FREQUENCY-AWARE NETWORK FOR ASYNCHRONOUS FORGETTING IN CONTINUAL LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

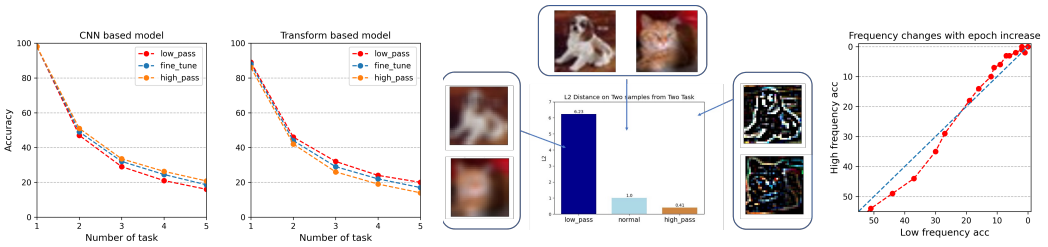
As a challenging problem, continual learning aims to avoid forgetting old knowledge as much as possible when the old model learns new tasks. Most current algorithms perform the same processing on each pixel of an input image. However, they ignore the phenomenon that the neural networks have asynchronous forgetting for pixels of different frequencies when learning a series of tasks. Just as people have different memory abilities for the details and the whole of the image, neural networks also have different memory abilities for high and low-frequency parts when learning images, resulting in asynchronous forgetting. This paper exploits this phenomenon for better network architecture design and employs a knowledge consolidation strategy for features learned by different modules. In terms of network architecture, we design a dual-stream network with high and low frequencies separated, using characteristics of convolutional neural network and transformer based network to process the high-frequency and low-frequency information of the image, respectively. In the aspect of knowledge consolidation, we design a dynamic distillation loss function, which dynamically adjusts the consolidation weight of high-frequency and low-frequency information according to the training process of the network. We verify the effectiveness of our method through a series of experiments.

1 INTRODUCTION

When using a trained model to learn a new task, the accuracy of previous tasks will decrease significantly. This is a phenomenon called catastrophic forgetting. Continual learning (CL) enables humans to acquire novel experience continually while maintaining existing knowledge. In a dynamic and open environment, it is critical for modern artificial intelligence to have the ability of CL because data distribution in real-world applications usually changes. Motivated by this, plenty of works Abdelsalam et al. (2021); Wu et al. (2021); Parisi et al. (2019) have emerged recently to alleviate the catastrophic forgetting problem. However, most of the current algorithms process each pixel of the image equally through the same network architecture. Just as humans have different abilities of remembering the details of an image and overall contours of an image, for example, you may forget the details of a painting you saw yesterday, but you still remember its overall composition. Neural networks have a similar mechanism. From the signal processing point of view, the local details correspond to the high-frequency information of the image, and the overall contour corresponds to the low-frequency information of the image. Thus, neural networks have different memory abilities of retaining the knowledge of different frequency parts of an image. When learning a new task, the neural network has asynchronous forgetting for different frequency information of the learned image. This asynchrony is reflected in two aspects. One is the final result, that is, after learning the new task, the network shows different memory ability for high and low frequency information. The other is the learning process, that is, in the process of learning new tasks, the speed of forgetting high and low frequency information is also different. This frequency-related asynchronous forgetting phenomenon has also been mentioned in recent literature Zhao et al. (2022) .

So, how to take advantage of this out-of-sync phenomenon to alleviate the catastrophic forgetting problem? We make improvements in terms of network architecture design and knowledge consolidation. For the network architecture design, we adopt two different architectures to process the high frequency and low frequency of images respectively. The current mature network architectures include convolutional-neural-network-based (CNN-based) architectures and transformer-based ar-

chitectures. According to the existing literature, CNN and transformer have different effects on different frequencies of images Bai et al. (2022)Park & Kim (2022). CNN is easier to capture the high frequency features due to its local receptive fields, and transformer is easier to capture the low frequency features due to its long-term correlation. Therefore, we design a dual-stream network, using CNN and transformer to process high-frequency information and low-frequency information of input images respectively. For the knowledge consolidation, distillation Li & Hoiem (2017) is commonly adopted to keep the knowledge of the old tasks. It regards the trained model as teacher and distills previous knowledge to maintain unchanged responses of the network on the old tasks. We re-frame the traditional distillation operation and apply different weights to the high-frequency and low-frequency parts at different stages of training. The above analysis is based on intuition.



(a) (b) (c)

Figure 1: (a) Test the change of model accuracy of different architectures after high and low frequency filtering on CIFAR10 dataset. (b) L2 distance of the two photos in the new and old tasks after high and low frequency filtering, based on the unfiltered distance. (c) With the increase of epoch, the accuracy of high-frequency information and low-frequency information changes.

Next, we will verify our hypothesis through experiments. In this paper, we consider a challenging scenario called class-incremental learning, in which each task in the sequence contains a set of classes disjoint from other tasks. The model needs to learn a single classifier built for all classes seen so far and can classify all classes seen at different stages without task-id provided.Rebuffi et al. (2017)

Figure 1 (a) shows the accuracy curves of CNN-based model and transformer-based model tested on CIFAR10 dataset processing with different frequency filtering. From this, we can learn that the forgetting ability of the high-frequency and low-frequency parts of the image is related to the architecture. In the left panel of Figure 1 (a), the ability of CNN-based model to preserve high-frequency information is better than that of low-frequency information, while the conclusion in the right panel is opposite, the transformer-based model preserves low-frequency information better as when the number of tasks increases. The accuracy gap between the two frequency parts is growing with the progress of the task, which motivates us to adopt different architectures for different frequency parts of the image for CL. At the same time, the difference between high and low frequencies of images will also bring trouble to CL. we select two samples from the new task and the old task respectively and calculate their L2 distance as a baseline. Then we test the L2 distance between the two images after high-frequency and low-frequency filtering. Figure 1 (b) shows the difference. Compared with the baseline distance, the difference between the low-frequency information of the image is larger, while the difference between the high-frequency information is smaller. When the model is trained, low-frequency components and high-frequency components are learned together. The impact of these differences can be eliminated to a certain extent through the training of balanced datasets. But for the CL setting, the number of old and new categories is often unbalanced. So the factor of inconsistent frequency difference have an impact on the model. Therefore, CL tasks exacerbate the effects of this frequency inconsistency.

In Figure 1 (c), we test the accuracy of high-frequency information and low-frequency information changes with the increase of epochs. It can be found that during the learning process, the forgetting speed of different frequency of the image is not synchronized. In the initial stage of model training, the low-frequency part of the image is forgotten more. As the training progresses, the forgetting of the high-frequency part begins to increase. This inspires us to improve the traditional distillation regularization method. In the initial stage, a larger constraint weight can be taken for the low-frequency paarts, and then the constraint weight on the low-frequency features can be gradually

reduced in the subsequent training epochs. The constraint weight for the high-frequency part is the opposite.

After the above analysis, we first explain the existence of asynchronous forgetting in different parts of the image. According to this feature, we deal with it from the perspective of network architecture design and knowledge consolidation respectively. We demonstrate the effectiveness of our method through the conventional CL experimental setup.

The main contributions are as follows :

- We analyze the phenomenon of asynchronous forgetting in continual learning.
- From the perspective of model architecture design, we propose a dual-stream network architecture to process high and low frequency information respectively.
- From the perspective of knowledge consolidation of old tasks, we propose a dynamic distillation loss function, which can dynamically adjust the consolidation weight of different frequency parts along with the model learning process.

2 Related Work

Class incremental learning addresses the setting where training data is arriving sequentially, and data from previous classes is discarded when data for new classes becomes available. Recent literature proposed various approaches to tackle this issue. The following section will give a brief introduction.

2.1 Continual Learning Methods

The current continual learning (CL) methods are mainly divided into three categories: regularization methods; experience replay methods; dynamic network methods. Regularization techniques force constraints on the update of network parameters to mitigate catastrophic forgetting. This is done by incorporating additional penalty terms into the loss function. Knowledge distillation is also used in CL, which can be regarded as a regularization method. Distillation can be used to hinder catastrophic forgetting by appointing a previous snapshot of the model as teacher and distilling from it while new tasks are learned. Dynamic network methods dynamically expand the network in learning each new task Yan et al. (2021). Experience replay methods aim to find ways to use previous task datasets, which can be further divided into two approaches: directly using past task data and using pseudo-data generation techniques Boschini et al. (2022). Experience replay methods often have good performance but they tend to store a large amount of past data and face the problem of sample imbalance, which makes the model tend to predict the category of new tasks. So the current research methods often use the past samples to replay and combine the regularization method to maintain the old knowledge. For example, the Incremental Classifier and Representation Learning (iCaRL) algorithm Rebuffi et al. (2017) selects samples based on their corresponding feature space representation and extracts the representation of all samples and calculates the average for each category. This method iteratively selects samples for each category. At each step, a sample is selected when it is added to its category, the sample mean obtained is closest to the true class mean. In this article, we combine experience replay and distillation to keep the model's memory of old tasks. We separate the high and low frequency parts of the image and use different weights to distill knowledge to alleviate the asynchronous forgetting. Zhao et al. Zhao et al. (2022) also points out that the forgetting of the network is related to frequency. Furthermore, we point out that in the training process of the network, the forgetting of high and low frequencies is also asynchronous.

2.2 Vision Transformers

Self-attention based transformer architecture has revolutionized from natural language processing (NLP). Vision transformer (ViT) has caused a lot of discussion in the field of computer vision recently. Owing to long range association of different patches can be established, ViTs have shown promising results in image classification, object detection, and segmentation to name a few Dosovitskiy et al. (2021). Recently, there are many excellent network architectures, including data-efficient image transformer (DeiT) Touvron et al. (2021) which uses knowledge distillation from a CNN through a distillation token, cross-covariance image transformer (XCiT) El-Nouby et al. (2021)

which performs self-attention across feature channels to counter the quadratic complexity associated with self-attention between tokens. Recently, the network structure that comprehensively uses the advantages of CNN and transformer has also emerged. Swin transformer Liu et al. (2021) and nested hierarchical transformer (NesT) Zhang et al. (2022) are among the popular hybrid ViTs. In this article, we use the advantages of CNN and transformer to build our dual stream network, inspired by the current studies, such as Conformer Peng et al. (2021), CMT Guo et al. (2022),ViTAE Dai et al. (2021),CoAtNet Xu et al. (2021).

3 METHOD

In this section, we first introduce the infrastructure of our proposed frequency-aware dual-stream network, which uses CNN and ViT to process high-frequency and low-frequency information respectively. Then, we introduce our proposed dynamic distillation loss function to reduce catastrophic forgetting. The overall framework is shown in Figure 2.

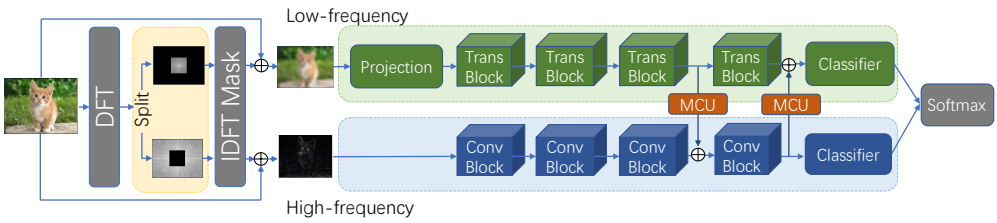


Figure 2: The overall framework of the proposed frequency-aware dual-stream CL network. An image is first processed by Discrete Fourier transform (DFT) and thus decomposed into high-frequency and low-frequency parts. Then the two parts go through inverse discrete Fourier transform (IDFT) to generate a mask for the original image. The original photo is processed by masks representing high and low frequencies, and then enter into the subsequent modules. Among them, the module after low-frequency processing is processed by a transformer-based structure, and the module after high-frequency processing is processed by a CNN-based structure. The two modules interact in the deep layer of the network to fuse features of different frequencies. Finally, the outputs of two branches are added and pass through the softmax layer to get the final result.

3.1 PROBLEM FORMULATION

In this work, we follow the learning protocol for image classification from Wang et al. (2018). More specifically, we consider a training set $D = (D_1, ..., D_T)$ consisting of T tasks where the dataset for the t-th task $D_t = (x_i^t, y_i^t)_{i=1}^{n_t}$ contains n_t input-target pairs (x_i^t, y_i^t). When the tasks arrive sequentially and exclusively, we assume the input-target pairs(x_i^t, y_i^t) in each task are independent and identically distributed (i.i.d.). The goal is to learn a supervised model $f_\theta : X \to Y$ parametrized by θ that outputs a class label y given an unseen image x.

3.2 FREQUENCY-AWARE DUAL-STREAM NETWORK

In this section, we introduce the dual-stream framework of frequency sensing. In a standard CL task, all pixels share the same filter. However, in dealing with the forgetting problem of neural networks, pixels should not be treated equally. As shown in Figure 1, network's ability to maintain knowledge for high-frequency pixels and low-frequency pixels are not synchronized. Based on this conclusion, we deal with high-frequency and low-frequency pixels separately. Considering that the network architecture design has different preferences for high-frequency and low-frequency information, we use different filters for high-frequency and low-frequency information. During the network training phase, each pixel is assigned to a branch of the network according to a frequency-aware mask M. Notating the branch number as N and the function of branch n as f_n, the input tensor as X and the output tensor as Y.

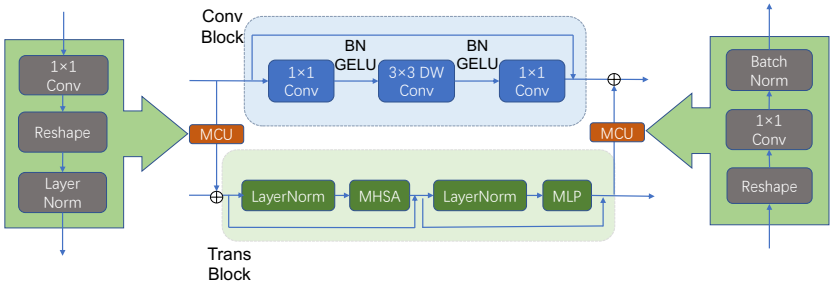


Figure 3: Network architecture of CNN branch and transformer branch.

$$
\begin{aligned}
&Y_i = \sum_{n=1}^{N} f_n \cdot M_{i,n}(X_i), \\
&s.t.\ \ M_{i,n} \in \{0,1\} \ \textit{and} \ \sum_{n}^{N} M_{i,n} = 1.
\end{aligned}
\tag{1}
$$

where i is the pixel index. The constrains mean that only one branch n where $M_{i,n} = 1$ will be chosen for X_i. It is very important to assign pixels to the right branch. Discrete Fourier transform (DFT) is used to generate a frequency mask. Specifically, DFT is performed on the image to convert the image from the spatial domain to the frequency domain. We set a threshold S to divide the frequency into high frequency and low frequency parts. Then, the two parts are converted back to the spatial domain through inverse discrete Fourier transform (IDFT) to generate a frequency mask. As shown in Figure 3, after generating the frequency mask, the high frequency masked branch passes through the CNN-based network, and the low frequency masked branch passes through the transformer-based network. For a mask $M \in \{0,1\}$, the low-pass filtering M_l^S and high-pass filtering M_h^S with the filter size S are formally defined as, H and W means photo hight and wide:

$$
M_l^S(\boldsymbol{X}) = \mathcal{F}^{-1}(\boldsymbol{m} \odot \mathcal{F}(\boldsymbol{X})), \ \text{where} \ \boldsymbol{m}_{i,j} = \begin{cases} 1, \ \text{if} \ \min(|i - \frac{H}{2}|, |j - \frac{W}{2}|) \leqslant \frac{S}{2} \\ 0, \ \text{otherwise} \end{cases}, \tag{2}
$$

$$
M_h^S(\boldsymbol{X}) = \mathcal{F}^{-1}(\boldsymbol{m} \odot \mathcal{F}(\boldsymbol{X})), \text{where} \ \boldsymbol{m}_{i,j} = \begin{cases} 0, \ \text{if} \ \min(|i - \frac{H}{2}|, |j - \frac{W}{2}|) \leqslant \frac{\min(H,W) - S}{2} \\ 1, \ \text{otherwise} \end{cases}, \tag{3}
$$

The construction of specific CNN and transformer modules is shown in Figure 3.

- CNN Branch: As shown in Figure 3, the CNN branch adopts feature pyramid structure. Following the definition in residual network (ResNet). We split the whole branch into 4 stages. Each stage is composed of multiple convolution blocks. In each block, we borrow the residual structure, 1×1 convolution kernel and 3×3 depth wise convolution block and 1×1 convolution kernel. Compared with the standard ResNet network, we halve the number of blocks to reduce the model size.
- Transformer Branch: Following ViT , this branch contains N repeated transformer blocks. As shown in Figure 3, each transformer block consists of a multi-head self-attention module and a multilayer perceptron (MLP) block. LayerNorms are applied before each layer and residual connections in both the self-attention layer and MLP block.

To enable the two modules to interact, we build a mapping bridge called MCU between the two modules to align the outputs of the two modules, this setting follow Peng et al. (2021). We fuse the two modules only in the last two stages of the network.

3.3 Dynamic Distillation Loss Function

In CL tasks, regularization constraints are often used to preserve information about old tasks. Previous analyses have shown that high-frequency and low-frequency information of images tend to have

different forgetting abilities. At the same time, the forgetting speed of high-frequency content and low-frequency content is also out of sync. To overcome this problem, we adopt a dynamic distillation loss function, which includes two aspects. First, we decouple the high-frequency information and low-frequency information according to their different memory capabilities, and adopt different weight coefficients, rather than treat them as a whole. Secondly, we solve the asynchronous forgetting problem of high-frequency and low-frequency information. That is, low-frequency information is forgotten more at the beginning. With the deepening of online learning, the forgetting speed of high-frequency part gradually accelerates. We use dynamic weighting coefficients for the high-frequency and low-frequency parts.

$$\mathcal{L} = \mathcal{L}_{new} + \alpha_t \cdot \mathcal{L}_{low} + (1 - \alpha_t) \cdot \mathcal{L}_{high} \tag{4}$$

where the weight parameter α_t is a parameter that changes with the number of training epochs of the network. Here we take the cosine change rate, and the size is changed from 0.8 to 0.2. $\mathcal{L}_{new}$ is the loss function of cross entropy loss function on new tasks, $\mathcal{L}_{low}$ and $\mathcal{L}_{high}$ is distillation loss of the model in the high-frequency part and low-frequency part, respectively, $\mathcal{L}_{high}$ conventional distillation loss function based on resnet, $\mathcal{L}_{low}$ as shown in the Figure 6 .

4 EXPERIMENTS

In this section, involved datasets, evaluation metrics, and the implementation details will be introduced in detail. Then, we will present several state-of-the-art competitors as well as the experimental results of our method. Finally, the ablation studies will prove the effectiveness of our proposed approach.

4.1 DATASETS

We validate our results on three datasets, respectively on CIFAR-100 and ImageNet-100.

- The CIFAR-10 dataset: composed of 60k 32×32 RGB images of 10 classes, with 6000 images per class. Every class has 5000 images for training and 1000 images for testing.
- The CIFAR-100 dataset: composed of 60k 32×32 RGB images of 100 classes, with 600 images per class. Every class has 500 images for training and 100 images for testing.
- The ImageNet-100 dataset: This is a subset of 100 classes from ImageNet. Image size 224×224. Each class contains 1300 samples for training and 50 samples for testing. We split into 10 disjoint tasks, where each task contains 10 classes.

4.2 DETAILS

For each task, we use the AdamW optimizer. Each task trains 200 epochs, the initial learning rate is set to 0.05, the weight decay factor is 0.001 in a cosine schedule, and the batch size is set to 512. Each dataset we set memory buffer size 2000. When training transformer based network, we choose patch size 2 in CIFAR100 and patch size 16 in ImageNet-100. During the training phase, we follow the data augmentation techniques in DeiT Touvron et al. (2021). These techniques include Mixup, CutMix, Erasing, RandAugment and Stochastic Depth. Following UCIR, PODNet, and DER, at the end of each task (except the first) we finetune our model for 20 epochs with a learning rate of $5e^{-5}$ on a balanced dataset. We made a comparison with the methods mentioned in baselines respectively.

4.3 EVALUATION METRICS

We evaluate the performance of the method by average accuracy. We construct CL tasks by dividing the given data set. For example, given the data set CIFAR100, set the number of learning tasks in the incremental phase $S = 10$. This means that total of five tasks are learned, and each task learns ten categories. At the completion of each task learning, the performance on all learning categories in the past is tested as the final result. Set the test data set as $D_{0:i}^{test}$, where $0 : i$ denote all seen classes

Methods	10Steps		20Steps		50Steps	
	#Paras	Avg	#Paras	Avg	#Paras	Avg
Bound	11.22	80.41	11.22	81.49	11.22	81.74
iCaRL	11.22	$65.27_{\pm 1.02}$	11.22	$61.20_{\pm 0.83}$	11.22	$56.08_{\pm 0.83}$
UCIR	11.22	$58.66_{\pm 0.71}$	11.22	$58.17_{\pm 0.30}$	11.22	$56.86_{\pm 0.83}$
BIC	11.22	$68.80_{\pm 1.20}$	11.22	$66.48_{\pm 0.32}$	11.22	$62.09_{\pm 0.85}$
WA	11.22	$69.46_{\pm 0.29}$	11.22	$67.33_{\pm 0.15}$	11.22	$64.32_{\pm 0.28}$
PODNet	11.22	$58.03_{\pm 1.27}$	11.22	$53.97_{\pm 0.85}$	11.22	$51.19_{\pm 1.02}$
DER	112.27	$75.36_{\pm 0.36}$	224.55	$74.09_{\pm 0.33}$	561.39	$72.41_{\pm 0.36}$
DyTox	10.73	$74.10_{\pm 0.10}$	10.74	$71.62_{\pm 0.11}$	10.77	$68.90_{\pm 0.05}$
Ours	12.93	$\mathbf{75.42}_{\pm 0.42}$	12.93	$72.68_{\pm 0.78}$	12.93	$69.36_{\pm 0.76}$

Table 1: Results on CIFAR100 (average over 3 runs). *#Paras* means the average number of parameters used during inference over steps, which is counted by million. *Avg* means the average accuracy (%) over steps. .

so far. The average accuracy is reported as the final evaluation.

$$ACC = \frac{1}{T}\sum_{i=1}^{T} R_{T,i},\ FGT = \frac{1}{T-1}\sum_{i=1}^{T-1} F_i \tag{5}$$

where $R_{T,i}$ denotes the test accuracy on task i after the model has finished task T, $F_i = \max_{t\in 1,\ldots,T-1}(R_{t,i} - R_{T,i})$ denotes the forgetting on task i, some articles use $BWT = \frac{1}{T-1}\sum_{j=1}^{T-1}(R_{T,i} - R_{i,i})$.

4.4 BASELINES

We compare our proposed method against several state-of-the-art continual learning algorithms:

- EWC Chaudhry et al. (2019):techniques force constraints on the update of network parameters to mitigate catastrophic forgetting.
- iCaRL Rebuffi et al. (2017): using herding method to get previous data and distillation to avoid catastrophic forgetting.
- UCIR Shim et al. (2021): uses cosine classifier and euclidean distance between the final flattened features as a distillation loss.
- RPSNet Rajasegaran et al. (2019):uses random path selection network for incremental learning.
- WA Yang & Xu (2020): uses a knowledge distillation loss and re-weights at each epoch the classifier weights associated to new classes so that they have the same average norm as the classifier weights of the old classes.
- DyTox Douillard et al. (2022): transformers for continual learning with dynamic token expansion.
- PODnet Douillard et al. (2020): uses a cosine classifier and a specific distillation loss (POD) applied at multiple intermediary features of the ResNet backbone.
- DER Yan et al. (2021): a novel two-stage learning approach that utilizes a dynamically expandable representation for more effective incremental concept modeling.
- Fine-tuning: simply trains the model in the order the data is presented without any specific method for forgetting. avoidance.
- Joint learning: considers training the model online on an i.i.d. data stream and can be regarded as the upper bound. performance.

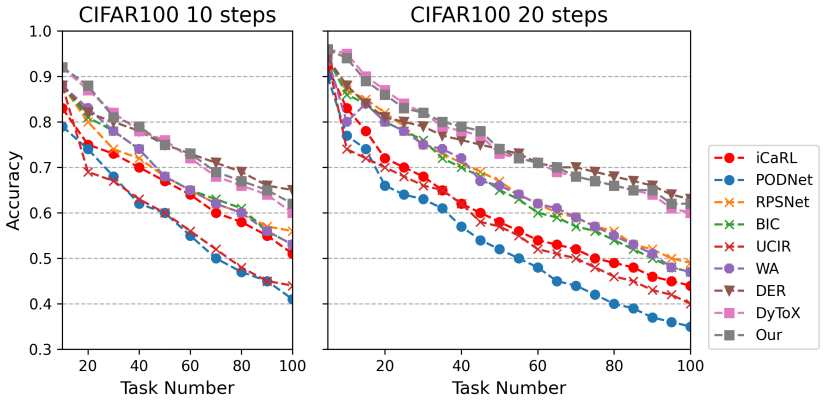


Figure 4: The evolution of performance as a function of the number of tasks CIFAR-100 (10 steps) and CIFAR-100 (20 steps). In the 0-th phase, θ_{base} is trained from scratch, the remaining classes are given evenly in the subsequent phases.

4.5 Comparative Performance Evaluation

Table 1 shows the overall experimental results, and Fig.4 shows the trend of accuracy with the number of tasks. From the two charts, we can draw the following conclusion: in most cases, the effect of our model is significantly better than the baseline experiment, and the parameters of the model do not increase much. This shows that our algorithm has a good effect on retaining past knowledge and learning new tasks. In addition, our method has achieved good gains in the comparison results in various scenarios, which shows that our method is not limited to some specific CL scenarios. From the curve change, we can see that our method performs well in most stage, and our curve is smooth. In the whole CL process, our method learns good feature expression. Compared with DER, our method is less effective, but our parameters are greatly reduced.

Methods	**10Steps**	
	#Paras	**Avg**
Bound	11.2	95.10
E2E	11.2	89.92
iCaRL	11.2	83.60
BIC	11.2	90.60
WA	11.2	91.00
RPSNet	–	87.90
DER	112.3	93.23
DyTox	11.0	92.04
Ours	13.2	**93.31**

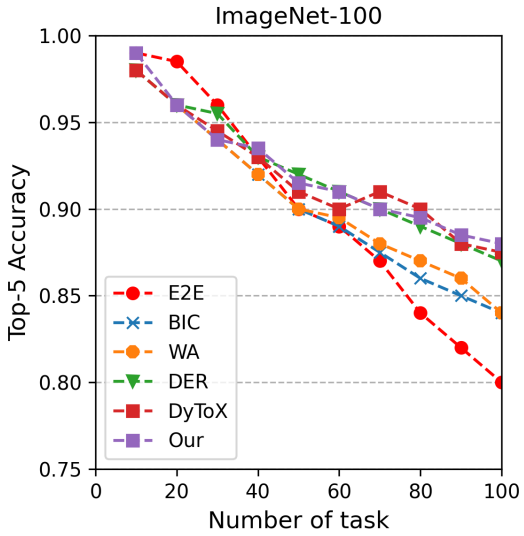


Figure 5: Results on ImageNet100. The accuracy reports top-5 acc.

4.6 Ablation Study

We further investigate our model performance with an ablation study and summarize it in Table 2. We conduct experiments on the CIFAR10 and CIFAR100 dataset with 10 steps setting. In this table,

Table 2: Ablation Study of our method

	Method	+ DA	+ DD	+ FA	CIFAR10	CIFAR100
ACC	Two CNN	✗	✓	✓	73.31	72.92
	Two Transformer	✗	✓	✓	56.28	48.67
	No Frequency Aware	✓	✓	✗	74.34	73.26
	No Dynamic Distillation	✓	✗	✓	75.41	74.12
FGT	No Dynamic Distillation	✓	✗	✓	21.68	23.41
	No Frequency Aware	✓	✓	✗	32.54	31.62

DA means Different architectures, DD means Dynamic Distillation, FA means Frequency Aware. These results shows the effect of the comparative experiment to analyze the function of each module.
Network architecture design In order to verify the necessity of our network structure design, we design an experiment using CNN based dual network and transformer based dual network. The two networks also conduct feature fusion in the later stage of the model. Through the experiment, we can find that the structure of the same network can not be used to learn effective expression. Among them, the performance of the transformer-based dual network is worse, because of the amount of data, the transformer based network is difficult to learn.

Necessity of frequency decoupling For the necessity of frequency decoupling, we input two branches of the dual flow network into the original samples. In this case, because the high and low frequency information cannot be decoupled, we cannot separate them according to the high and low frequency conditions in the feature retention stage.

Necessity of dynamic distillation In order to verify the necessity of dynamic loss function, we designed a static loss function as a contrast experiment. In the training phase of the network, the distillation weight corresponding to the high-frequency part and low-frequency part does not change. It can be found that because the change of the solidified weight of knowledge cannot correspond to the learning process of the model, its effect is lost.

4.7 Conclusion

In this article, we explore the asynchronous problem of forgetting in CL. We propose a dual stream processing architecture to process high-frequency and low-frequency information respectively, and adopt a dynamic distillation loss function for the asynchronous problem. We verify our effect through experiments. For the future, it is expected to study the asynchronous problem from more perspectives.

References

Mohamed Abdelsalam, Mojtaba Faramarzi, Shagun Sodhani, and Sarath Chandar. Iirc: Incremental implicitly-refined classification. *arXiv preprint*, 2021.

Jiawang Bai, Li Yuan, Shu-Tao Xia, Shuicheng Yan, Zhifeng Li, and Wei Liu. Improving vision transformers by revisiting high-frequency components. *European Conference on Computer Vision*, 2022.

Matteo Boschini, Lorenzo Bonicelli, Pietro Buzzega, Angelo Porrello, and Simone Calderara. Class-incremental continual learning into the extended der-verse. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-GEM. In *ICLR*, 2019. URL `https://openreview.net/forum?id=Hkf2_sC5FX`.

Zihang Dai, Hanxiao Liu, Quoc V. Le, and Mingxing Tan. Coatnet: Marrying convolution and attention for all data sizes. In *Advances in Neural Information Processing Systems*, pp. 3965–3977, 2021.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. 2021.

Arthur Douillard, Eduardo Valle, Charles Ollion, Thomas Robert, and Matthieu Cord. Insight from the future for continual learning. In *European Conference on Computer Vision*, 2020.

Arthur Douillard, Alexandre Ramé, Guillaume Couairon, and Matthieu Cord. Dytox: Transformers for continual learning with dynamic token expansion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Alaaeldin El-Nouby, Hugo Touvron, Mathilde Caron, Piotr Bojanowski, Matthijs Douze, Armand Joulin, Ivan Laptev, Natalia Neverova, Gabriel Synnaeve, Jakob Verbeek, et al. Xcit: Cross-covariance image transformers. *arXiv preprint arXiv:2106.09681*, 2021.

Jianyuan Guo, Kai Han, Han Wu, Chang Xu, Yehui Tang, Chunjing Xu, and Yunhe Wang. CMT: convolutional neural networks meet vision transformers. 2022.

Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. 2021.

German I. Parisi, Ronald Kemker, Jose L. Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 2019.

Namuk Park and Songkuk Kim. How do vision transformers work? *International Conference on Learning Representations*, 2022.

Z. Peng, W. Huang, S. Gu, L. Xie, Y. Wang, J. Jiao, and Q. Ye. Conformer: Local features coupling global representations for visual recognition. In *International Conference on Computer Vision*, 2021.

Jathushan Rajasegaran, Munawar Hayat, Salman Khan, Fahad Shahbaz Khan, and Ling Shao. Random path selection for incremental learning. *Advances in Neural Information Processing Systems*, 2019.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017.

Dongsub Shim, Zheda Mai, Jihwan Jeong, Scott Sanner, Hyunwoo Kim, and Jongseong Jang. Online class-incremental continual learning with adversarial shapley value. *Association for the Advancement of Artificial Intelligence*, 2021.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training data-efficient image transformers distillation through attention. In *International Conference on Learning Representations*, pp. 10347–10357, 2021.

Xiaolong Wang, Yufei Ye, and Abhinav Gupta. Zero-shot recognition via semantic embeddings and knowledge graphs. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

Ziyang Wu, Christina Baek, Chong You, and Yi Ma. Incremental learning via rate reduction. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

Yufei Xu, Qiming Zhang, Jing Zhang, and Dacheng Tao. Vitae: Vision transformer advanced by exploring intrinsic inductive bias. In *Advances in Neural Information Processing Systems*, pp. 28522–28535, 2021.

Shipeng Yan, Jiangwei Xie, and Xuming He. Der: Dynamically expandable representation for class incremental learning. 2021.

Yuzhe Yang and Zhi Xu. Rethinking the value of labels for improving class-imbalanced learning. *arXiv preprint*, 2020.

Zizhao Zhang, Han Zhang, Long Zhao, Ting Chen, Sercan Ö Arik, and Tomas Pfister. Nested hierarchical transformer: Towards accurate, data-efficient and interpretable visual understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 3417–3425, 2022.

H. Zhao, Y. Fu, M. Kang, Q. Tian, F. Wu, and X. Li. Mgsvf: Multi-grained slow vs. fast framework for few-shot class-incremental learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

A Appendix

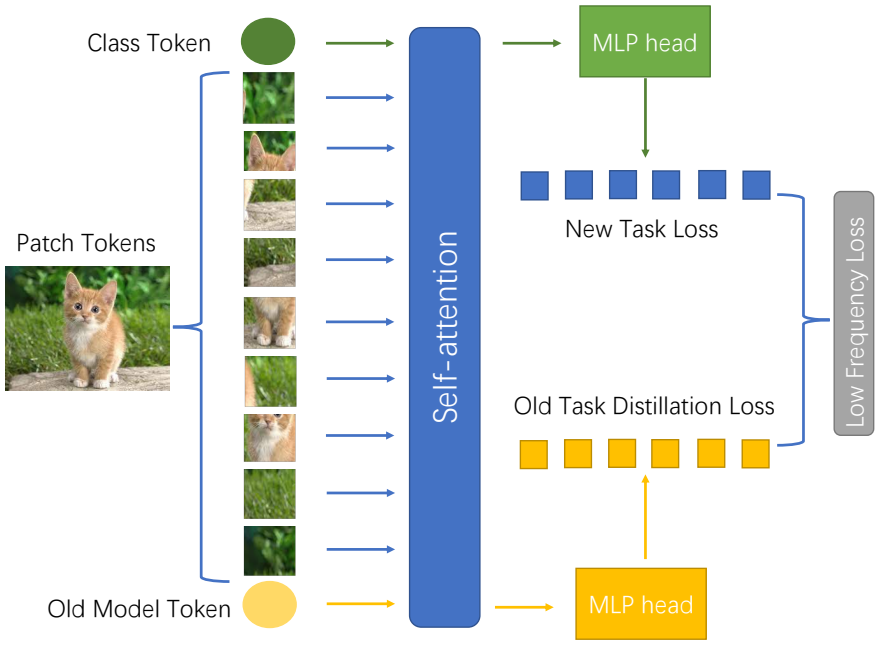


Figure 6: Low frequency disllation loss.