have developed a task-aware expandable framework to tackle the challenges associated with long-tail class incremental learning. We identify the most gradient-sensitive parameters in the final model after each training task by utilizing forward propagation to achieve this. We then expand and train these parameters to improve the model's representational capabilities without causing memory overflow. Additionally, we have introduced a centroid-enhanced method to facilitate the training of task-aware parameters. This method can be adapted to various class-incremental learning methods. Our approach was rigorously evaluated through comprehensive experimentation. The experimental results demonstrate the superiority of our method. We believe that our work will lay the groundwork for future research in the field of long-tailed class-incremental learning.

Acknowledgments. The paper is supported by the National Natural Foundation Science of China (62101061).

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Cao, K., Wei, C., Gaidon, A., Arechiga, N., Ma, T.: Learning imbalanced datasets with label-distribution-aware margin loss. Advances in neural information processing systems **32** (2019) 4
2. Chu, P., Bian, X., Liu, S., Ling, H.: Feature space augmentation for long-tailed data. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIX 16. pp. 694–710. Springer (2020) 5
3. Douillard, A., Cord, M., Ollion, C., Robert, T., Valle, E.: Podnet: Pooled outputs distillation for small-tasks incremental learning. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16. pp. 86–102. Springer (2020) 9
4. French, R.M.: Catastrophic forgetting in connectionist networks. Trends in cognitive sciences **3**(4), 128–135 (1999) 1
5. He, H., Cai, J., Zhang, J., Tao, D., Zhuang, B.: Sensitivity-aware visual parameter-efficient tuning. arXiv preprint arXiv:2303.08566 (2023) 6
6. He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14. pp. 630–645. Springer (2016) 9
7. Hou, S., Pan, X., Loy, C.C., Wang, Z., Lin, D.: Learning a unified classifier incrementally via rebalancing. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 831–839 (2019) 9
8. Kalla, J., Biswas, S.: Robust feature learning and global variance-driven classifier alignment for long-tail class incremental learning. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 32–41 (2024) 5, 9, 10, 12
9. Kang, B., Xie, S., Rohrbach, M., Yan, Z., Gordo, A., Feng, J., Kalantidis, Y.: Decoupling representation and classifier for long-tailed recognition. arXiv preprint arXiv:1910.09217 (2019) 5
10. Kim, D., Han, B.: On the stability-plasticity dilemma of class-incremental learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20196–20204 (2023) 1
11. Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A.A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al.: Overcoming catastrophic forgetting in neural networks. Proceedings of the national academy of sciences **114**(13), 3521–3526 (2017) 9, 10, 11
12. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009) 8
13. Li, Z., Hoiem, D.: Learning without forgetting. IEEE transactions on pattern analysis and machine intelligence **40**(12), 2935–2947 (2017) 4, 9, 10, 11, 13
14. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. pp. 2980–2988 (2017) 4
15. Liu, B., Li, H., Kang, H., Hua, G., Vasconcelos, N.: Gistnet: a geometric structure transfer network for long-tailed recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8209–8218 (2021) 4
16. Liu, X., Hu, Y.S., Cao, X.S., Bagdanov, A.D., Li, K., Cheng, M.M.: Long-tailed class incremental learning. In: European Conference on Computer Vision. pp. 495–512. Springer (2022) 2, 5, 9, 10, 12

17. Martinetz, T., Schulten, K., et al.: A" neural-gas" network learns topologies (1991) 7
18. Muñoz, D., Narváez, C., Cobos, C., Mendoza, M., Herrera, F.: Incremental learning model inspired in rehearsal for deep convolutional networks. Knowledge-Based Systems **208**, 106460 (2020) 2
19. Muzammal, N., Kanchana, R., Khan Salman, H., Munawar, H., Fahad, S.K., Ming-Hsuan, Y.: Intriguing properties of vision transformers. arXiv preprint arXiv:2105.10497 **3** (2021) 6
20. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch (2017) 9
21. Prudent, Y., Ennaji, A.: An incremental growing neural gas learns topologies. In: Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005. vol. 2, pp. 1211–1216. IEEE (2005) 7
22. Raghu, M., Unterthiner, T., Kornblith, S., Zhang, C., Dosovitskiy, A.: Do vision transformers see like convolutional neural networks? Advances in Neural Information Processing Systems **34**, 12116–12128 (2021) 6
23. Rebuffi, S.A., Kolesnikov, A., Sperl, G., Lampert, C.H.: icarl: Incremental classifier and representation learning. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 2001–2010 (2017) 4, 5, 8, 9, 10, 11, 13
24. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International journal of computer vision **115**, 211–252 (2015) 8
25. Tao, X., Hong, X., Chang, X., Dong, S., Wei, X., Gong, Y.: Few-shot class-incremental learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12183–12192 (2020) 7
26. Wang, F.Y., Zhou, D.W., Liu, L., Ye, H.J., Bian, Y., Zhan, D.C., Zhao, P.: Beef: Bi-compatible class-incremental learning via energy-based expansion and fusion. In: The Eleventh International Conference on Learning Representations (2022) 2
27. Wang, F.Y., Zhou, D.W., Ye, H.J., Zhan, D.C.: Foster: Feature boosting and compression for class-incremental learning. In: European conference on computer vision. pp. 398–414. Springer (2022) 2, 4, 9, 10, 11, 13
28. Wang, L., Zhang, X., Su, H., Zhu, J.: A comprehensive survey of continual learning: Theory, method and application. IEEE Transactions on Pattern Analysis and Machine Intelligence (2024) 1, 9
29. Wang, S., Shi, W., Dong, S., Gao, X., Song, X., Gong, Y.: Semantic knowledge guided class-incremental learning. IEEE Transactions on Circuits and Systems for Video Technology (2023) 4
30. Wang, X., Yang, X., Yin, J., Wei, K., Deng, C.: Long-tail class incremental learning via independent sub-prototype construction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 28598–28607 (2024) 5
31. Wang, Y.X., Ramanan, D., Hebert, M.: Learning to model the tail. Advances in neural information processing systems **30** (2017) 4
32. Welling, M.: Herding dynamical weights to learn. In: Proceedings of the 26th Annual International Conference on Machine Learning. pp. 1121–1128 (2009) 9
33. Xia, P., Zhang, L., Li, F.: Learning similarity with cosine similarity ensemble. Information sciences **307**, 39–52 (2015) 7
34. Xiang, L., Ding, G., Han, J.: Learning from multiple experts: Self-paced knowledge distillation for long-tailed classification. In: Computer Vision–ECCV 2020: 16th

European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16. pp. 247–263. Springer (2020) 2
35. Yan, S., Xie, J., He, X.: Der: Dynamically expandable representation for class incremental learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3014–3023 (2021) 2, 4, 9, 10, 11, 13
36. Yin, X., Yu, X., Sohn, K., Liu, X., Chandraker, M.: Feature transfer learning for face recognition with under-represented data. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5704–5713 (2019) 4
37. Zhang, Y., Kang, B., Hooi, B., Yan, S., Feng, J.: Deep long-tailed learning: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence (2023) 2
38. Zhou, D.W., Wang, F.Y., Ye, H.J., Zhan, D.C.: Pycil: A python toolbox for class-incremental learning (2023) 9
39. Zhou, D.W., Wang, Q.W., Qi, Z.H., Ye, H.J., Zhan, D.C., Liu, Z.: Deep class-incremental learning: A survey. arXiv preprint arXiv:2302.03648 (2023) 2, 4, 5
40. Zhou, D.W., Wang, Q.W., Ye, H.J., Zhan, D.C.: A model or 603 exemplars: Towards memory-efficient class-incremental learning. arXiv preprint arXiv:2205.13218 (2022) 2, 4, 5, 9, 10, 11, 13