Continual Learning in the Frequency Domain

Ruiqi Liu[1,2], **Boyu Diao**[1,2,†], **Libo Huang**[1], **Zijia An**[1,2], **Zhulin An**[1,2], **Yongjun Xu**[1,2]
[1]Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
[2]University of Chinese Academy of Sciences, Beijing, China
liuruiqi23@mails.ucas.ac.cn,
{diaoboyu2012, anzijia23p, anzhulin, xyj}@ict.ac.cn,
www.huanglibo@gmail.com

Abstract

Continual learning (CL) is designed to learn new tasks while preserving existing knowledge. Replaying samples from earlier tasks has proven to be an effective method to mitigate the forgetting of previously acquired knowledge. However, the current research on the training efficiency of rehearsal-based methods is insufficient, which limits the practical application of CL systems in resource-limited scenarios. The human visual system (HVS) exhibits varying sensitivities to different frequency components, enabling the efficient elimination of visually redundant information. Inspired by HVS, we propose a novel framework called Continual Learning in the Frequency Domain (CLFD). To our knowledge, this is the first study to utilize frequency domain features to enhance the performance and efficiency of CL training on edge devices. For the input features of the feature extractor, CLFD employs wavelet transform to map the original input image into the frequency domain, thereby effectively reducing the size of input feature maps. Regarding the output features of the feature extractor, CLFD selectively utilizes output features for distinct classes for classification, thereby balancing the reusability and interference of output features based on the frequency domain similarity of the classes across various tasks. Optimizing only the input and output features of the feature extractor allows for seamless integration of CLFD with various rehearsal-based methods. Extensive experiments conducted in both cloud and edge environments demonstrate that CLFD consistently improves the performance of state-of-the-art (SOTA) methods in both precision and training efficiency. Specifically, CLFD can increase the accuracy of the SOTA CL method by up to 6.83% and reduce the training time by 2.6×. Code is available at https://github.com/EMLS-ICTCAS/CLFD.git

1 Introduction

Continual learning (CL) enables machine learning models to adjust to new data while preserving previous knowledge in dynamic environments [25]. Traditional training methods often underperform in CL because the adjustments to the parameters prioritize new information over old information, leading to what is commonly known as *catastrophic forgetting* [36]. While recent CL methods primarily concentrate on addressing the issue of forgetting, it is imperative to also consider learning efficiency when implementing CL applications on edge devices with constrained resources [37, 31], such as the NVIDIA Jetson Orin NX.

To mitigate *catastrophic forgetting*, a wide range of methods have been employed: *regularization-based* methods [49, 42, 30, 11, 3] constrain updates to essential parameters, minimizing the drift in network parameters that are crucial for addressing previous tasks; *architecture-based* methods [14, 24,

[†]Corresponding Author.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

35, 45, 52] allocate distinct parameters for each task or incorporate additional network components upon the arrival of new tasks to decouple task-specific knowledge; and *rehearsal-based* methods [2, 6, 8, 12, 10] effectively prevent forgetting by maintaining an episodic memory buffer and continuously replaying samples from previous tasks. Among these methods, rehearsal-based methods have been proven to be the most effective in mitigating *catastrophic forgetting* [6]. However, when the buffer size is constrained by memory limitations (e.g., on edge devices), accurately approximating the joint distribution using limited samples becomes challenging. Moreover, rehearsal-based methods often require frequent data retrieval from buffers. This process significantly increases both computational demands and memory usage, consequently limiting the practical application of rehearsal-based methods in resource-constrained environments.

By reducing the size of the input image, both the training FLOPs and peak memory usage can be significantly decreased, thereby enhancing the training efficiency of rehearsal-based methods. Concurrently, this method allows rehearsal-based methods to store more samples within the same buffer. However, directly downsampling the input image can significantly degrade the model's performance due to information loss. Owing to the natural smoothness of images, the human visual system (HVS) exhibits greater sensitivity to low-frequency components than to high-frequency components [48, 34], enabling the efficient elimination of visually redundant information. Inspired by HVS, we transfer the CL methods from the spatial domain to the frequency domain and reduce the size of input feature maps in the frequency domain. Several studies [48, 15, 13] have focused on accelerating model training in the frequency domain. However, two primary limitations hinder their direct application to CL: (1) These studies utilize Discrete Cosine Transform (DCT) to map images into the frequency domain, resulting in a complete loss of spatial information, which prevents the use of data augmentation techniques in rehearsal-based methods. (2) These studies introduce a significant number of cross-task learnable parameters, consequently increasing the risk of *catastrophic forgetting*.

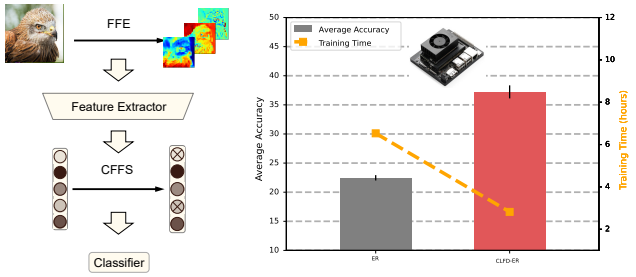

Figure 1: **Left**: Overview of CLFD. CLFD consists of two components: Frequency Domain Feature Encoder (FFE) and Class-aware Frequency Domain Feature Selection (CFFS). **Right**: On the NVIDIA Jetson Orin NX edge device, CLFD demonstrates a notable enhancement in both accuracy and efficiency compared to ER [4] on the split CIFAR-10 dataset.

To this end, we propose a novel framework called Continual Learning in the Frequency Domain (CLFD), which comprises two components: Frequency Domain Feature Encoder (FFE) and Class-aware Frequency Domain Feature Selection (CFFS). To reduce the size of input images, we propose the FFE. This method utilizes Discrete Wavelet Transform (DWT) to transform the original RGB image input into the frequency domain, thereby preserving both the frequency domain and spatial domain features of the image, which facilitates data augmentation. Furthermore, acknowledging that distinct tasks exhibit varying sensitivities to different frequency components, we propose the CFFS method to balance the reusability and interference of frequency domain features. CFFS calculates the frequency domain similarity between inputs across different classes and selects distinct frequency domain features for classification. This method promotes the use of analogous frequency domain features for categorizing semantically similar inputs while concurrently striving to diminish the overlap of frequency domain features among inputs with divergent semantics. Our framework avoids introducing any cross-task learnable parameters, thereby reducing the risk of *catastrophic forgetting*. Simultaneously, by optimizing only the input and output features of the feature extractor, it facilitates the seamless integration of CLFD with various rehearsal-based methods. Figure 1 (right) demonstrates that CLFD significantly improves both the training efficiency and accuracy of rehearsal-based methods when implemented on the edge device.

In summary, our contributions are as follows:

- We propose the CLFD, a novel framework designed to improve the efficiency of CL. This framework enhances the training by mapping input features in the frequency domain and compressing these frequency domain features. To the best of our knowledge, this study represents the first attempt to

utilize frequency domain features to enhance the performance and the efficiency of CL on edge devices.

- CLFD stores and replays encoded feature maps instead of original images, thereby enhancing the efficiency of storage resource utilization. Concurrently, CLFD minimizes interference among frequency domain features, significantly boosting the performance of rehearsal-based methods across all benchmark datasets. These improvements can increase accuracy by up to 6.83% compared to the SOTA methods.
- We evaluate the CLFD framework on an actual edge device, showcasing its practical feasibility. The results indicate that our framework can achieve up to a 2.6× improvement in training speed and a 3.0× reduction in peak memory usage.

2 Related Work

2.1 Continual Learning

The current CL methods can be categorized into three primary types: *Regularization-based* methods [42, 30, 11, 22, 23] limit updates to key parameters to minimize drift in network parameters essential for previous tasks. *Architecture-based* methods [14, 24, 35, 45, 52] assign distinct parameters to each task or add network components for new tasks to decouple task-specific knowledge. *Rehearsal-based* methods [2, 6, 8, 12, 10] mitigate forgetting by maintaining an episodic memory buffer and continuously replaying samples from previous tasks to approximate the joint distribution of tasks during training. Among these, our framework focuses on rehearsal-based methods, as these methods are acknowledged as the most effective in mitigating *catastrophic forgetting* [6]. ER [38] enhances CL by integrating training samples from both the current and previous tasks. Expanding upon this concept, DER++ [6] enhances the learning process by retaining previous model output logits and utilizing a consistency loss during the model update. ER-ACE [7] safeguards learned representations and minimizes drastic adjustments required for adapting to new tasks, thereby mitigating *catastrophic forgetting*. Moreover, CLS-ER [4] mimics the interaction between rapid and prolonged learning processes by maintaining two supplementary semantic memories.

A limited number of works explore training efficiency in CL [46, 27, 16]. Among these methods, SparCL [46] reduces the FLOPs required for model training through the implementation of dynamic weight and gradient masks, along with selective sampling of crucial data. These methods accelerate the training process through pruning and sparse training. Nevertheless, our framework enhances efficiency by reducing the size of the input feature map, which is an orthogonal optimization to pruning.

2.2 Frequency domain learning

Some studies [48, 15, 18, 21] utilize DCT to map images into the frequency domain and enhance the inference speed of the models. However, these methods are not conducive to enhancing rehearsal-based methods. Previous research [6] indicates that data augmentation can significantly boost the performance of rehearsal-based methods. Nevertheless, utilizing DCT results in a total loss of spatial information, thereby restricting the application of data augmentation. Other studies [29, 33, 32, 47, 17, 13] employ DWT to improve the classification performance of models. While wavelet transform effectively preserves the spatial features of images, these methods are not well-suited for CL due to the substantial increase in learnable parameters they introduce. In CL, this proliferation of parameters significantly raises the risk of *catastrophic forgetting* across tasks. MgSvf [51] utilizes the frequency domain in the context of CL, focusing on the influence of different frequency components on model performance. In contrast, our framework delves into the differences in redundancy between the spatial and frequency domains. Compared to the spatial domain, CL in the frequency domain can more effectively remove redundant information from images, thereby improving the efficiency of CL.

3 Method

Our method, called Continual Learning in the Frequency Domain, is a unified framework that integrates two components: the Frequency Domain Feature Encoder, which transforms the initial RGB image inputs into the wavelet domain, and the Class-aware Frequency Domain Feature Selection,

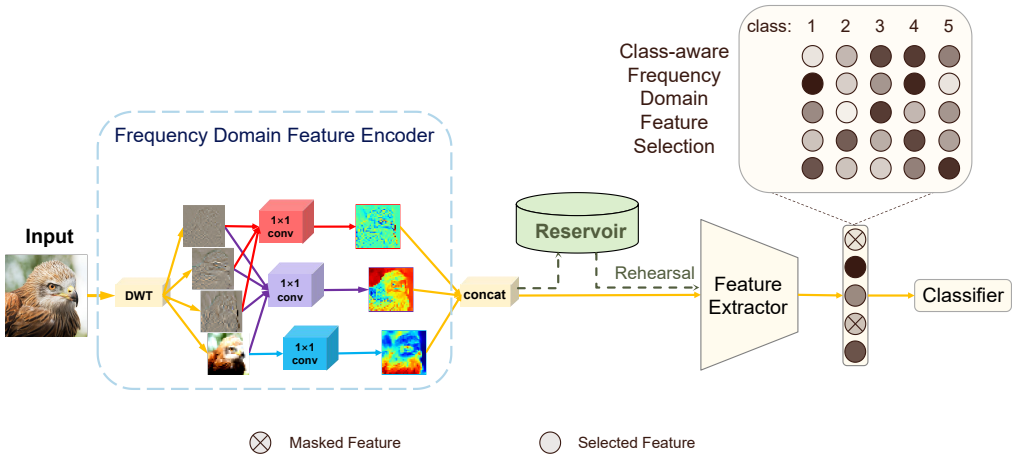

Figure 2: Illustration of the CLFD workflow. Initially, the original RGB image input is transformed into the wavelet domain through a Frequency Domain Feature Encoder. Subsequently, the feature extractor extracts the frequency domain features of these input feature maps. We propose a Class-aware Frequency Domain Feature Selection to selectively utilize specific frequency domain features, which are then inputted into the classifier for subsequent classification.

which balances the reusability and interference of frequency domain features. The entire framework is illustrated in Figure 2.

3.1 Problem Setting

The problem of CL involves sequentially learning T tasks from a dataset, where in each task t corresponds to a training set $\mathcal{D}^t = (x_i, y_i)_{i=1}^{N_t}$. Each task is characterized by a task-specific data distribution represented by the pairs (x_i, y_i). To improve knowledge retention from previous tasks, we employ a fixed-size memory buffer, $\mathcal{M} = (x_i, y_i)_{i=1}^{\mathcal{B}}$, which stores data from tasks encountered earlier. Given the inherent limitations in CL, the model's storage capacity for past experiences is finite, thus $\mathcal{B} \ll N_t$. To address this constraint, we utilize reservoir sampling [44] to efficiently manage the memory buffer. In the simplest testing configuration, we assume that the identity of each upcoming test instance is known, a scenario defined as Task Incremental Learning (Task-IL). If the class subset of each sample remains unidentified during CL inference, the situation escalates to a more complex Class Incremental Learning (Class-IL) setting. This research primarily focuses on the more intricate Class-IL setting, while the performance of Task-IL is used solely for comparative analysis.

3.2 Discrete Wavelet Transform

DWT offers effective signal representation in both spatial and frequency domains [28], facilitating the reduction of input feature size. Compared with the DWT, DCT coefficients predominantly capture the global information of an image, but they fail to preserve the spatial continuity that is typical in normal images. In DCT, local spatial information is mixed, resulting in a loss of distinct local features. In contrast, DWT effectively integrates both spatial and frequency domain information, maintaining a balance between the two. Furthermore, the DWT method can be seamlessly integrated with data augmentation techniques in rehearsal-based methods, enhancing its applicability and effectiveness.

For 2D signal $X \in \mathbb{R}^{N \times N}$, The signal after DWT can be represented as:

$$X' = \begin{bmatrix} L \\ H \end{bmatrix} X \begin{bmatrix} L^T & H^T \end{bmatrix} = \begin{bmatrix} LXL^T & LXH^T \\ HXL^T & HXH^T \end{bmatrix} = \left[\begin{array}{c|c} X_{ll} & X_{lh} \\ \hline X_{hl} & X_{hh} \end{array}\right], \tag{1}$$

where L and H represent the low-frequency and high-frequency filters of orthogonal wavelets, respectively. These filters are truncated to the size of $\lfloor \frac{N}{2} \rfloor \times N$. The term X_{ll} refers to the low-frequency component, while X_{lh}, X_{hl}, X_{hh} represents the high-frequency components. We

select the Haar wavelet as the basis for the wavelet transform because of its superior computational efficiency [28], which is well-suited for our tasks.

3.3 Frequency Domain Feature Encoder

Previous methods [29, 47, 28] typically discarded high-frequency components X_{lh}, X_{hl}, X_{hh} and retained low-frequency component X_{ll}. However, focusing solely on low-frequency component leaves many potentially useful frequency components unexplored. Low-frequency component compress the global topological information of an image at various levels, while high-frequency components reveal the image's structure and texture [20].

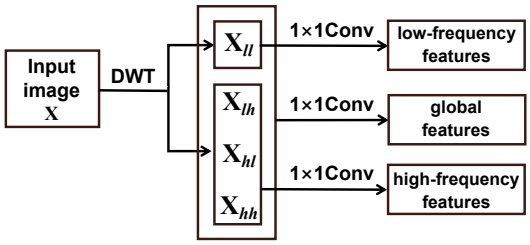

Figure 3: The Utilization of DWT in FFE.

Therefore, we employ three 1×1 point convolutions to integrate various frequency components. As shown in Figure 3, we use a 1×1 point convolution to merge low-frequency component X_{ll} to obtain low-frequency features, another one to merge high-frequency components X_{lh}, X_{hl}, X_{hh} to obtain high-frequency features, and a final one to merge all frequency components to obtain global features. These three merged features compose the input feature maps. By utilizing both low and high-frequency components in CL, we can better prevent the loss of critical information while reducing input feature size. Since each merged feature's width and height are half of the original image, another advantage of working in the frequency domain is that the spatial size of the original image (H $\times$ W $\times$ 3) is reduced by half in both width and height (H/2 $\times$ W/2 $\times$ 3) after FFE. With the reduced spatial size, the computational load and peak memory requirements of CL models decrease. Moreover, the reduction in spatial size means that more replay samples can be stored under the same storage resources, while also reducing the bandwidth required for accessing data. However, setting a specific frequency domain feature encoder for each task may result in significant *catastrophic forgetting*. Therefore, we freeze the FFE at the end of the first task's training.

3.4 Class-aware Frequency Domain Feature Selection

Considering that tasks are predominantly sensitive to specific frequency domain features extracted by a feature extractor, different tasks prioritize distinct frequency domain features. To this end, we propose the CFFS, designed to manage the issue of overlap in frequency domain features among samples from different classes. This method promotes comparable classes to utilize similar frequency domain features for classification, while also ensuring that samples from dissimilar classes employ divergent features. Consequently, this method reduces interference among various tasks and mitigates overfitting issues. For specific classes, we select a predetermined number of frequency domain features based on the absolute values of these features. Subsequently, unselected features are masked to prevent their interference in the classification process. We utilize a counter $\mathcal{F} \in \mathbb{R}^{C\times N}$ to track the number of selections for each frequency domain feature among samples associated with a specific class. N and C denote the dimensions and the classes of frequency domain features, respectively. We utilize cosine similarity to evaluate the frequency domain similarity between two class samples. To decrease computational complexity, only low-frequency component X_{ll} is utilized for calculating cosine similarity. The similarity between class i and class j is expressed as follows:

$$S_{\mathrm{ij}} = \frac{\mathbf{f}_{\mathrm{i}}^{\top}\mathbf{f}_{\mathrm{j}}}{\|\mathbf{f}_{\mathrm{i}}\| \cdot \|\mathbf{f}_{\mathrm{j}}\|}. \tag{2}$$

The value of cosine similarity is determined solely by the direction of the features, regardless of their magnitude. Consequently, $\mathbf{f}_{\mathrm{i}}$ represents the sum of the low-frequency component of samples in class i. We then select the class that exhibits the greatest similarity and the class that displays the least similarity to the current class:

$$y_j^{+} = \underset{i\in\{1,\ldots,K\}}{\operatorname{argmax}}\ \mathrm{S}_{\mathrm{ij}} \quad , \quad y_j^{-} = \underset{i\in\{1,\ldots,K\}}{\operatorname{argmin}}\ \mathrm{S}_{\mathrm{ij}}, \tag{3}$$

where K represents the total number of classes in the preceding task. Several studies [39, 43] employ *Heterogeneous Dropout* [1] to enhance the selection of underutilized features for subsequent tasks. While this method helps manage the overlap of feature selection across different tasks, it overlooks

Algorithm 1 Class-aware Frequency Domain Feature Selection Algorithm

Input: number of tasks T, training epochs of the t-th task K_t, dropout parameter λ and β_c, frequency dropout epochs $\mathcal{E}$
Initialize: $P_f = 1, P_s = 1$
for $t = 1, \ldots, T$ **do**
  for $e = 1, \ldots, K_t$ **do**
    if $e < \mathcal{E}$ **then**
      if $t > 1$ **then**
        Dropout features based on frequency dropout probabilities $1 - P_f$
        if $e == 1$ **then**
          Update P_f at the end of the first epoch (Eq. 4)
    else
      Dropout features based on semantic dropout probabilities $1 - P_s$
      Update P_s at the end of each epoch (Eq. 5)
    Select the top 60% of frequency domain features by response values for classification
    Update $\mathcal{F}$

similarities among classes. This oversight can negatively impact the effectiveness of selecting features in the frequency domain. To this end, we propose the *Frequency Dropout* method, which adjusts the probability of discarding frequency domain features based on class similarity. Specifically, let $[\mathcal{F}_y]_j$ denote the number of the j-th frequency domain feature when learning class y. The probability of selecting this feature in class c while learning a new task is expressed as follows:

$$[P_f]_{c,j} = \lambda \exp\left(\frac{-[\mathcal{F}_{y_c^-}]_j}{\max_i[\mathcal{F}_{y_c^-}]_i} \cdot \alpha_c^-\right) + (1-\lambda)\left(1 - \exp\left(\frac{-[\mathcal{F}_{y_c^+}]_j}{\max_i[\mathcal{F}_{y_c^+}]_i} \cdot \alpha_c^+\right)\right), \quad \alpha_c^- = \frac{\overline{S}_c}{S_{cy_c^-}} \quad , \quad \alpha_c^+ = \frac{S_{cy_c^+}}{\overline{S}_c}, \tag{4}$$

where $\overline{S}_c$ denotes the average cosine similarity between class c and all classes in previous tasks. The parameters α_c^- and α_c^+ control the intensity of the selection process. A higher value indicates a greater overlap of activated features with analogous classes and a reduced overlap with non-analogous classes. The coefficient λ serves as a weighting factor that adjusts the selection of frequency domain features, determining whether the emphasis is more towards similar classes or less towards dissimilar ones. The *Frequency Dropout* probability is updated at the beginning of each task. After completing training for $\mathcal{E}$ epochs on a given task, *Semantic Dropout* [39] is employed instead of *Frequency Dropout*. It encourages the model to use the same set of frequency domain features for classification by setting the retention probability of frequency domain features in each class. This probability is proportional to the number of times that frequency domain feature has been selected in that class so far:

$$[P_s]_{c,j} = 1 - \exp\left(\frac{-[\mathcal{F}_c]_j}{\max_i [\mathcal{F}_c]_i}\beta_c\right), \tag{5}$$

where β_c controls the strength of dropout. The probability of *Semantic Dropout* is updated at the end of each epoch, thereby enhancing the model's established frequency domain feature selection. This adjustment effectively regulates the extent of overlap in the utilization of frequency domain features. Algorithm 1 outlines the procedure for the CFFS.

4 Experiment

4.1 Experimental Setup

Datasets. We conduct comprehensive experimental analyses on extensively used public datasets, including Split CIFAR-10 (S-CIFAR-10) [6] and Split Tiny ImageNet (S-Tiny-ImageNet) [9]. The S-CIFAR-10 dataset is structured into five tasks, each encompassing two classes, while the S-Tiny-ImageNet dataset is divided into ten tasks, each comprising twenty classes. Additionally, the standard input image size for these datasets is 32×32 pixels.

Table 1: Comparison on different CL methods. CLFD consistently reduces the peak memory footprint of corresponding CL methods while simultaneously improving average accuracy. The highest results are marked in bold, and shadowed lines indicate the results from our framework.

Buffer	Method	S-CIFAR-10			S-Tiny-ImageNet		
		Class-IL	Task-IL	Mem	Class-IL	Task-IL	Mem
–	JOINT	92.20±0.15	98.31±0.12	-	59.99±0.19	82.04±0.10	-
	SGD	19.62±0.05	61.02±3.33	-	7.92±0.26	18.31±0.68	-
–	oEWC [41]	19.49±0.12	68.29±3.92	530MB	7.58±0.10	19.20±0.31	970MB
	SI [50]	19.48±0.17	68.05±5.91	573MB	6.58±0.31	36.32±0.13	1013MB
	LwF [30]	19.61±0.05	63.29±2.35	316MB	8.46±0.22	15.85±0.58	736MB
50	ER [38]	29.42±3.53	86.36±1.43	497MB	8.14±0.01	26.80±0.94	1333MB
	DER++ [6]	42.15±7.07	83.51±2.48	646MB	8.00±1.16	23.53±2.67	1889MB
	ER-ACE [7]	40.96±6.00	85.78±2.78	502MB	6.68±2.75	35.93±2.66	1314MB
	CLS-ER [4]	45.91±2.93	89.71±1.87	1016MB	11.09±11.52	40.76±9.17	3142MB
50	CLFD-ER	45.56±3.71	84.45±0.85	205MB	7.61±0.03	34.67±1.91	514MB
	CLFD-DER++	51.02±2.76	81.15±1.92	241MB	10.69±0.27	31.55±0.39	658MB
	CLFD-ER-ACE	**52.74**±1.91	87.13±0.41	204MB	10.71±2.91	38.05±11.98	514MB
	CLFD-CLS-ER	50.13±3.67	85.30±1.01	401MB	**12.61**±0.95	37.80±3.08	1032MB
125	ER [38]	38.49±1.68	89.12±0.92	497MB	8.30±0.01	34.82±6.82	1333MB
	DER++ [6]	53.09±3.43	88.34±1.05	646MB	11.29±0.19	32.92±2.01	1889MB
	ER-ACE [7]	56.12±2.12	90.49±0.58	502MB	11.09±3.86	41.85±3.46	1314MB
	CLS-ER [4]	53.57±2.73	90.75±2.76	1016MB	16.35±4.61	46.11±7.69	3142MB
125	CLFD-ER	55.76±1.85	88.29±0.16	205MB	8.89±0.07	42.40±0.83	514MB
	CLFD-DER++	58.81±0.29	84.76±0.66	241MB	15.42±0.37	40.94±1.30	658MB
	CLFD-ER-ACE	58.68±0.66	89.35±0.34	204MB	15.88±2.51	44.71±10.54	514MB
	CLFD-CLS-ER	**59.98**±1.38	87.09±0.43	401MB	**18.73**±0.91	49.75±2.01	1032MB

Evaluation metrics. We use the average accuracy on all tasks to evaluate the performance of the final model:

$$ACC_t = \frac{1}{t}\sum_{\tau=1}^{t} R_{t,\tau} \tag{6}$$

We denote the classification accuracy on the τ-th task after training on the t-th task as $R_{t,\tau}$. Moreover, we evaluate the training time, training FLOPs and peak memory footprint [46] to demonstrate the efficiency of each method. More experimental results can be found in Appendix F.

Baselines. We compare CLFD with several representative baseline methods, including three regularization-based methods: oEWC [41], SI [50] and LwF [30], as well as four rehearsal-based methods: ER [38], DER++ [6], ER-ACE [7] and CLS-ER [4]. In our evaluation, we incorporate two non-continual learning benchmarks: SGD as the lower bound and JOINT as the upper bound.

Implementation Details We expand the Mammoth CL repository in PyTorch [6]. For the S-CIFAR-10 and S-Tiny-ImageNet datasets, we utilize a standard ResNet18 [19] without pretraining as the baseline model, following the method outlined in DER++ [6]. All models are trained using the Stochastic Gradient Descent optimizer with a fixed batch size of 32. Additional details regarding other hyperparameters are detailed in Appendix D and E. For the S-Tiny-ImageNet dataset, models undergo training for 100 epochs, whereas for the S-CIFAR-10 dataset, training lasts for 50 epochs per task. In rehearsal-based methods, each training batch consists of an equal mix of new task samples and samples retrieved from the buffer. To ensure robustness, all experiments are conducted 10 times with different initializations, and the results are averaged across these runs.

4.2 Experimental Result

Table 1 presents a comparative analysis of the results on the S-CIFAR-10 and S-Tiny-ImageNet datasets, evaluated under Class-IL and Task-IL settings. The results elucidate that CLFD significantly

Table 2: **Ablation Study:** The Influence of systematically removing different components of CLFD-ER on model performance in S-CIFAR-10.

Frequency Domain Feature Encoder	Class-aware Frequency Domain Feature Selection	Class-IL	Task-IL
✗	✗	29.42±3.53	86.36±1.43
✓	✗	39.19±0.83	**88.01**±0.06
✗	✓	37.80±5.78	85.78±2.43
✓	✓	**45.56**±3.71	84.45±0.85

enhances the performance of various rehearsal-based CL methods. Specifically, the CLFD model has notably achieved SOTA accuracies across all buffer sizes in benchmark evaluations. By reducing the peak memory footprint by 2.4 ×, CLFD can augment the average accuracy of the ER method by up to 16.14%. Furthermore, when integrated with the SOTA method CLS-ER, CLFD can also increase its average accuracy by 6.41% and reduce its peak memory footprint by 2.5 ×. The superior performance of CLFD indicates that our proposed framework effectively mitigates *catastrophic forgetting* by improving the efficiency of storage resource utilization and minimizing interference among various frequency domain features. Moreover, the improvements implemented by CLFD across four different established rehearsal-based methods underscore its adaptability as a unified framework, highlighting its potential for integration with diverse CL methods.

4.3 Edge Device Results

We evaluate the acceleration performance of the CLFD utilizing the NVIDIA Ampere architecture GPU and Octa-core Arm CPU on the NVIDIA Jetson Orin NX 16GB platform. We measure the training time and accuracy of various methods using the S-CIFAR-10 dataset with a buffer size of 125. To expedite the training process, data augmentation techniques were omitted, resulting in accuracy results that vary from those reported in Table 1. Figure 4 illustrates the training time and average accuracy of various methods. When combined with various CL methods, CLFD significantly reduces training time while simultaneously enhancing model accuracy. By achieving approximately 2.4 × training acceleration, CLFD can attain the highest average accuracy of 47.64% when integrated with ER-ACE. This suggests that CLFD significantly enhances the efficiency of CL on edge devices by reducing the input feature map size.

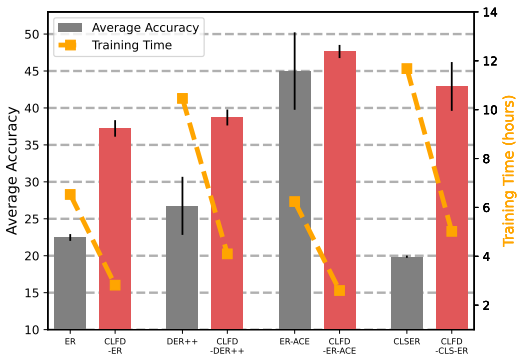

Figure 4: Comparison of different methods for S-CIFAR-10 dataset using Nvidia Jetson Orin NX with a buffer size of 125. When combined with various CL methods, CLFD significantly reduces training time while simultaneously enhancing model accuracy.

4.4 Ablation Study

In Table 2, we present a comprehensive ablation study of CLFD-ER employing a buffer size of 50 on the S-CIFAR-10 dataset. The results indicate that each component of our method makes a substantial contribution to enhancing accuracy. Comparing row 1 and 2, we can see that FFE enhances the utilization of storage resources by storing encoded features from the frequency domain rather than the original images. This method significantly improves the accuracy of rehearsal-based methods by optimizing the representation of stored data. Comparing rows 1 and 3, we can see that CFFS enhances the model's accuracy by mitigating the interference among frequency domain features across different tasks. CLFD achieves superior average accuracy by comprehensively integrating all components, thereby substantiating the efficacy of its individual elements.

Table 3: Comparison of CLFD and SparCL on S-CIFAR-10 dataset (Sparsity Ratio: 0.75, Buffer Size: 50).

Method	S-CIFAR-10		
	Class-IL(↑)	Task-IL(↑)	FLOPs Train $\times 10^{15}$(↓)
ER [38]	29.42±3.53	86.36±1.43	11.1
SparCL-ER [46]	43.74±2.91	85.01±3.86	2.0
CLFD-ER	45.56±3.71	84.45±0.85	2.8
CLFD-SparCL-ER	**55.15±0.89**	**88.52±0.29**	**0.6**

Table 4: Comparison of different frequency components.

Method	S-CIFAR-10	
	Class-IL(↑)	Task-IL(↑)
X_{ll}	49.59	**83.92**
X_{lh}	41.77	79.92
X_{hl}	44.45	82.76
X_{hh}	34.19	74.27
FFE	**51.02**	81.15

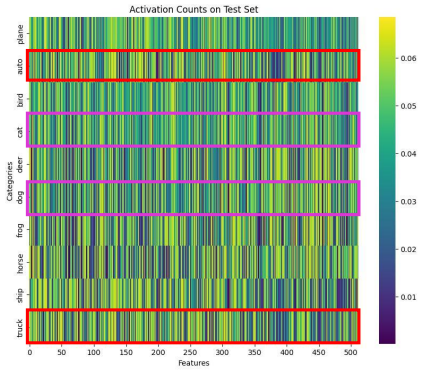

Figure 5: Frequency domain feature counts of the feature extractor trained on S-CIFAR10 with a buffer size of 125.

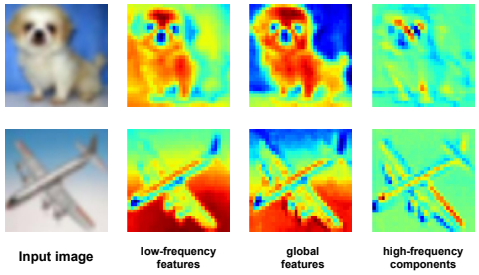

Figure 6: Visualization results of the FFE encoded image.

4.5 Model Analysis

In this section, we provide an in-depth analysis of CLFD.

Sparse Training We compare our method with SparCL [46], a SOTA sparse training method. Table 3 presents the average accuracy and training FLOPs for CLFD and SparCL. Our method significantly enhances the average accuracy compared to SparCL. Although our method incurs higher training FLOPs than SparCL, this does not directly correlate with actual training speed. Our method accelerates training speed without requiring additional optimization. Conversely, pruning and sparse training methods that utilize masks often fail to translate into actual training time savings without optimizations at the compiler level. Furthermore, the combination of CLFD and SparCL not only achieves the highest accuracy but also leads to the lowest training FLOPs. The successful integration of CLFD and SparCL serves as an example of CLFD's adaptability to sparse training and pruning methods.

The impact of different frequency components To explore the influence of different frequency components, we employ them as input feature maps and assess the CLFD-DER++ accuracy on the S-CIFAR-10 dataset under 50 buffer size. Table 4 presents the average accuracy across various frequency components. Utilizing the low-frequency components of images as input feature maps yields the highest accuracy among different frequency components. This suggests that CNN models demonstrate greater sensitivity to low-frequency channels compared to high-frequency channels. This finding aligns with the characteristics of the HVS, which also prioritizes low-frequency information. Despite this, FFE achieves the highest average accuracy, suggesting that high-frequency components are also significant. Optimal preservation of image information during downsampling is achieved only through the integration of components across different frequencies. Some examples of the encoded frequency domain feature maps are visualized in Figure 6.

Frequency Domain Feature Selection To evaluate the effectiveness of frequency dropout in reducing interference among frequency domain features across diverse tasks, we calculate the

selection of specific frequency domain features for different classes. Specifically, we track the classification activities on the test set and conduct normalized counts of selections for the frequency domain features, as illustrated in Figure 5. We observe that feature selection patterns exhibit higher correlations among semantically similar classes. For instance, the classes "cat" and "dog" often select identical sets of features. Similarly, significant similarities in feature selection patterns are evident between "auto" and "truck". This result demonstrates the effectiveness of CFFS.

5 Conclusion

Inspired by the human visual system, we propose CLFD, a comprehensive framework designed to enhance the efficiency of CL training and augment the precision of rehearsal-based methods. To effectively reduce the size of feature maps and optimize feature reuse while minimizing interference across various tasks, we propose the Frequency Domain Feature Encoder and the Class-aware Frequency Domain Feature Selection. The FFE employs wavelet transform to convert input images into the frequency domain. Meanwhile, the CFFS selectively uses different frequency domain features for classification depending on the frequency domain similarity of classes. Extensive experiments conducted across various benchmark datasets and environments have validated the effectiveness of our method, which enhances the accuracy of the SOTA method by up to 6.83%. Moreover, it achieves up to a 2.6× increase in training speed and a 3.0× reduction in peak memory usage. We discuss the limitations and broader impacts of our method in Appendix A and B, respectively.

Acknowledgement: This work was supported by the Chinese Academy of Sciences Project for Young Scientists in Basic Research (YSBR-107) and the Beijing Natural Science Foundation (4244098).

References

[1] Ali Abbasi et al. “Sparsity and heterogeneous dropout for continual learning in the null space of neural activations”. In: *Conference on Lifelong Learning Agents*. PMLR. 2022, pp. 617–628.

[2] Rahaf Aljundi et al. “Gradient based sample selection for online continual learning”. In: *Advances in neural information processing systems* 32 (2019).

[3] Rahaf Aljundi et al. “Memory aware synapses: Learning what (not) to forget”. In: *Proceedings of the European conference on computer vision*. 2018, pp. 139–154.

[4] Elahe Arani, Fahad Sarfraz, and Bahram Zonooz. “Learning fast, learning slow: A general continual learning method based on complementary learning system”. In: *International Conference on Learning Representations* (2022).

[5] Miles Brundage et al. “The malicious use of artificial intelligence: Forecasting, prevention, and mitigation”. In: *arXiv preprint arXiv:1802.07228* (2018).

[6] Pietro Buzzega et al. “Dark experience for general continual learning: a strong, simple baseline”. In: *Advances in neural information processing systems* 33 (2020), pp. 15920–15930.

[7] Lucas Caccia et al. “New insights on reducing abrupt representation change in online continual learning”. In: *International Conference on Learning Representations* (2022).

[8] Hyuntak Cha, Jaeho Lee, and Jinwoo Shin. “Co2l: Contrastive continual learning”. In: *Proceedings of the IEEE/CVF International conference on computer vision*. 2021, pp. 9516–9525.

[9] Arslan Chaudhry et al. “Continual learning with tiny episodic memories”. In: *Workshop on Multi-Task and Lifelong Reinforcement Learning*. 2019.

[10] Arslan Chaudhry et al. “Efficient lifelong learning with a-gem”. In: *International Conference on Learning Representations* (2019).

[11] Arslan Chaudhry et al. “Riemannian walk for incremental learning: Understanding forgetting and intransigence”. In: *Proceedings of the European conference on computer vision*. 2018, pp. 532–547.

[12] Arslan Chaudhry et al. “Using hindsight to anchor past knowledge in continual learning”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 35. 8. 2021, pp. 6993–7001.

[13] Tianshui Chen et al. “Learning a wavelet-like auto-encoder to accelerate deep neural networks”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1. 2018.

[14] Sayna Ebrahimi et al. “Adversarial continual learning”. In: *Proceedings of the European conference on computer vision*. Springer. 2020, pp. 386–402.

[15] Max Ehrlich and Larry S Davis. “Deep residual learning in the jpeg transform domain”. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 3484–3493.

[16] Utku Evci et al. “Rigging the lottery: Making all tickets winners”. In: *International conference on machine learning*. PMLR. 2020, pp. 2943–2952.

[17] Shin Fujieda, Kohei Takayama, and Toshiya Hachisuka. “Wavelet convolutional neural networks for texture classification”. In: *arXiv preprint arXiv:1707.07394* (2017).

[18] Lionel Gueguen et al. “Faster neural networks straight from jpeg”. In: *Advances in Neural Information Processing Systems* 31 (2018).

[19] Kaiming He et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[20] Huaibo Huang et al. “Wavelet-srnet: A wavelet-based cnn for multi-scale face super resolution”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 1689–1697.

[21] Jiaxing Huang et al. “Fsdr: Frequency space domain randomization for domain generalization”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 6891–6902.

[22] Libo Huang et al. “eTag: Class-Incremental Learning via Embedding Distillation and Task-Oriented Generation”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 38. 11. 2024, pp. 12591–12599.

[23] Libo Huang et al. “KFC: Knowledge Reconstruction and Feedback Consolidation Enable Efficient and Effective Continual Generative Learning”. In: *The Second Tiny Papers Track at ICLR 2024*. 2024.

[24] Zixuan Ke, Bing Liu, and Xingchang Huang. “Continual learning of a mixed sequence of similar and dissimilar tasks”. In: *Advances in neural information processing systems* 33 (2020), pp. 18493–18504.

[25] James Kirkpatrick et al. “Overcoming catastrophic forgetting in neural networks”. In: *Proceedings of the national academy of sciences* 114.13 (2017), pp. 3521–3526.

[26] Alex Krizhevsky, Geoffrey Hinton, et al. “Learning multiple layers of features from tiny images”. In: (2009).

[27] Namhoon Lee, Thalaiyasingam Ajanthan, and Philip HS Torr. “Snip: Single-shot network pruning based on connection sensitivity”. In: *International Conference on Learning Representations* (2019).

[28] A Levinskis. “Convolutional neural network feature reduction using wavelet transform”. In: *Elektronika ir Elektrotechnika* 19.3 (2013), pp. 61–64.

[29] Qiufu Li et al. “Wavelet integrated CNNs for noise-robust image classification”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 7245–7254.

[30] Zhizhong Li and Derek Hoiem. “Learning without forgetting”. In: *IEEE transactions on pattern analysis and machine intelligence* 40.12 (2017), pp. 2935–2947.

[31] Hangda Liu et al. “A resource-aware workload scheduling method for unbalanced GEMMs on GPUs”. In: *The Computer Journal* (2024), bxae110.

[32] Lin Liu et al. “Wavelet-based dual-branch network for image demoiréing”. In: *Proceedings of the European conference on computer vision*. Springer. 2020, pp. 86–102.

[33] Pengju Liu et al. “Multi-level wavelet-CNN for image restoration”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2018, pp. 773–782.

[34] Zhenhua Liu et al. “Frequency-domain dynamic pruning for convolutional neural networks”. In: *Advances in neural information processing systems* 31 (2018).

[35] Noel Loo, Siddharth Swaroop, and Richard E Turner. “Generalized variational continual learning”. In: *International Conference on Learning Representations* (2021).

[36] Michael McCloskey and Neal J Cohen. “Catastrophic interference in connectionist networks: The sequential learning problem”. In: *Psychology of learning and motivation*. Vol. 24. Elsevier, 1989, pp. 109–165.

[37] Lorenzo Pellegrini et al. “Continual learning at the edge: Real-time training on smartphone devices”. In: *arXiv preprint arXiv:2105.13127* (2021).

[38] Anthony Robins. “Catastrophic forgetting, rehearsal and pseudorehearsal”. In: *Connection Science* 7.2 (1995), pp. 123–146.

[39] Fahad Sarfraz, Elahe Arani, and Bahram Zonooz. “Sparse coding in a dual memory system for lifelong learning”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 37. 8. 2023, pp. 9714–9722.

[40] Fahad Sarfraz, Elahe Arani, and Bahram Zonooz. “Synergy between synaptic consolidation and experience replay for general continual learning”. In: *Conference on Lifelong Learning Agents*. PMLR. 2022, pp. 920–936.

[41] Jonathan Schwarz et al. “Progress & compress: A scalable framework for continual learning”. In: *International conference on machine learning*. PMLR. 2018, pp. 4528–4537.

[42] Joan Serra et al. “Overcoming catastrophic forgetting with hard attention to the task”. In: *International conference on machine learning*. PMLR. 2018, pp. 4548–4557.

[43] Preetha Vijayan et al. “TriRE: A Multi-Mechanism Learning Paradigm for Continual Knowledge Retention and Promotion”. In: *Advances in Neural Information Processing Systems* 36 (2024).

[44] Jeffrey S Vitter. “Random sampling with a reservoir”. In: *ACM Transactions on Mathematical Software (TOMS)* 11.1 (1985), pp. 37–57.

[45] Zifeng Wang et al. “Learn-prune-share for lifelong learning”. In: *2020 IEEE International Conference on Data Mining (ICDM)*. IEEE. 2020, pp. 641–650.

[46] Zifeng Wang et al. “Sparcl: Sparse continual learning on the edge”. In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 20366–20380.

[47] Travis Williams and Robert Li. “Wavelet pooling for convolutional neural networks”. In: *International conference on learning representations*. 2018.

[48] Kai Xu et al. “Learning in the frequency domain”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 1740–1749.

[49] Lu Yu et al. “Semantic drift compensation for class-incremental learning”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 6982–6991.

[50] Friedemann Zenke, Ben Poole, and Surya Ganguli. “Continual learning through synaptic intelligence”. In: *International conference on machine learning*. PMLR. 2017, pp. 3987–3995.

[51] Hanbin Zhao et al. “Mgsvf: Multi-grained slow versus fast framework for few-shot class-incremental learning”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 46.3 (2021), pp. 1576–1588.

[52] Tingting Zhao et al. “Deep bayesian unsupervised lifelong learning”. In: *Neural Networks* 149 (2022), pp. 95–106.

A Limitations

One limitation of our framework is its focus on optimizing rehearsal-based methods. While the use of a rehearsal buffer throughout the CL process is widely accepted, there exist scenarios where rehearsal buffers are prohibited. Nonetheless, by reducing the size of the input feature map, our framework has the potential to accelerate various CL methods, though the implications for model performance require further investigation. Furthermore, our analysis was limited to scenarios with finite rehearsal buffer sizes, whereas many current studies focus on scenarios with infinite rehearsal buffer sizes. This shift is primarily due to memory constraints being less significant compared to computational costs. We will also continue to investigate the performance of our method in scenarios involving infinite rehearsal buffer sizes.

B Broader Impacts

Inspired by HVS, we propose a novel framework called CLFD, which utilizes frequency domain features to enhance the performance and efficiency of CL training. Its success has opened up opportunities for enhancing existing CL methods in the frequency domain. CLFD contributes to the responsible and ethical deployment of artificial intelligence technologies by improving CL performance and efficiency. This is accomplished through the efficient ability of models to update and refine their knowledge without the need for extensive retraining. This further facilitates the real-world application of CL.

Although CLFD serves as a comprehensive framework aimed at improving the efficiency of various CL methods, we must remain cognizant of its potential negative societal impacts. While CLFD improves model stability, it does so by compromising the expense of model plasticity, resulting in reduced accuracy when applied to new tasks. This trade-off requires specific consideration in applications where accuracy is crucial, such as healthcare [46]. Furthermore, as a powerful tool for enhancing the efficiency of CL methods, CLFD could also strengthen models designed for malicious applications [5]. Therefore, it is recommended that the community devise additional regulations to mitigate the malicious use of artificial intelligence.

C Dataset Licensing Information

- CIFAR-10 [26] is licensed under the MIT license.
- The licensing information for Tiny-ImageNet is unavailable. However, the dataset is accessible to researchers for non-commercial purposes.

D Hyperparameter Selection

Table A1 presents the selected optimal hyperparameter combinations for each method in the main paper. The hyperparameters include the learning rate (lr), batch size (bs), and minibatch size (mbs) for rehearsal-based methods. Other symbols correspond to specific methods. It should be noted that the batch size and minibatch size are held constant at 32 for all CL benchmarks.

E Experiment Details

E.1 Experiment Platform

We conduct comprehensive experiments utilizing the NVIDIA GTX 2080Ti GPU paired with the Intel Xeon Gold 5217 CPU, as well as the NVIDIA Jetson Orin NX 16GB, boasting NVIDIA Ampere architecture GPU and Octa-core Arm CPU.

E.2 Implementation Details

We set $\lambda = 0.5, \beta_c = 2$ in equation (4) and equation (5). We also set $\mathcal{E}$ at a value of 0.4 of the training epochs. In CFFS, we only select 60% of the frequency domain features for classification. We utilize the code from DER++ [6]. We extend our gratitude to the authors for their support and for providing

Table A1: Hyperparameters selected for our experiments.

Method	*Buffer*	**Split Tiny ImageNet**	*Buffer*	**Split CIFAR-10**
SGD	-	*lr:* 0.03	-	*lr:* 0.1
oEWC	-	*lr:* 0.03 λ*:* 90 γ*:* 1.0	-	*lr:* 0.03 λ*:* 10 γ*:* 1.0
SI	-	*lr:* 0.03 c*:* 1.0 ξ*:* 0.9	-	*lr:* 0.03 c*:* 0.5 ξ*:* 1.0
LwF	-	*lr:* 0.01 α*:* 1 *T:* 2.0	-	*lr:* 0.03 α*:* 0.5 *T:* 2.0
ER	50	*lr:* 0.1 *epoch:* 100	50	*lr:* 0.1 *epoch:* 50
	125	*lr:* 0.03 *epoch:* 100	125	*lr:* 0.1 *epoch:* 50
CLFD-ER	50	*lr:* 0.1 *epoch:* 100	50	*lr:* 0.1 *epoch:* 50
	125	*lr:* 0.03 *epoch:* 100	125	*lr:* 0.1 *epoch:* 50
DER++	50	*lr:* 0.03 α*:* 0.1 β*:* 1.0 *epoch:* 100	50	*lr:* 0.03 α*:* 0.1 β*:* 0.5 *epoch:* 50
	125	*lr:* 0.03 α*:* 0.2 β*:* 0.5 *epoch:* 100	125	*lr:* 0.03 α*:* 0.2 β*:* 0.5 *epoch:* 50
CLFD-DER++	50	*lr:* 0.03 α*:* 0.1 β*:* 0.5 *epoch:* 100	50	*lr:* 0.03 α*:* 0.1 β*:* 0.5 *epoch:* 50
	125	*lr:* 0.03 α*:* 0.1 β*:* 0.5 *epoch:* 100	125	*lr:* 0.03 α*:* 0.2 β*:* 0.5 *epoch:* 50
ER-ACE	50	*lr:* 0.03 *epoch:* 50	50	*lr:* 0.03 *epoch:* 50
	125	*lr:* 0.03 *epoch:* 50	125	*lr:* 0.03 *epoch:* 50
CLFD-ER-ACE	50	*lr:* 0.03 *epoch:* 50	50	*lr:* 0.03 *epoch:* 50
	125	*lr:* 0.03 *epoch:* 50	125	*lr:* 0.03 *epoch:* 50
CLS-ER	50	*lr:* 0.1 r_S*:* 0.04 r_P*:* 0.08 α_S*:* 0.999 α_P*:* 0.999 λ*:* 0.1 *epoch:* 50	50	*lr:* 0.03 r_S*:* 0.05 r_P*:* 0.2 α_S*:* 0.999 α_P*:* 0.999 λ*:* 0.15 *epoch:* 50
	125	*lr:* 0.1 r_S*:* 0.05 r_P*:* 0.08 α_S*:* 0.999 α_P*:* 0.999 λ*:* 0.1 *epoch:* 50	125	*lr:* 0.03 r_S*:* 0.1 r_P*:* 0.9 α_S*:* 0.999 α_P*:* 0.999 λ*:* 0.15 *epoch:* 50
CLFD-CLS-ER	50	*lr:* 0.1 r_S*:* 0.08 r_P*:* 0.16 α_S*:* 0.999 α_P*:* 0.999 λ*:* 0.15 *epoch:* 50	50	*lr:* 0.03 r_S*:* 0.1 r_P*:* 0.5 α_S*:* 0.999 α_P*:* 0.999 λ*:* 0.15 *epoch:* 50
	125	*lr:* 0.1 r_S*:* 0.1 r_P*:* 0.16 α_S*:* 0.999 α_P*:* 0.999 λ*:* 0.15 *epoch:* 50	125	*lr:* 0.03 r_S*:* 0.2 r_P*:* 0.9 α_S*:* 0.999 α_P*:* 0.999 λ*:* 0.15 *epoch:* 50

the research community with the Mammoth framework, which facilitates a fair comparison of various CL methods under standardized experimental conditions. To ensure a fair comparison, we endeavor to closely align the experimental settings with those used in the Mammoth framework. However, we modified the data augmentation techniques within the Mammoth framework. The details of our data augmentation techniques are presented as follows.

E.3 Data Augmentation

In line with [6], we employ random crops and horizontal flips as data augmentation techniques for both examples from the current task and the replay buffer. To ensure uniformity in data augmentation between the original images and the input features encoded with the FFE, random cropping is restricted to even pixels only.

F Additional Experiment Results

F.1 Stability-Plasticity Trade-off

If the CL model can retain previously learned information, it is considered stable; if it can effectively acquire new information, it is considered plastic. To better understand how various methods balance

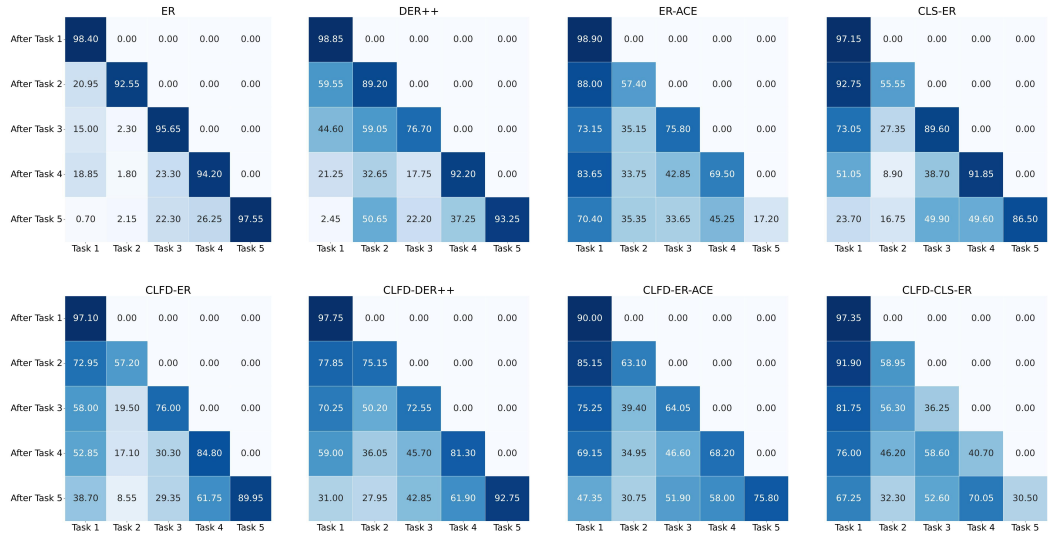

Figure 7: Performance of different methods by task. The heatmaps display the test set results for each task (x-axis) evaluated at the end of each sequential learning task (y-axis). We conducted experiments on the S-CIFAR-10 dataset using a buffer size of 50.

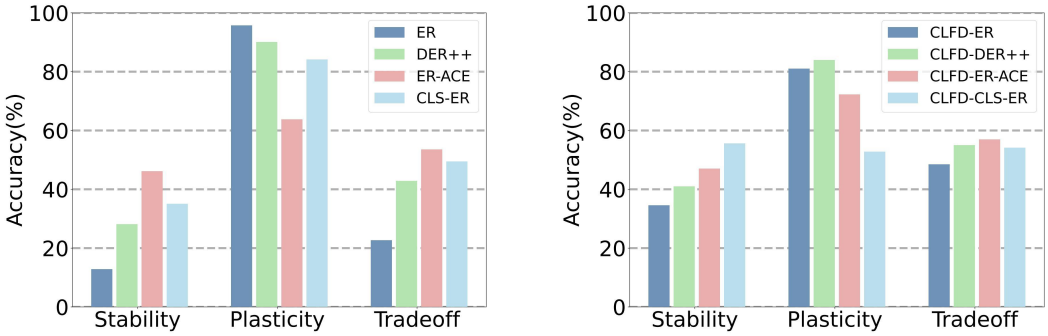

Figure 8: Stability-Plasticity Trade-off for CL models trained on S-CIFAR-10 under 50 buffer size.

stability and plasticity, we investigate the evolution of task performance as the model learns tasks sequentially. Figure 7 shows that CLFD consistently enhances the balance of all CL methods, delivering more uniform performance across all tasks. In addition, to further demonstrate the effectiveness of our method, we introduce a trade-off measure [40] that approximates how the model balances its stability and plasticity. Upon the model's completion of the final task T, its stability is evaluated by calculating the average performance across all preceding $T-1$ tasks as follows:

$$S = \frac{\sum_{\tau=1}^{T-1} R_{T,\tau}}{T-1} \tag{7}$$

The plasticity of the model (P) is evaluated by computing the average performance of each task after its initial learning i.e. the diagonal of the heatmap:

$$P = \frac{\sum_{\tau=1}^{T} R_{\tau,\tau}}{T} \tag{8}$$

Thus, the trade-off measure determines the optimal balance between the model's stability (S) and plasticity (P). This measure is calculated as the harmonic mean of S and P.

$$\textit{Trade-off} = \frac{2SP}{S+P} \tag{9}$$

Figure 8 provides the stability-plasticity trade-off measure for different CL methods. ER and DER++ exhibit high plasticity, enabling them to rapidly adapt to new information. However, they lack the

Table A2: Forgetting results on S-CIFAR-10 dataset.

Method	S-CIFAR-10			
	Class-IL(↓)		Task-IL(↓)	
Buffer Size	50	125	50	125
ER [38]	83.61±5.33	72.24±2.87	12.55±2.12	9.13±1.35
DER++ [6]	60.67±8.86	48.80±7.21	15.83±4.41	10.12±1.97
ER-ACE [7]	32.81±24.39	26.42±18.77	13.32±4.46	7.54±0.88
CLS-ER [4]	43.96±94.92	48.23±37.12	6.07±2.26	6.52±0.64
CLFD-ER	45.01±16.71	33.91±6.95	5.20±1.32	2.74±1.05
CLFD-DER++	41.93±4.65	32.82±1.06	12.62±3.98	9.70±0.94
CLFD-ER-ACE	24.98±6.34	**21.13**±1.45	4.92±2.19	**3.04**±0.35
CLFD-CLS-ER	**18.19**±2.19	22.20±2.90	**4.22**±3.09	3.82±0.54

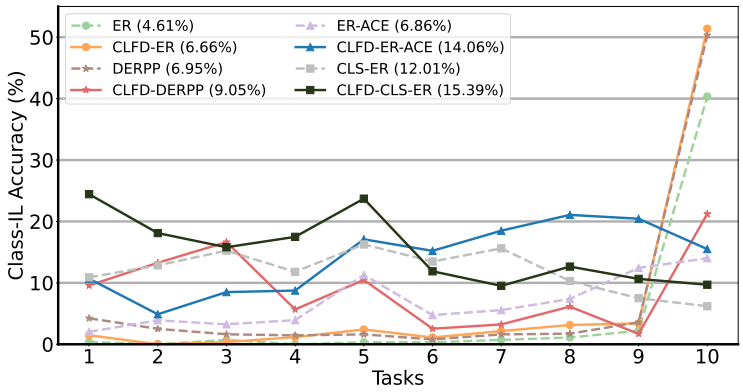

Figure 9: Comparison of Class-IL accuracy for different methods on the Split ImageNet-R dataset, divided into 10 tasks. The figure reports the accuracy of individual tasks at the end of CL training. The values in parentheses in the legend indicate the average accuracy.

ability to effectively retain previously acquired knowledge, leading to task recency bias issues. CLFD consistently improves the stability of CL methods, thereby reducing task reception bias and enhancing the balance between stability and plasticity.

F.2 Forgetting

We use the Final Forgetting (FF) to measure the model's anti-forgetting performance:

$$FF_t = \frac{1}{t-1}\sum_{j=1}^{t-1} \max_{i\in\{1,\dots,t-1\}} (R_{i,j} - R_{t,j}), \tag{10}$$

A smaller FF value indicates that the model exhibits less forgetting of previous knowledge, thereby demonstrating stronger anti-forgetting performance. Table A2 presents the FF result. Our method consistently reduces forgetting in CL methods, demonstrating the effectiveness of CLFD in preserving prior knowledge rather than solely focusing on improving the accuracy of subsequent tasks.

F.3 Split ImageNet-R Result

To further enhance the credibility of our results, we conduct additional experiments on ImageNet-R dataset. ImageNet-R, an extension of the ImageNet dataset, comprises 200 classes and a total of 30,000 images, with 20% designated for the test set. We divide ImageNet-R into ten tasks, each containing 20 classes. Input images are resized to 224 × 224 pixels. We test the Class-IL accuracy of each task with a buffer size of 500, maintaining consistent hyperparameters across all methods as those used in Split Tiny ImageNet. Figure 9 presents the experimental results, demonstrating that our method still significantly enhances the performance of various rehearsal-based CL methods,

Table A3: Task-IL and Class-IL accuracy under different feature selection proportions.

Feature Selection Proportions	CLFD-ER	CLFD-ER-ACE
10%	Task-IL: 85.67 Class-IL: 51.03	Task-IL: 85.06 Class-IL: 50.37
30%	Task-IL: 84.91 Class-IL: 48.91	Task-IL: 86.74 Class-IL: 50.64
50%	Task-IL: 83.88 Class-IL: 45.69	Task-IL: 87.05 Class-IL: 52.12
90%	Task-IL: 87.97 Class-IL: 39.58	Task-IL: 89.83 Class-IL: 54.20

even on more complex datasets. It is worth noting that the improvement in accuracy is not the sole advantage of our framework. By integrating our framework with rehearsal-based methods on the Split ImageNet-R dataset, training speed increased by up to 1.7×, and peak memory usage decreased by up to 2.5×. This demonstrates that our framework can significantly enhance the training efficiency of CL, thereby promoting its application on edge devices.

F.4 Feature Selection Proportions in CFFS

We conduct additional ablation experiments to investigate the impact of feature selection proportions on model performance. We focus on two methods: CLFD-ER and CLFD-ER-ACE, as Figure 8 shows that ER is the most plastic method, while ER-ACE is the most stable. We conduct tests on the S-CIFAR-10 dataset with a buffer size of 50. Table A3 presents the result. In the Task-IL setting, both CLFD-ER and CLFD-ER-ACE achieve higher accuracy with higher feature selection proportions, as the inclusion of more features enables the model to better distinguish between classes within the tasks. It is important to highlight that in CLFD-ER, lower feature selection proportions can also lead to positive outcomes. This phenomenon can be attributed to the high plasticity of the CLFD-ER method, where reducing feature selection proportions aids in mitigating potential interference among various tasks. In the Class-IL setting, CLFD-ER achieves higher accuracy with lower feature selection proportions, while CLFD-ER-ACE exhibits superior performance with higher feature selection proportions. This observation suggests that for methods with high plasticity, we need to decrease the feature selection proportions to mitigate the overlap of frequency domain features, thereby minimizing the impact of new tasks on old tasks. Conversely, for methods with high stability, increasing the feature selection proportions allows us to utilize more frequency domain features to learn the current classes effectively.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims align with the experimental results. Based on the various experiments detailed in our paper, it is expected that our method can be generalized.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: See Appendix A.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: Our paper is based on experimental results, without any inclusion of theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: See Section 4.1 and Appendix E.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide code to reproduce the primary experimental results.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: See Section 4.1, Appendix D and Appendix E. In addition, we provide code to reproduce the primary experimental results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: See Table 1, Table 2, Table 3, Table A2 and Figure 4

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: See Section 4.1 and Appendix E.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper conforms with the NeurIPS Code of Ethics standards in all respects.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: See Appendix B

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We mentioned and cited the datasets (S-CIFAR-10 and S-Tiny-ImageNet), as well as all comparative methods with their respective papers. The licenses for the datasets and models used are provided in the cited references and explicitly stated in Appendix C.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We provide the code for our proposed method in the supplement.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper is not about research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.