Salient Frequency-aware Exemplar Compression for Resource-constrained Online Continual Learning

Junsu Kim[1], **Suhyun Kim**[2*]

[1]Korea University, Republic of Korea
[2]Korea Institute of Science and Technology, Republic of Korea
j0807s@korea.ac.kr, dr.suhyun.kim@gmail.com

Abstract

Online Class-Incremental Learning (OCIL) enables a model to learn new classes from a data stream. Since data stream samples are seen only once and the capacity of storage is constrained, OCIL is particularly susceptible to Catastrophic Forgetting (CF). While exemplar replay methods alleviate CF by storing representative samples, the limited capacity of the buffer inhibits capturing the entire old data distribution, leading to CF. In this regard, recent papers suggest image compression for better memory usage. However, existing methods raise two concerns: computational overhead and compression defects. On one hand, computational overhead can limit their applicability in OCIL settings, as models might miss learning opportunities from the current streaming data if computational resources are budgeted and preoccupied with compression. On the other hand, typical compression schemes demanding low computational overhead, such as JPEG, introduce noise detrimental to training. To address these issues, we propose Salient Frequency-aware Exemplar Compression (SFEC), an efficient and effective JPEG-based compression framework. SFEC exploits saliency information in the frequency domain to reduce negative impacts from compression artifacts for learning. Moreover, SFEC employs weighted sampling for exemplar elimination based on the distance between raw and compressed data to mitigate artifacts further. Our experiments employing the baseline OCIL method on benchmark datasets such as CIFAR-100 and Mini-ImageNet demonstrate the superiority of SFEC over previous exemplar compression methods in streaming scenarios.

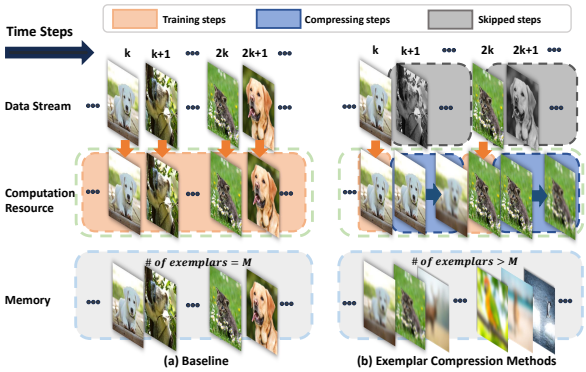

Figure 1: The streaming scenario underscores the significant impact of computational costs associated with compression in Online Class Incremental Learning (OCIL). (a) We assume the baseline (Caccia et al. 2022) captures the entire distribution of streaming data (b) Existing exemplar compression methods (Caccia et al. 2020; Wang et al. 2022; Luo et al. 2023) enable saving more exemplars within the restricted memory. However, the compression introduces additional computational overhead (i.e., forward and/or backward passes). Since training and compression processes compete for the same limited resources, these exemplar compression methods risk missing opportunities to learn from streaming data.

1 Introduction

With the increasing prevalence of personal intelligent devices and applications, vast amounts of data are generated and consumed daily. To enhance user experience, ensure privacy, and maintain real-time performance, there is a pressing need to update deep learning models directly with streaming data. This necessity has led to the emergence of a new learning paradigm called Online Class Incremental Learning (OCIL) (Mai et al. 2022; Wang et al. 2023a; Liu 2020; Lange et al. 2021).

Meanwhile, Deep Neural Networks (DNNs) are undergoing exponential growth in complexity and size (Ghodlami et al. 2024). However, edge platforms are constrained by limited computational power, memory, and storage resources, facing significant challenges in keeping pace with the rapid expansion of DNN applications. Although advancements in hardware technologies, such as Compute Express Link (CXL) memory (Samsung; SK hynix), have been introduced to mitigate some of the resource constraints, the adoption of such technologies remains in progress (SK hynix; Ha et al. 2023). Consequently, edge platforms continue to suffer from a pronounced scaling gap between the capabilities of existing hardware and the demands of emerging DNN applications.

In this context, OCIL is particularly vulnerable to Catastrophic Forgetting (CF), a phenomenon where a model for-

*Corresponding author.

gets previously acquired knowledge (McCloskey and Cohen 1989). As edge platforms have limited computational resources, a continual learner could potentially miss the opportunity to learn streaming data due to the computation overhead for learning if the computation resources are already in use when new data arrives. (Ghunaim et al. 2023; Zhang et al. 2024; Prabhu et al. 2023; Seo, Koh, and Choi 2024; Ma et al. 2023). A recent study reveals that computationally efficient methods, such as ER and ER-ACE, outperform numerous other OCIL methods (Ghunaim et al. 2023). Moreover, the restricted capacity of the buffer leads to data imbalances between old and new classes, which becomes more significant as the task progresses (Ahn et al. 2021; Caccia et al. 2022; Lin et al. 2023; Luo et al. 2023).

To overcome memory constriction, several papers suggest storing compressed data in the buffer, thereby increasing the number of exemplars available for replay (Caccia et al. 2020; Wang et al. 2022; Luo et al. 2023). For example, AQM learns Vector-Quantised Variational Auto-Encoder (VQ-VAE) to store encoded representations instead of full images (Caccia et al. 2020), and MRDC compresses images using JPEG at various quality levels, selecting the optimal one for training (Wallace 1992; Wang et al. 2022). However, these methods introduce additional computational overhead due to the optimization required for compression, which can be particularly problematic on resource-constrained edge devices. As the optimization demands forward and/or backward passes, the previous works incur additional computational burdens and occupy the limited computation resources (e.g., GPUs) during training.

Since the compression and training processes share the same limited resources, as shown in Figure 1, the computational overhead from compression limits the ability of a model to capture the entire data stream for learning. In our experiments, AQM loses the chance to process 50% of the incoming data, while MRDC and CIM only handle 37.5% and 20% of the total streaming data, respectively, compared to the simple baseline ER-ACE (Caccia et al. 2022). Therefore, despite the effectiveness of these optimization-based compression methods, their efficacy in streaming scenarios on resource-limited edge platforms remains compromised.

In this sense, we focus on developing **a computationally efficient compression method that simultaneously minimizes compression defects for learning**. JPEG (Wallace 1992) is a viable option for computational efficiency as it achieves a high compression rate without necessitating additional forward and/or backward passes. However, JPEG may lead to considerable compression defects for training. JPEG compression is designed for the human visual system, which is less sensitive to high frequencies than low frequencies. Consequently, JPEG uses quantization tables that apply significantly larger quantization coefficients to high frequencies than low frequencies. The intensive quantization on high frequencies can lead to a loss of valuable information for training since networks take advantage of both low and high frequencies when training (Abello, Hirata, and Wang 2021; Wang et al. 2023b; Chen, Ren, and Yan 2022; Chen et al. 2021; Lv and Zhu 2021; Zhang et al. 2023).

To achieve computational efficiency and the reduction of compression artifacts, we propose a simple yet effective JPEG-based compression framework for OCIL called Salient Frequency-aware Exemplar Compression (SFEC). Akin to JPEG compression, SFEC achieves computational efficiency and a high compression rate via frequency quantization and entropy encoding. However, unlike JPEG, SFEC intensively quantizes less salient frequencies, which are derived from gradients of spatial frequencies.

Even though SFEC quantizes less salient spatial frequencies, the compression may lead to artifacts due to the loss of information from quantization. To further alleviate the impact of compression defects on training, we introduce a compression-aware buffer management scheme. Specifically, our framework stores the reconstruction error between the original and reconstructed data after compression with a compressed image. After the buffer is filled, SFEC utilizes the distance as a weight for exemplar eviction. As a result, SFEC eliminates significantly distorted data with a high probability and successfully mitigates the effects caused by compression artifacts during training. For evaluation, we conduct experiments on two OCIL benchmark datasets, CIFAR-100 and MiniImageNet, under the streaming scenarios depicted in the recent paper (Ghunaim et al. 2023). With varying buffer sizes, SFEC consistently shows improvement in accuracy over the baseline and other compression methods by a significant margin. This paper makes the following contributions:

1. We observe the computational overhead from the previous exemplar compression methods, and the impact of the computation cost for training in streaming scenarios for OCIL.
2. We introduce Salient Frequency-aware Exemplar Compression (SFEC), a computationally efficient and adaptive compression framework for OCIL.
3. We propose compression-aware buffer management that effectively compromises compression defects with minimal cost.
4. We conduct a thorough evaluation in streaming scenarios on OCIL benchmark datasets with the baseline, ER-ACE. Our results show the superiority of SFEC over previous exemplar compression methods.

2 Related Work

2.1 Online Class Incremental Learning

Existing Online Class Incremental Learning (OCIL) is mainly based on ER (Rolnick et al. 2019), which implements a small buffer to store and replay old class data (Rolnick et al. 2019; Rebuffi et al. 2017; Aljundi et al. 2019a,b; Lin et al. 2023; Caccia et al. 2022; Gu et al. 2022; Guo, Liu, and Zhao 2022; Ahn et al. 2021; Buzzega et al. 2020; Chaudhry et al. 2018). For example, ER-ACE (Caccia et al. 2022) addresses Catastrophic Forgetting (CF) by regularizing the computation of softmax over previous and new classes. Meanwhile, MIR (Aljundi et al. 2019a) retrieves exemplars from memory whose losses increase the most with each model update. GSS (Aljundi et al. 2019b) updates the buffer with the incoming data that enriches the diversity of gradients in the memory.

2.2 Continual Exemplar Compression

Recently, several papers suggest compressing input data before memory update to better utilize the limited space of a buffer while reducing compression artifacts for training (Caccia et al. 2020; Wang et al. 2022; Luo et al. 2023). AQM (Caccia et al. 2020) trains Vector Quantised-Variational Auto-Encoder (VQ-VAE) for compression and replay. MRDC (Wang et al. 2022) compresses images using JPEG with the best compression quality among several candidates with several forward passes. A more recent approach, CIM (Luo et al. 2023) generates a saliency mask from learned activation functions and bi-level optimization. Subsequently, CIM down-samples non-discriminative pixels (i.e., out of the saliency mask).

2.3 JPEG-based Compression

JPEG (Wallace 1992) is a well-studied compression algorithm including RGB to YUV conversion, chroma-subsampling, Discrete Cosine Transform (DCT), and quantization. After quantization, JPEG constructs the bitstream using entropy encoding. JPEG decoding performs the inverse order of the encoding process. Typically, JPEG exploits the default quantization tables for the human visual system, which aggressively quantizes high frequencies. If users define a quality level for JPEG compression, JPEG scales the default quantization tables according to the scaling factor, quality (Tuba and Bacanin 2014). As JPEG compression may hinder model inference, GRACE (Xie and Kyu-Han 2019) and AutoJPEG (Xie et al. 2022) attempt to obtain optimal YUV conversion weights and quantization tables for model inference while maximizing the bandwidth (i.e., compression rate) for offloading.

2.4 Saliency

Saliency detection, originating from (Itti, Koch, and Niebur 1998), classifies the informative region of images. Numerous approaches suggest novel detection mechanisms (Achanta et al. 2009; Hou and Zhang 2007; Simonyan, Vedaldi, and Zisserman 2014; Wang et al. 2015; Shrikumar, Greenside, and Kundaje 2017; Li et al. 2015; Hou, Harel, and Koch 2011; Schauerte and Stiefelhagen 2012; Xie and Kyu-Han 2019). Saliency detection is categorized as whether it uses frequency information or neural networks. For example, Spectral Residual (Hou and Zhang 2007) identifies salient regions by distinguishing them from approximated background statistics in the frequency domain. Alternatively, model-based saliency maps utilize gradients of the input (Simonyan, Vedaldi, and Zisserman 2014; Wang et al. 2015). In this paper, we employ the gradients of the spatial frequencies as the saliency scores (Simonyan, Vedaldi, and Zisserman 2014; Wang et al. 2015).

3 Preliminaries

3.1 Online Class Incremental Learning

Online Class Incremental Learning (OCIL) splits a data stream into a sequence of learning tasks, $\mathcal{T} = \{\mathcal{T}_1, ..., \mathcal{T}_N\}$. When learning each task, $\mathcal{T}_t$, the model observes the corresponding data stream $\mathcal{D}_t$, which contains $(\mathcal{X}_t \bigcup \mathcal{Y}_t)$ and save the representative data as exemplars, $\mathcal{D}_o$. $\mathcal{D}_o$ consists of $(\mathcal{X}_o \bigcup \mathcal{Y}_o)$ in the memory buffer, $\mathcal{M}$. Unlike traditional OCIL, exemplar compression methods store compressed images $\mathcal{X}^c$ instead of $\mathcal{X}_o$ in $\mathcal{M}$. The performance of the model is estimated by a subset (i.e., held-out set) of $\{\mathcal{D}_1, ..., \mathcal{D}_t\}$ after each task. Thus, the model learns the combined dataset $(\mathcal{X}, \mathcal{Y}) = (\mathcal{X}_t, \mathcal{Y}_t) \bigcup (\mathcal{X}_o^c, \mathcal{Y}_o)$.

3.2 Streaming scenarios

In streaming scenarios with budgeted computational resources, the training complexity of any method for OCIL can lead to missing data from the current stream as illustrated in Figure 1 and in the recent study (Ghunaim et al. 2023). During task $\mathcal{T}_t$, a continual learner is trained on streaming data $\mathcal{D}_t$. This data stream, $\mathcal{D}_t$, involves a sequence of data points $\{\mathcal{X}_1, ..., \mathcal{X}_k\}$ represented in time steps k. As the training cost increases with certain OCIL methodologies, the learner misses the data stream more (i.e., the learner can observe only $\{\mathcal{X}_1, \mathcal{X}_3, ..., \mathcal{X}_{k-1}\}$). In contrast, a computationally efficient OCIL framework enables the model to capture the entire data stream.

4 Methodology

We propose Salient Frequency-aware Exemplar Compression (SFEC) framework for OCIL, focusing on computational efficiency and reduction in compression artifacts for training. To achieve minimal computational overhead, SFEC follows JPEG compression since JPEG does not require extra forward and/or backward passes (Wallace 1992). However, JPEG is optimized for human visual perception, which aggressively compresses high-frequency components (Xie and Kyu-Han 2019; Tuba and Bacanin 2014). In contrast to the human visual system, neural networks utilize both high and low frequencies to acquire new knowledge (Abello, Hirata, and Wang 2021; Wang et al. 2023b; Chen, Ren, and Yan 2022). Thus, employing JPEG in its standard for compressing exemplars can cause critical compression artifacts, adversely affecting model performance. To tackle this, we suggest adaptive quantization that aggressively quantizes less salient frequencies rather than high frequencies. As the gradients of the spatial frequencies attribute to the saliency as described in Section 2.4, we calculate the saliency score s_n, where $s_n = |g_n|$. Furthermore, we develop a simple yet effective buffer management scheme that evicts significantly distorted images after compression as they potentially degrade model performance during replay. Section 4.2 provides the detailed implementation.

4.1 Saliency-aware Frequency Quantization

Saliency-aware frequency quantization within SFEC framework is designed to efficiently find and store salient information in images. To measure the saliency score and facilitate JPEG-based compression, the incoming image ($\mathcal{X} \in \mathcal{R}^{\mathcal{C}\times\mathcal{H}\times\mathcal{W}}$, $\mathcal{C}$, $\mathcal{H}$, $\mathcal{W}$, mean the number of channels, height, and width) is converted into YUV channels ($\hat{\mathcal{X}} \in \mathcal{R}^{\mathcal{C}\times\mathcal{H}\times\mathcal{W}}$). Then, we split $\hat{\mathcal{X}}$ into patches ($\hat{\mathcal{X}} \in \mathcal{R}^{\mathcal{C}\times\frac{\mathcal{H}}{\mathcal{P}}\times\frac{\mathcal{W}}{\mathcal{P}}}$, $\mathcal{P}$ is the size of patches) to obtain a manageable magnitude of DCT coefficients, same to the typical JPEG compression.

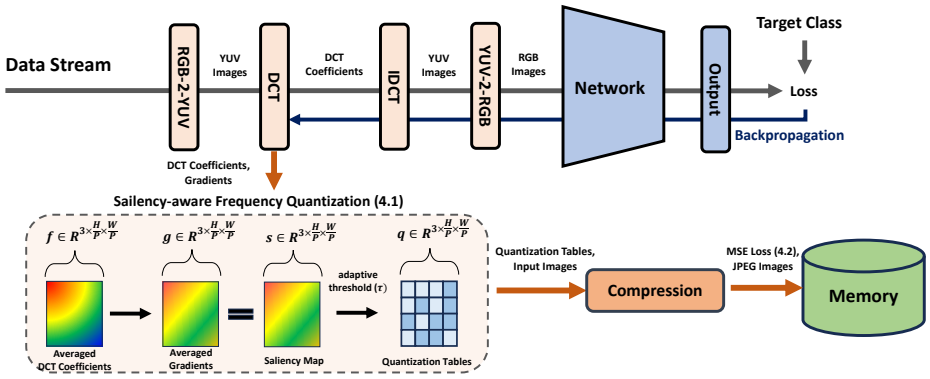

Figure 2: The overall framework of Salient Frequency-aware Exemplar Compression (SFEC). Given the input from the data stream, the framework converts the RGB input into YUV and spatial frequencies (f) using Discrete Cosine Transform (DCT). Then, SFEC reconstructs the data into its original format via Inverse Discrete Cosine Transform (IDCT) and YUV-2-RGB conversion for feed-forward. Subsequently, SFEC obtains gradients (g) of spatial frequencies, which attribute the saliency (s) of spatial frequencies. SFEC quantizes frequencies lower than the mean of the saliency score ($\bar{s}$) of the each image as described in Section 4.1. After compression, SFEC updates the buffer with reconstruction loss between original and compressed images to eliminate significantly distorted images after compression with high probability. Section 4.2 provides the details.

DCT is applied to each patch, resulting in spatial frequencies ($f \in \mathcal{R}^{C\times\frac{H}{P}\times\frac{W}{P}}$). For feed-forward the network, the original RGB data $\mathcal{X}$ is reconstructed through Inverse Discrete Cosine Transform (IDCT) and YUV-to-RGB conversion. The gradients of the spatial frequencies ($g \in \mathcal{R}^{C\times\frac{H}{P}\times\frac{W}{P}}$) are then obtained via backpropagation. The saliency score ($s \in \mathcal{R}^{C\times\frac{H}{P}\times\frac{W}{P}}$) is calculated as follows:

$$s = \bar{g} \tag{1}$$

where $\bar{g}$ are average gradients of the spatial frequencies across patches. Consequently, three patch sizes of saliency maps that represent YUV channels for each image are generated. Finally, we generate quantization tables for each channel as follows.

$$\hat{q_n} = \begin{cases} \bar{f}_n & s_n < \tau \\ 1 & s_n > \tau \end{cases} \tag{2}$$

where τ represents a threshold to determine whether the quantization is applied. To achieve adaptive quantization for each level, we set τ as

$$\tau = \frac{\lambda}{n} \times \sum_{i=1}^{n} \hat{s_i} \tag{3}$$

Although a hyperparameter λ is defined for generalization in our framework, we demonstrate its robustness to variations in λ. This suggests that additional forward and/or backward passes for optimizing λ, analogous to the previous works, are unnecessary (see Section 5.2 for details). Finally, our framework merges the partitioned quantization tables $\hat{q}$ and form the complete quantization tables ($q \in \mathcal{R}^{C\times\frac{H}{P}\times\frac{W}{P}}$). The tables, q, are applied for all patches of DCT coefficients f.

$$\hat{f} = [f \odot 1/q] \tag{4}$$

where $\hat{f}$ denotes quantized DCT coefficients, and [] indicates a rounding operation. The remaining steps adhere to the JPEG compression process (e.g., Huffman encoding). As a result, the buffer saves JPEG images as shown in Figure 2.

4.2 Compression-Aware Buffer Management

An inherent issue with lossy compression is loss of information, which can lead to artifacts. Although our framework compresses images based on saliency information to preserve valuable frequencies, compression may introduce undesirable defects in images for training. Notably, such distorted images repeatedly affect training while remaining in the buffer. To overcome this, we design a new buffer policy that leverages the extent of distortion from compression. The buffer management scheme records the distance between compressed and original data to calculate weights for elimination. As the distance scale may vary depending on each class, we normalize the distance across each class and utilize it as the weight. Akin to Reservoir sampling (Vitter 1985), the new data is inserted in the buffer with the probability of N_m/N_t, where N_m is the total number of images in the buffer and N_t denotes the total number of samples that the model has observed. Finally, our framework deletes the indices sampled from Multinomial Distribution $Multn$ using the weights if the sampled indices exist in the buffer.

We employ reconstruction error between images before and after the compression as a measure of distance. In contrast to the herding (Rebuffi et al. 2017), used for offline continual learning, OCIL does not allow to store all images of each class, indicating that the embedding mean of each class is unavailable during training. Therefore, we adopt reconstruction loss instead of using the distance between the embedding space.

Given the original RGB image $\mathcal{X}_i$, and the reconstructed image $\mathcal{X}_i^c$ after compression, the distance is

$$d_i(\mathcal{X}_i^c, \mathcal{X}_i) = \|(\mathcal{X}_i^c - \mathcal{X}_i)\| \tag{5}$$

Our compression framework updates the buffer with compressed images and their corresponding distances. If the buffer is filled with compressed images, SFEC calculates the weights for deletion tailored to the distance. The dimension of the weights is identical to N_t, and each W_i is calculated as follows:

$$W_i = \begin{cases} (N_c \times d_i)/(\sum_{k=1}^{N_c} d_k \times N_t) & if\ i \leq N_m \\ 1/N_t & if\ i > N_m \end{cases} \quad (6)$$

N_c represents the total number of indices with in the corresponding class. The distance for each index is normalized to sum to 1 across indices within the same class.

Lastly, the buffer removes the indices sampled from the multinomial distribution, $Multn$, if the indices exist in the buffer.

$$idx \sim Multn(N_t \mid W) \quad (7)$$

5 Experiments

5.1 Experimental Settings

Datasets and implementation details The experiments are conducted on two benchmark datasets: CIFAR-100 (Krizhevsky, Hinton et al. 2009) and Mini-ImageNet (Vinyals et al. 2016). Specifically, Mini-ImageNet is downsized to $[64 \times 64]$. The datasets are split into ten tasks, and each task consists of 10 classes. Also, we conduct experiments varying the size of the memory buffer, M, with {500, 1000, 2000, 5000} for all methodologies and datasets. The backbone model is Reduced ResNet18, which is consistent with numerous studies in online continual learning (Lin et al. 2023; Caccia et al. 2022, 2020; Gu et al. 2022; Aljundi et al. 2019a,b; Chaudhry et al. 2018). The network is trained on samples drawn from both the data stream and the memory, utilizing a learning rate of 0.1 with the SGD optimizer. For other hyperparameters of the existing methods, we follow the papers (Caccia et al. 2020; Wang et al. 2022; Luo et al. 2023). For instance, MRDC sets five quality candidates for JPEG compression, which are 10, 25, 50, 75, 90. CIM uses a compression ratio of 4.0 for non-discriminative pixels. We evaluate AQM only on CIFAR-100 using the one block of VQ-VQE due to its varying computational cost and memory requirements across datasets. We use a quality level of 75 for naive JPEG, which is the default value. Meanwhile, SFEC utilizes the same patch size (8×8) and the YUV conversion weights with JPEG. In this paper, we conducted all experiments using $\lambda = 1$.

We examine ER-ACE as the baseline since it requires the same computational overhead as ER, the simplest OCIL strategy, while achieving superior performance than ER (Caccia et al. 2022). ER-ACE performs a single gradient update over a batch from the stream and a batch from the memory. All comparison methods are incorporated in ER-ACE, including the delay caused by computation overhead, depicted as δ.

Methods	GFLOPs	C_s	Delay (δ)
ER-ACE (Caccia et al. 2022)	6.3G	1	0
AQM (Caccia et al. 2020)	12.6G	2	1
MRDC (Wang et al. 2022)	16.8G	$8/3$	$5/3$
CIM (Luo et al. 2023)	69.6G	6	5
SFEC (Ours)	6.3G	1	0

Table 1: Computational overhead and the corresponding delay of exemplar compression methods.

Streaming scenarios and delay models We consider the fast and slow streaming scenarios discussed in (Ghunaim et al. 2023). The fast streaming scenario refers to the environment in which the continual learner learns from the stream, quickly incoming and passing by. We assume the additional overhead to the baseline limits the learner from capturing the whole data stream as described in Section 3.2. On the contrary, the slow streaming scenario depicts the circumstance that the stream allows sufficient time for processing complex learning methods until the next data comes. In this case, computationally efficient methods can perform multiple gradient updates using the same data while the more complex methods are still processing.

The computational overhead of each method is presented in Table 1. The relative computation cost and the corresponding delay are denoted as C_s and δ. Following the paper (Ghunaim et al. 2023), we also measure the GFLOPs for forward passes using FlopsProfiler and manually calculate GFLOPs for the backward passes. Then, the relative computational complexity C_s is calculated based on the baseline, ER-ACE. If a method requires C_s of 2, the corresponding delay is 1. While the baseline updates the model with the whole data under the fast streaming scenario, CIM misses five batches of stream data after performing one model update due to compression. In the slow streaming scenario, we assume that CIM captures the entire data stream. During CIM compresses batch images, the baseline performs five times more model updates with the same data, which is depicted as ER-ACE++.

Overhead analysis Our framework introduces a negligible amount of backward passes on input sizes (i.e., $3 \times \mathcal{C} \times \mathcal{H} \times \mathcal{W}$, including the backward passes on f, $\hat{\mathcal{X}}$, $\mathcal{X}$). This computation overhead is equivalent to adding parameters to the models for backpropagation, which are about 10K on CIFAR-100 dataset. Such overhead is minimal compared to the total number of model parameters (i.e., less than 1% as Reduced ResNet18 contains 1.1M parameters). Furthermore, even for Mini-ImageNet, the computation overhead accounts for less than 3% of the Reduced Resnet18 and is still negligible compared to existing approaches. Since CPUs handle image storage and buffer management, saving images and the buffer management are decoupled from the training loop, and their latency can be hidden.

Evaluation metric The performance of the model is estimated by a held-out dataset of $\{\mathcal{D}_1, ..., \mathcal{D}_N\}$ after each task $\mathcal{T}_\sqcup$. The accuracy of the model on i-th task after training k-th task is defined as a_k^i. We can measure the average accuracy rate A_k. In this paper, we use the final average accuracy, A_N,

Table 2: Final average accuracy (higher is better) in the fast streaming scenario. The additional delay from each method is depicted as a δ. MIR (Aljundi et al. 2019a) and GSS (Aljundi et al. 2019b) are memory update and retrieval strategies, which demonstrate the impact of the delay for buffer-relevant methods. The complexity of MIR and GSS follows the description in (Ghunaim et al. 2023). All the results are reported as an average of 10 runs, with the best scores highlighted in boldface.

Dataset [sample size]	CIFAR-100 [32×32]				Mini-ImageNet [64×64]			
Methods	M=500	M=1000	M=2000	M=5000	M=500	M=1000	M=2000	M=5000
ER-ACE ($\delta = 0$)	14.91 ± 0.8	18.26 ± 0.7	21.74 ± 0.4	26.12 ± 0.8	11.36 ± 0.9	13.59 ± 1.0	17.41 ± 1.2	20.96 ± 1.7
MIR ($\delta = 1.5$)	14.59 ± 0.7	16.89 ± 0.8	21.83 ± 0.6	25.98 ± 0.7	10.39 ± 1.1	12.18 ± 1.5	18.16 ± 0.9	21.44 ± 1.8
GSS ($\delta = 5$)	5.54 ± 1.1	7.27 ± 1.0	9.40 ± 0.7	10.17 ± 1.6	4.0 ± 0.8	4.73 ± 0.8	5.52 ± 0.8	6.19 ± 0.8
AQM ($\delta = 1$)	8.67 ± 0.7	8.73 ± 1.1	9.75 ± 1.0	10.37 ± 1.1	N/A	N/A	N/A	N/A
MRDC ($\delta = 5/3$)	14.59 ± 0.7	16.08 ± 1.5	17.85 ± 1.9	17.89 ± 2.2	8.80 ± 1.2	8.47 ± 1.1	8.30 ± 1.3	7.74 ± 0.9
JPEG ($\delta = 0$)	14.56 ± 0.7	17.42 ± 1.2	19.23 ± 1.0	20.2 ± 0.6	12.86 ± 2.2	12.42 ± 3.1	14.18 ± 3.6	14.99 ± 3.5
CIM ($\delta = 5$)	N/A	N/A	N/A	N/A	11.19 ± 0.7	12.99 ± 0.8	15.52 ± 0.6	19.93 ± 1.4
SFEC ($\delta = 0$)	**16.79 ± 1.0**	**20.38 ± 0.8**	**23.85 ± 1.2**	**27.48 ± 1.0**	**14.19 ± 0.7**	**17.64 ± 1.4**	**19.50 ± 1.8**	**22.45 ± 0.8**

Table 3: Final average accuracy (higher is better) under the slow streaming scenario. ER-ACE and SFEC performs multiple gradient updates when other methods compress images, which are ER-ACE++ and SFEC++, respectively. The additional delay of each method is denoted as a δ, and the best scores are in boldface. All the results are reported as an average of 10 runs.

Dataset [sample size]	CIFAR-100 [32×32]				Mini-ImageNet [64×64]			
Methods	M=500	M=1000	M=2000	M=5000	M=500	M=1000	M=2000	M=5000
AQM ($\delta = 1$)	9.76 ± 0.9	9.54 ± 1.1	10.03 ± 1.1	10.53 ± 1.7	N/A	N/A	N/A	N/A
ER-ACE++	15.31 ± 1.0	18.49 ± 0.8	21.96 ± 0.7	27.23 ± 0.7	11.54 ± 0.7	14.62 ± 0.7	17.42 ± 0.8	20.81 ± 1.3
SFEC++	17.26 ± 1.0	20.60 ± 1.1	24.60 ± 0.8	26.75 ± 0.9	14.94 ± 0.7	18.06 ± 1.0	20.75 ± 1.1	22.27 ± 0.9
MRDC ($\delta = 5/3$)	16.27 ± 0.6	16.24 ± 1.8	16.74 ± 1.8	17.11 ± 1.4	11.69 ± 0.7	12.11 ± 1.3	12.42 ± 1.0	11.97 ± 1.0
ER-ACE++	15.34 ± 1.0	18.58 ± 0.8	21.98 ± 0.7	27.34 ± 0.7	11.58 ± 0.7	14.73 ± 0.6	17.56 ± 0.8	20.93 ± 1.3
SFEC++	17.33 ± 1.0	20.66 ± 1.1	24.64 ± 0.8	26.86 ± 0.9	**15.06 ± 0.7**	18.21 ± 1.0	20.93 ± 1.1	22.65 ± 1.3
CIM ($\delta = 5$)	N/A	N/A	N/A	N/A	9.11 ± 0.6	7.24 ± 0.5	7.89 ± 1.1	8.34 ± 1.2
ER-ACE++	15.52 ± 0.8	18.62 ± 0.8	22.17 ± 0.5	27.21 ± 0.9	12.37 ± 0.6	15.41 ± 0.5	18.83 ± 0.7	22.36 ± 1.5
SFEC++	**17.41 ± 0.4**	**20.91 ± 0.7**	**25.20 ± 1.0**	**28.43 ± 0.7**	15.05 ± 0.7	**19.14 ± 0.7**	**21.99 ± 0.8**	**23.18 ± 1.7**

for the evaluation metric, which is adopted by the prior studies (Lin et al. 2023; Caccia et al. 2022, 2020; Gu et al. 2022; Aljundi et al. 2019a,b; Chaudhry et al. 2018).

$$A_k = \frac{1}{k} \sum_{i=1}^{k} a_k^i \tag{8}$$

5.2 Results and Analyses

Fast streaming scenario Table 2 summarizes the results of CIFAR-100 and Mini-ImageNet benchmark dataset in the fast streaming scenario. As shown in the table, we observe performance degradation from other methods as they cause critical computation overheads, resulting in the failure to capture the entire stream data. On the contrary, SFEC consistently improves the baseline, ER-ACE, by a significant margin. Specifically, SFEC outperforms the baseline 1.87% and 2.61% on CIFAR-100 and Mini-ImageNet, respectively. Notably, SFEC achieves more accuracy gain on the more complex dataset, Mini-ImageNet. In contrast, all previous compression methods, including naive JPEG, exhibit a decrease in accuracy under the fast streaming scenario due to the additional complexity and compression defects.

Slow streaming scenario Table 3 reports the final average accuracy of the exemplar compression methods and the baseline. We have the following observation on the slow streaming experiments. 1) MRDC surpasses the baseline in the fast streaming scenario at $M = 500$. However, MRDC downgrades the baseline model performance at all different memory buffer sizes even without delay, indicating JPEG compression introduces compression artifacts that hinder effective training despite diverse exemplars. 2) Our framework mostly shows the improvement with all different settings on CIFAR-100 and Mini-ImageNet, which implies SFEC effectively mitigates compression defects and simultaneously diversifies exemplars.

ER-ACE	JPEG	SFQ	CABM	CIFAR-100	Mini-ImageNet
✔				21.74 ± 0.4	17.41 ± 1.2
✔	✔			19.23 ± 1.0	14.18 ± 3.6
✔	✔	✔		23.56 ± 0.4	18.27 ± 3.0
✔	✔	✔	✔	**23.85 ± 1.2**	**19.50 ± 1.8**

Table 4: The contribution of each component in SFEC to the overall improvement. We report the final average accuracy on CIFAR-100 and Mini-ImageNet with $M = 2000$ as an average of 10 runs.

Ablation study In this study, we split our methodology to assess the contributions of each component, shown in Table 4. Also, we report the average number of exemplars stored in the buffer of each compression method in Table 5.

As depicted in Table 4, JPEG compression downgrades the baseline performance. The accuracy degradation is especially significant on the more complex dataset, Mini-ImageNet. JPEG exploits the same quantization table for all DCT coefficients (i.e., patches). Thus, increasing the number of DCT coefficients with high resolution results in boosting the accumulation of the quantization error. On the contrary, Saliency-aware Frequency Quantization (SFQ) enhances performance on both datasets by leveraging salient frequencies for compression. Compressing less salient frequencies prevents critical distortion from the model perspective during training. Moreover, Compression-Aware Buffer Management (CABM) successfully mitigates JPEG-based

Memory	Dataset	AQM	MRDC	CIM	SFEC
$M = 0.5k$	CIFAR-100	9205	1092	N/A	826
	Mini-ImageNet	N/A	2468	895	1160
$M = 1k$	CIFAR-100	21205	2191	N/A	1645
	Mini-ImageNet	N/A	4726	1946	2334
$M = 2k$	CIFAR-100	39466	4223	N/A	3661
	Mini-ImageNet	N/A	8385	4081	4638
$M = 5k$	CIFAR-100	44144	9322	N/A	8371
	Mini-ImageNet	N/A	14133	9866	10714

Table 5: Averaged number of exemplars stored in the buffer after the final task on 10 runs. Since SFEC is based on JPEG, the number of exemplars are similar to MRDC. While AQM shows outstanding number of encoded exemplars, CIM stores less number of data compared to SFEC.

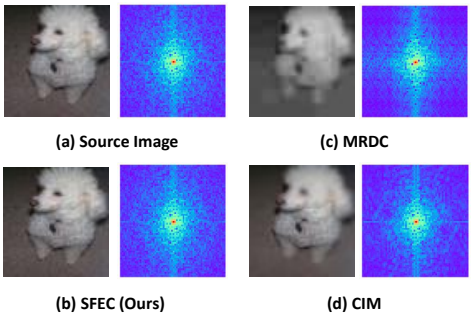

Figure 3: Visualization of compression artifacts in RGB channels (left) and frequency domain (right) of the 'miniature poodle' class on Mini-ImageNet.

compression artifacts by eliminating the defective images after compression and shows significant improvement.

Table 5 describes the number of exemplars remaining in the buffer after the final task. Since SFEC and MRDC are based on JPEG, the number of exemplars is comparable. While AQM shows an outstanding number of encoded exemplars, it shows relatively poor performance in streaming scenarios. CIM stores less data than SFEC. Since the bi-level optimization is unreliable in the OCIL setting, the saliency masks obtained from the optimization are large and show less compression rate.

Artifacts and visualization Figure 3 displays compression artifacts in the RGB channels and the frequency domain. Given that the source image and its spatial frequencies are shown in Figure 3 (a), the reconstructed image and spatial frequencies post-MRDC compression reveal a loss of high frequencies located at the corners. CIM causes a loss of both low and high frequencies due to down-sampling the majority part of images. In contrast, Figure 3 (d) depicts that SFEC preserves more RGB and frequency information compared to the recent exemplar compression methods.

Plugging in the State-of-the-art As SFEC requires minimal overhead and does not necessitate modifications to other continual learning methods, we integrate SFEC in the State-of-the-art (SOTA) OCIL frameworks, such as PCR. (Lin et al. 2023). Table 6 shows the averaged final accuracy of 3 runs. As depicted in our experiment, SFEC achieves performance improvements with PCR, indicating that SFEC is a promising option for enhancing the performance of other OCIL methods.

Memory	Methods	CIFAR-100	Mini-ImageNet
$M = 0.5k$	PCR	22.28 ± 0.6	15.91 ± 0.3
	PCR + SFEC	24.07 ± 1.7	18.81 ± 1.0
$M = 1k$	PCR	24.81 ± 0.5	17.92 ± 0.6
	PCR + SFEC	25.98 ± 1.3	19.19 ± 1.2
$M = 2k$	PCR	26.85 ± 0.8	20.53 ± 1.2
	PCR + SFEC	27.94 ± 1.0	21.72 ± 0.7
$M = 5k$	PCR	29.41 ± 1.0	21.13 ± 1.2
	PCR + SFEC	30.48 ± 1.2	22.56 ± 0.7

Table 6: Final average accuracy (higher is better) of the State-of-the-art OCIL method, PCR (Lin et al. 2023), and SFEC plugged in PCR. Without considering training complexity, SFEC enhances the performance of PCR in varying settings. The results are presented as an average of 3 runs.

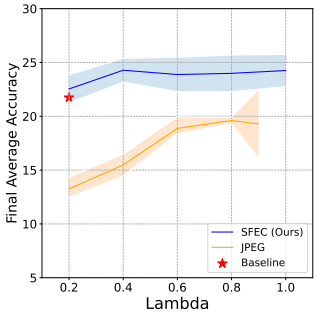

Figure 4: The illustration of hyperparameter sensitivity. The x-axis represents the λ in Equation 3 and quality for JPEG. The results are the final aveage accuracy, as an average of 3 runs on CIFAR-100 at $M = 2000$.

Hyperparameter sensitivity Figure 4 demonstrates the hyperparameter sensitivity of the proposed method, SFEC and JPEG. The results are the final average accuracy, as an average of 3 runs on CIFAR-100 at $M = 2000$. The x-axis refers to variations in λ described in Equation 3 and the specific quality for JPEG compression (i.e., a quality level of 100 is equal to 1.0). While the final accuracy using JPEG compression significantly varies depending on a quality level, SFEC shows consistent performance despite variations in λ. Therefore, additional forward and/or backward passes for optimizing λ are redundant in our experiments.

6 Conclusion

We present Salient Frequency-aware Exemplar Compression (SFEC) framework. SFEC allows for storing more representative exemplars within a limited capacity, with a minimal computational overhead. Our approaches compress the non-salient spatial frequencies and prevent performance degradation from learning outliers caused from compression. We show SFEC framework outperforms various exemplar compression methods in streaming scenarios.

Acknowledgments

This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2024-RS-2023-00258649, 90%) supervised by the IITP(Institute for Information & Communications Technology Planning & Evaluation). Especially, all of the experiments in this paper was conducted under the support of MSIT support program supervised by IITP. Also, this paper was result of the research project supported by SK hynix Inc. Specifically, the implementation of the research idea was proceeded with the support of SK hynix Inc. We would like to thank Minsoo Kang and Yujin Kim for their involvement in this project.

References

Abello, A. A.; Hirata, R.; and Wang, Z. 2021. Dissecting the high-frequency bias in convolutional neural networks. In *CVPRW*.

Achanta, R.; Hemami, S.; Estrada, F.; and Susstrunk, S. 2009. Frequency-tuned salient region detection. In *2009 IEEE conference on computer vision and pattern recognition*, 1597–1604. IEEE.

Ahn, H.; Kwak, J.; Lim, S.; Hyeonsu Bang, H. K.; and Moon, T. 2021. Ss-il: Separated softmax for incremental learning. In *ICCV*, 844–853.

Aljundi, R.; Belilovsky, E.; Tuytelaars, T.; Laurent Charlin, M. C.; Lin, M.; and PageCaccia, L. 2019a. Online continual learning with maximal interfered retrieval. In *NeurIPS*, 11849–11860.

Aljundi, R.; Lin, M.; Goujaud, B.; and Bengio, Y. 2019b. Gradient based sample selection for online continual learning. In *NeurIPS*.

Buzzega, P.; Boschini, M.; Porrello, A.; Abati, D.; and Calderara, S. 2020. Dark experience for general continual learning: a strong, simple baseline. In *NeurIPS*.

Caccia, L.; Aljundi, R.; Asadi, N.; Tuytelaars, T.; Pineau, J.; and Belilovsky, E. 2022. New insights on reducing abrupt representation change in online continual learning. In *ICLR*.

Caccia, L.; Belilovsky, E.; Caccia, M.; and Pineau, J. 2020. Online learned continual compression with adaptive quantization modules. In *ICML*, 1240–1250.

Chaudhry, A.; Ranzato, M.; Rohrbach, M.; and Elhoseiny, M. 2018. Efficient lifelong learning with a-gem. *arXiv preprint arXiv:1812.00420*.

Chen, G.; Peixi Peng, L. M.; Li, J.; Du, L.; and Tian, Y. 2021. Amplitude-phase recombination: Rethinking robustness of convolutional neural networks in frequency domain. In *ICCV*.

Chen, Y.; Ren, Q.; and Yan, J. 2022. Rethinking and Improving Robustness of Convolutional Neural Networks: a Shapley Value-based Approach in Frequency Domain. In *NeurIPS*.

Gholami, A.; Yao, Z.; Kim, S.; Hooper, C.; Mahoney, M. W.; and Keutzer, K. 2024. AI and memory wall. *IEEE Micro*.

Ghunaim, Y.; Bibi, A.; Alhamoud, K.; Alfarra, M.; Hammoud, H. A. A. K.; Prabhu, A.; Torr, P. H.; and Ghanem, B. 2023. Real-time evaluation in online continual learning: A new hope. In *CVPR*, 11888–11897.

Gu, Y.; Yang, X.; Wei, K.; and Deng, C. 2022. Online class-incremental continual learning via dual view consistency. In *CVPR*.

Guo, Y.; Liu, B.; and Zhao, D. 2022. Online continual learning through mutual information maximization. In *ICML*, 8109–8126.

Ha, M.; Ryu, J.; Choi, J.; Ko, K.; Kim, S.; Hyun, S.; Moon, D.; Koh, B.; Lee, H.; Kim, M.; Kim, H.; and Park, K. 2023. Dynamic Capacity Service for Improving CXL Pooled Memory Efficiency. *IEEE Micro*, 43(2): 39–47.

Hou, X.; Harel, J.; and Koch, C. 2011. Image signature: Highlighting sparse salient regions. *IEEE transactions on pattern analysis and machine intelligence*, 34(1): 194–201.

Hou, X.; and Zhang, L. 2007. Saliency detection: A spectral residual approach. In *2007 IEEE Conference on computer vision and pattern recognition*, 1–8. Ieee.

Itti, L.; Koch, C.; and Niebur, E. 1998. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 20(11): 1254–1259.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images.

Lange, D.; Matthias; Aljundi, R.; Masana, M.; Parisot, S.; Jia, X.; Leonardis, A.; Slabaugh, G.; and Tuytelaars, T. 2021. A continual learning survey: Defying forgetting in classification tasks. In *IEEE TPAMI*, 3365–3385.

Li, J.; Duan, L.-Y.; Chen, X.; Huang, T.; and Tian, Y. 2015. Finding the secret of image saliency in the frequency domain. *IEEE transactions on pattern analysis and machine intelligence*, 37(12): 2428–2440.

Lin, H.; Zhang, B.; Feng, S.; Li, X.; and Ye, Y. 2023. PCR: Proxy-based Contrastive Replay for Online Class-Incremental Continual Learning. In *CVPR*.

Liu, B. 2020. Learning on the job: Online lifelong and continual learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 13544–13549.

Luo, Z.; Liu, Y.; Schiele, B.; and Sun, Q. 2023. Class-incremental exemplar compression for class-incremental learning. In *CVPR*, 11371–11380.

Lv, B.; and Zhu, Z. 2021. Implicit bias of adversarial training for deep neural networks. In *International Conference on Learning Representations*.

Ma, X.; Jeong, S.; Zhang, M.; Wang, D.; Choi, J.; and Jeon, M. 2023. Cost-effective on-device continual learning over memory hierarchy with Miro. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*, 1–15.

Mai, Z.; Li, R.; Jeong, J.; Quispe, D.; Kim, H.; and Sanner, S. 2022. Online continual learning in image classification: An empirical survey. *Neurocomputing*, 469(1): 28–51.

McCloskey, M.; and Cohen, N. J. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of learning and motivation*, 24(1): 109–165.

Prabhu, A.; Al Kader Hammoud, H. A.; Dokania, P. K.; Torr, P. H.; Lim, S.-N.; Ghanem, B.; and Bibi, A. 2023. Computationally budgeted continual learning: What does matter? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3698–3707.

Rebuffi, S.-A.; Kolesnikov, A.; Sperl, G.; and Lampert, C. H. 2017. ICaRL: Incremental classifier and representation learning. In *CVPR*, 2001–2010.

Rolnick, D.; Ahuja, A.; Schwarz, J.; Lillicrap, T.; and Wayne, G. 2019. Experience replay for continual learning. In *NeurIPS*.

Samsung. ???? Samsung CXL Solutions – CMM-H.

Schauerte, B.; and Stiefelhagen, R. 2012. Quaternion-based spectral saliency detection for eye fixation prediction. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part II 12*, 116–129. Springer.

Seo, M.; Koh, H.; and Choi, J. 2024. Budgeted Online Continual Learning by Adaptive Layer Freezing and Frequency-based Sampling.

Shrikumar, A.; Greenside, P.; and Kundaje, A. 2017. Learning important features through propagating activation differences. In *International conference on machine learning*, 3145–3153. PMLR.

Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2014. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *ICLRW*.

SK hynix. ???? SK hynix CXL Solutions – CMM-DDR5 .

SK hynix. 2023. HMSDK Github. https://github.com/skhynix/hmsdk.

Tuba, M.; and Bacanin, N. 2014. JPEG quantization tables selection by the firefly algorithm. In *2014 International Conference on Multimedia Computing and Systems (ICMCS)*, 153–158. IEEE.

Vinyals, O.; Blundell, C.; Lillicrap, T.; and Wierstra, D. 2016. Matching networks for one shot learning. In *NeurIPS*.

Vitter, J. S. 1985. Random sampling with a reservoir. *ACM Transactions on Mathematical Software (TOMS)*, 11(1): 37–57.

Wallace, G. K. 1992. The jpeg still picture compression standard. *IEEE Transactions on Consumer Electronics*, 38(18-34): 11888–11897.

Wang, L.; Lu, H.; Ruan, X.; and Yang, M.-H. 2015. Deep networks for saliency detection via local estimation and global search. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3183–3192.

Wang, L.; Zhang, X.; Su, H.; and Zhu, J. 2023a. A comprehensive survey of continual learning: Theory, method and application. *arXiv preprint*, arXiv:2302.00487.

Wang, L.; Zhang, X.; Yang, K.; Longhui Yu, C. L.; Hong, L.; Zhang, S.; Zhenguo Li, Y. Z.; and Zhu, J. 2022. Memory replay with data compression for continual learning. In *ICLR*.

Wang, S.; Veldhuis, R.; Brune, C.; and Strisciuglio, N. 2023b. What do neural networks learn in image classification? A frequency shortcut perspective. In *ICCV*.

Xie, X.; and Kyu-Han, K. 2019. Source Compression with Bounded DNN Perception Loss for IoT Edge Computer Vision. In *In The 25th Annual International Conference on Mobile Computing and Networking.*, 1–16.

Xie, X.; Zhou, N.; Zhu, W.; and Liu, J. 2022. Bandwidth-Aware Adaptive Codec for DNN Inference Offloading in IoT. In *ECCV*, 88–104.

Zhang, G.; Zhang, Y.; Zhang, T.; Li, B.; and Pu, S. 2023. PHA: Patch-Wise High-Frequency Augmentation for Transformer-Based Person Re-Identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14133–14142.

Zhang, W.; Mohamed, Y.; Ghanem, B.; Torr, P. H.; Bibi, A.; and Elhoseiny, M. 2024. Continual learning on a diet: Learning from sparsely labeled streams under constrained computation. *arXiv preprint arXiv:2404.12766*.