Online Continual Learning via Dynamic Expandable Recursive Model

Fei Ye
School of Information and Software Engineering,
University of Electronic Science and Technology of China
Chengdu, China
feiye@uestc.edu.cn

Adrian G. Bors
Department of Computer Science,
University of York
York, UK
adrian.bors@york.ac.uk

Abstract

The continual learning (CL) of novel concepts from new environments represents a popular and important topic aiming to manage catastrophic forgetting. Research studies have developed dynamic expansion models to deal with network forgetting in CL. Existing CL models usually explore the full capacity of activating parameters and representations while ignoring the previously learned representations when learning new tasks. In this paper, we propose a novel dynamic expansion model that incrementally accumulates and incorporates all previously learned representations into defining new experts to add to a mixture of experts in a recursive manner, aiming to reuse previously learned parameters and features to promote future task learning. We define a graph structure having each expert as a component node. We then propose a novel expandable expert graph attention mechanism that dynamically optimizes the graph when learning new tasks, maximizing the positive knowledge transfer. In addition, we propose a novel expert cooperation mechanism to promote the cooperation between all previous experts and with the currently updated expert. Furthermore, we propose a novel memory optimization approach, which encourages each expert to capture and learn completely different information, further improving performance. We provide the results of a series of experiments demonstrating that the proposed approach outperforms the state-of-the-art performance in CL.

CCS Concepts

• **Information systems → Online analytical processing**.

Keywords

Continual Learning, Dynamic Expansion Models, Lifelong Learning

ACM Reference Format:
Fei Ye and Adrian G. Bors. 2025. Online Continual Learning via Dynamic Expandable Recursive Model. In *Proceedings of the 33rd ACM International Conference on Multimedia (MM '25), October 27–31, 2025, Dublin, Ireland*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3746027.3755284

MM '25, Dublin, Ireland

ACM ISBN 979-8-4007-2035-2/2025/10
https://doi.org/10.1145/3746027.3755284

1 Introduction

Continual Learning, often referred to as Lifelong Learning, represents a rather complex, yet pragmatic learning framework where new data instances are continuously introduced into the training of a deep learning system, requiring the model to assimilate and acquire new knowledge while retaining previously learned information, [44]. This research area, increasingly becomes critical, because it facilitates the integration of various algorithms into real-time applications, such as autonomous driving and robot navigation. However, training deep learning models within the context of continual learning can lead to considerable performance decline on earlier tasks, a phenomenon known as catastrophic forgetting [44].

To tackle network forgetting in continual learning, recent research has introduced various methodologies, which can be broadly categorized into : rehearsal-based methods that actively retain past samples within memory buffers [4, 12], regularization-based methods that incorporate additional regularization terms in the loss function to mitigate the learning of new tasks [30, 40], and dynamic expansion frameworks that progressively introduce new processing units or layers in order to assimilate new information [13, 26]. Among these approaches, the utilization of a memory buffer system stands out as a straightforward yet effective solution for addressing model forgetting in continual learning. Nonetheless, a significant drawback of such systems is their limitation in accommodating an increasing number of tasks due to the finite memory capacity. To overcome this challenge, recent studies have proposed the dynamic creation of new sub-networks and parameters to facilitate the learning of new tasks [13, 26], thereby ensuring optimal performance across all previously learned tasks.

Most current dynamic expansion models [13, 18] primarily focus on activating certain primary parameters when learning a new task, while overlooking two critical aspects : (1) These approaches fail to leverage the full potential of all previously acquired parameters when addressing the learning of a new task; (2) They are not suitable for an online batch-to-batch learning framework. To tackle **the first challenge**, we introduce an innovative dynamic expansion framework that facilitates the accumulation of knowledge, while reusing the entire prior representations and parameters to improve new task learning. Specifically, we structure each expert within the proposed framework using two components: a transfer module and a linear classifier. The transfer module is designed to convert a general, semantically rich representation by using a pre-trained Vision Transformer [17] into a task-specific representation. The linear classifier, on the other hand, learns a predictive pattern based on the transferred features. In contrast to existing dynamic expansion models [13, 26], which continuously generate and learn new

sub-models without considering the comprehensive information from previously learned parameters, we propose a novel Dynamic Recursive Expansion (DRE) approach that systematically integrates all previously learned representations into the expert construction process, in a recursive fashion, thereby enhancing future task learning. Furthermore, we propose a novel Expandable Expert Graph Attention (EEGA) mechanism that formulates each expert as a node within a graph structure and regulates all previously learned representations throughout the optimization process. The proposed approach evolves and identifies an optimal knowledge relationship matrix to manage the information flow from all previously learned experts during the learning of a new task, thereby maximizing the benefits of positive knowledge transfer. To ensure better cooperation between all previously learned experts and the currently updating expert, we introduce a new Expert Cooperation Mechanism (ECM) that dynamically adjusts the significance of both previously learned and currently learned experts during the optimization process.

To tackle an online batch-to-batch learning paradigm, we present an innovative memory-updating strategy known as the Knowledge-Aware Memory Optimization (KAMO) approach. In contrast to existing memory techniques [2, 12], which focus on retaining data samples across all categories, the proposed KAMO approach is designed to preserve data samples that are distinct from all previously acquired samples. This memory-updating strategy enables each expert to assimilate diverse information throughout the optimization process. Specifically, the proposed KAMO approach assesses the selection score of an incoming sample by measuring the distance between representations derived from the currently updating expert and each previously learned expert. A data sample that yields a high selection score will be retained in the memory buffer due to its novelty for the learning system, thereby motivating the current expert to acquire sufficiently distinct knowledge. Additionally, to integrate the KAMO framework into a more practical continual learning context where task information is unavailable during the inference phase, we introduce a Knowledge Compression (KC) approach. This method learns a probabilistic generative model to compress and retain the acquired knowledge within a low-dimensional latent space, facilitating expert selection during the inference phase.

We assess the efficacy of the proposed Knowledge-Aware Memory Optimization (KAMO) framework through experiments conducted across various benchmarks. The empirical findings indicate that the KAMO framework mitigates catastrophic forgetting well while outperforming current state-of-the-art methodologies in continual learning. Our contributions can be summarized as follows: (1) We introduce an innovative KAMO framework that dynamically integrates all previously acquired representations into the construction process of a new expert in a recursive fashion, leveraging past information to enhance future task learning; (2) We propose a new EEGA mechanism to maximize the benefit of the positive knowledge transfer by optimizing a graph relation matrix to regulate all previously learned representations during new task learning; (3) We present a novel ECM approach to promote the new task learning by optimizing a better cooperation mechanism among all previously learned experts together with the currently updating expert; (4) We propose a new memory optimization strategy that encourages each expert to acquire sufficiently distinct information, fostering the knowledge diversity among experts; (5) We propose to learn a compact probabilistic representation for each expert, facilitating the selection of an appropriate expert for processing a given sample without requiring prior task information.

2 Related Work

Rehearsal-based methods represent a widely adopted strategy for mitigating network forgetting in continual learning, which involves the management of a memory buffer that consistently retains essential examples [5, 8, 21, 22, 27, 45, 49, 51, 57]. During the learning of new tasks, samples from this memory buffer are leveraged to update the model, thereby preventing catastrophic forgetting. Consequently, sample selection is pivotal within rehearsal-based techniques [22]. Furthermore, these methods can be integrated with regularization-based strategies to enhance the model's overall performance [2, 12, 14–16, 28, 39, 40, 43, 53, 63]. Beyond merely storing authentic training samples, rehearsal-based methods can also be executed through the training of deep generative models, such as Variational Autoencoders (VAEs) [33] or Generative Adversarial Networks (GANs) [19]. Generative models develop capabilities of producing high-fidelity past examples during new task learning, effectively addressing the issue of network forgetting [1, 31, 47, 54, 70].

Knowledge distillation (KD) techniques is based on a teacher model to enhance the performance of a student model during the optimization process, [20, 24]. This approach has demonstrated significant improvements in model efficacy within deep learning frameworks. Given its capabilities, KD has been investigated as a solution to mitigate network forgetting in continual learning scenarios. The core concept of applying KD in continual learning is to reduce the divergence between the outputs of the student and teacher models while learning new tasks, as introduced in Learning Without Forgetting (LWF) [37]. This method involves freezing the student module to function as a teacher after each task transition. Furthermore, integrating the KD methodology with rehearsal-based strategies into a cohesive optimization framework has been shown to enhance model performance, as proposed in the Incremental Classifier and Representation Learning (iCaRL) [48]. Specifically, iCaRL introduces an innovative nearest-mean-of-exemplars classification approach that bolsters the classifier's resilience to variations in data representations. The Contrastive Continual Learning [8] involves a novel self-KD strategy aimed at preserving previously acquired features and representations, with the goal of minimizing network forgetting.

Dynamic Network Architectures. Rehearsal and knowledge distillation (KD) methods demonstrate efficiency primarily with a limited and static set of tasks, while they struggle when task sequences become long. Recent research has tackled this issue by introducing dynamic expansion models that facilitate the incremental addition of new hidden layers and nodes within the architecture during the continual learning [13, 26, 29, 45, 52, 60, 67, 72]. A significant benefit for dynamic expansion models is their ability to maintain a good performance on the previously encountered tasks by freezing all parameters [52]. Moreover, the Vision Transformers (ViT) [17] have shown superior performance compared to Convolution Neural Networks (CNN) in dynamic expansion frameworks, [18, 68].

Interactive Continual Learning (ICL) [46] is a recent continual learning approach, which consists of a fast and a slow learning system, respectively. The approach proposed in this paper has several different properties from the ICL, summarized as follows: (1) The proposed approach can dynamically create new experts to deal with new tasks, while the ICL does not do that; (2) The ICL does not utilize the previously learned information to promote the learning of future tasks. In contrast, the proposed approach can fully explore the prior knowledge to promote future task learning; (3) This paper proposes a novel sample selection approach, called the Knowledge-Aware Memory Optimization (KAMO), which enables the model to be trained in an online batch-to-batch learning manner. In contrast, the ICL proposes to optimize the memory using a CL-vMF mechanism based on the von Mises-Fisher (vMF) distribution, which is different from our approach.

Another related approach proposed by Yu et al. [69], promotes a dynamic expansion framework to deal with continual learning. Our approach has several different properties from [69], summarized into two aspects: (1) [69] employs a CLIP model [42] as the basic backbone, while the proposed approach uses a pre-trained ViT as its basic backbone; (2) [69] does not utilize all previously learned parameters and representations when learning a new task. In contrast, this paper introduces a novel expandable expert graph attention mechanism to fuse all prior parameters and representations to promote future task learning. This paper, in contrast to existing methodologies [13, 29, 60], introduces an innovative dynamic expansion framework that leverages a pre-trained ViT as its core backbone. We also formulate a new dynamic recursive expansion strategy aimed at harnessing and accumulating essential data representation information to optimize the advantages derived from positive knowledge transfer.

3 Methodology

3.1 Problem Definition

In the following, we focus on tackling network forgetting within the framework of online continual learning, where each sample is encountered only once throughout the entire training process. Let $\mathcal{D}^j = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{n^j}$ and $\widehat{\mathcal{D}}^j = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{\hat{n}^j}$ represent the training and testing datasets, respectively, where n^j and $\hat{n}^j$ are the total number of training and testing samples. Each training dataset $\mathcal{D}^j$ is partitioned into C subsets in accordance with the class-incremental paradigm, denoted as $\{\mathcal{D}^j(1), \cdots, \mathcal{D}^j(C)\}$. Each subset $\mathcal{D}^j(i)$ corresponds to a specific task, referred to as the i-th task ($\mathcal{T}_i$). During the learning phase of each task ($\mathcal{T}_i$), we can only access the training subset $\mathcal{D}^j(i)$, while all previously encountered training subsets $\{\mathcal{D}^j(1), \cdots, \mathcal{D}^j(i-1)\}$ are unavailable. The primary objective of a continual learning model f_θ, trained on the i-th task ($\mathcal{T}_i$), is to minimize the cross-entropy loss $\mathcal{L}_{\text{CE}}$ associated with the current task as well as of all previously seen tasks, expressed as :

$$\theta^\star = \underset{\theta \in \Theta}{\operatorname{argmax}} \sum_{j=1}^{i} \sum_{c=1}^{n^j} \mathcal{L}_{\text{CE}}(\mathbf{y}_c^j, f_\theta(\mathbf{x}_c^j)), \tag{1}$$

where Θ is the parameter space and $\theta^\star$ is the optimal parameter set that can minimize the loss on all previously seen tasks. $\mathbf{x}_c^j$ and $\mathbf{y}_c^j$ denote the c-th labeled sample from the j-th task. However, finding the optimal parameter set $\theta^\star$ using Eq. (1) in continual learning is intractable because we can not access and use data samples from all previous tasks. Many continual learning studies have proposed various approaches to solve Eq. (1) by preserving fewer past data samples in a memory buffer [12] or preventing significant changes on the learned parameters [56]. Once the learning for all tasks is finished, we evaluate the models' performance on the whole testing dataset $\widehat{\mathcal{D}}^j$.

3.2 Overall Framework

Utilizing a singular model is insufficient for managing the learning of an increasing array of tasks due to the limitations in the model capacity. A dynamic expansion model presents a viable solution to this challenge. Nevertheless, the dynamic generation of each independent expert network within a framework incurs significant computational expenses. In this study, we propose an innovative dynamic expansion framework that leverages a pre-trained Vision Transformer (ViT) [17] as the foundational backbone. New experts are built upon this foundational backbone, offering robust feature representations imbued with rich semantic information. Consequently, we are able to design each expert as a compact neural network with a reduced number of layers.

Expert Network. Let $f_\theta : \mathcal{X} \to \mathcal{Z}$ be a pre-trained backbone, which receives a data sample $\mathbf{x}$ over the data space $\mathcal{X} \in \mathbb{R}^d$ and outputs a general representation $\mathbf{z}$ over the feature space $\mathcal{Z} \in \mathbb{R}^{d^z}$, where d and d^z are the dimensions of data and feature spaces, respectively. Let $E_j = \{f_{\zeta_j}, f_{\phi_j}\}$ be the j-th expert in a mixture framework, which consists of a feature representation network $f_{\zeta_j} : \mathcal{Z} \to \mathcal{Z}'$ and a linear classifier $f_{\phi_j} : \mathcal{Z}' \to \mathcal{Y}$, where $\mathcal{Z}' \in \mathbb{R}^{d^{z'}}$ and $\mathcal{Y} \in \mathbb{R}^K$ denote the feature and prediction space, respectively. Specifically, f_{ζ_j} receives the representation from the basic backbone and returns a feature vector that is used as an input for the linear classifier f_{ϕ_j}, which outputs a K-dimensional probability vector. The prediction process of the j-th expert is formulated as :

$$y' = \operatorname{argmax}\{F_{\text{Softmax}}(f_{\phi_j}(f_{\zeta_j}(f_\theta(\mathbf{x}))))\}, \tag{2}$$

where y' is the prediction and $F_{\text{Softmax}}(\cdot)$ is the softmax function. During the optimization process, the parameter set θ of the basic backbone is frozen while we continually update the parameters $\{\phi_j, \zeta_j\}$ using the cross-entropy loss, expressed as :

$$\begin{aligned}\mathcal{L}_{\text{CE}}(E_j, \mathbf{X}_t) = \sum\nolimits_{i'=1}^{|\mathbf{X}_t|} \Big\{ \sum\nolimits_{c=1}^{K} y_{i',c} \\ \log\left\{f_{\phi_j}(f_{\zeta_j}(f_\theta(\mathbf{x}_{i'})))(c)\right\}\Big\},\end{aligned} \tag{3}$$

where $y_{i',c}$ is the c-dimension of the i'-th sample from the data batch $\mathbf{X}_t$ at the t-th training iteration and $|\mathbf{X}_t|$ is the batch size. $f_{\phi_j}(f_{\zeta_j}(f_\theta(\mathbf{x}_{i'})))(c)$ is the c-th dimension of the prediction for the data $\mathbf{x}_{i'}$, achieved using the j-th expert (E_j).

3.3 Expandable Expert Graph Attention

The feature representation network f_{ζ_j} for the j-th expert (E_j) is designed to transform a general representation $\mathbf{z}$ from the foundational backbone f_θ into a flexible feature vector that captures task-specific information. However, various tasks may share semantic information, which can facilitate positive knowledge transfer

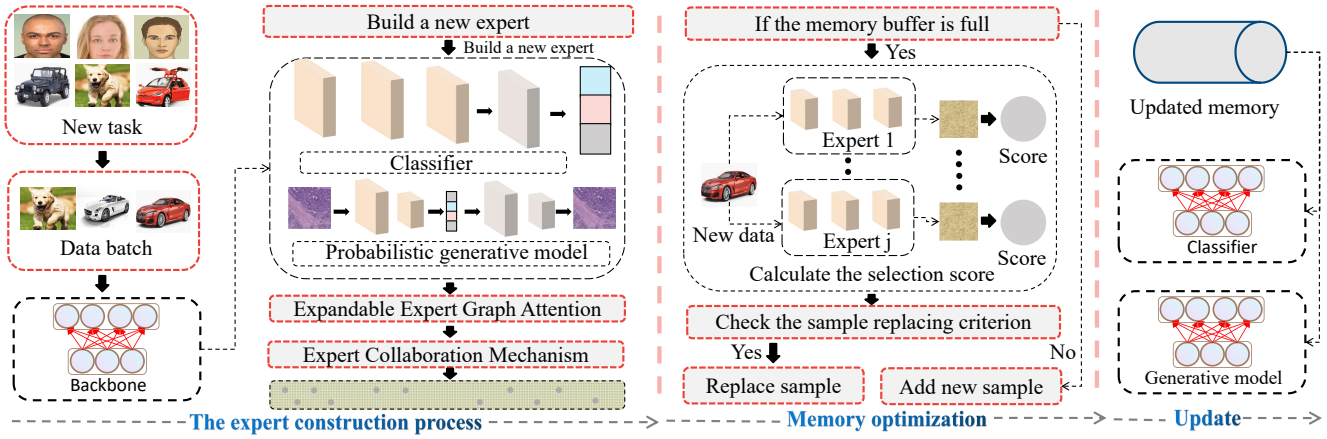


Figure 1: The learning procedure for the proposed framework, consisting of three steps at the j-th task learning. In the first step, we dynamically create a new expert $E_j = \{f_{\zeta_j}, f_{\theta_j}, q_{\omega_j^e}, q_{\omega_j^d}\}$ at the j-th task learning. Then we expand the relation matrix $\mathbf{w} \in \mathbb{R}^{(j,j)}$ and the dual adaptive vector $\widehat{\mathbf{w}}^j$, which are used to regulate representations during the optimization. In the second step, if the memory buffer is full, then we check the novelty of a new sample. If $|\mathcal{M}_t| = |\mathcal{M}|^{max}$, we replace a memorized sample using the new sample by Eq. (17). In the second step, we update the expert (E_j) on memorized and new samples, together.

when acquiring new tasks. To address this, we introduce an innovative recursive expansion process that dynamically integrates the entire previously acquired representation data to enhance future task learning. Specifically, let $\hat{\mathbf{z}}^1$ represent a feature vector derived from the feature extraction process $f_{\zeta_1}(f_\theta(\mathbf{x}))$ associated with the first expert (E_1). In the dynamic construction of the second expert (E_2), we preserve the first expert in a fixed state to prevent network forgetting while merging $\hat{\mathbf{z}}^1$ with the representation $\hat{\mathbf{z}}^2$ obtained through $f_{\zeta_2}(f_\theta(\mathbf{x}))$, resulting in an augmented feature vector denoted as $\tilde{\mathbf{z}}^2 = \hat{\mathbf{z}}^1 \cup \hat{\mathbf{z}}^2$, where $\cup$ signifies the concatenation of two feature vectors. This augmented representation $\tilde{\mathbf{z}}^2$ is then utilized as input for the linear classifier f_{ϕ_2} from the second expert (E_2). Similarly, when constructing the third expert (E_3), we consolidate all previously learned features $\hat{\mathbf{z}}^3$, expressed as $\tilde{\mathbf{z}}^3 = \tilde{\mathbf{z}}^2 \cup \hat{\mathbf{z}}^3$. For the j-th expert E_j, the construction process of an augmented representation results into :

$$\tilde{\mathbf{z}}^j = \bigcup_{i=3}^{j-1} \hat{\mathbf{z}}^i , \tag{4}$$

which incorporates all previously learnt representations in a recursive expansion manner, leading to positive knowledge transfer.

3.4 Expandable Expert Graph Attention

The feature composition from the previous section, according to Eq. (4), relies upon the equal contribution for each extracted feature representation throughout the optimization process. However, this fails to fully leverage the advantages of positive knowledge transfer. To address this challenge, we introduce an innovative expandable attention graph mechanism that modulates each previously acquired feature representation through a graph relational mechanism. Specifically, we conceptualize each previously acquired representation as a node within a graph structure and develop a relational matrix to characterize the knowledge architecture. Let $\mathbf{w} \in \mathbb{R}^{(j,j)}$ represent a scalable relational matrix, where its entries $w(j,i)$ indicate the implication of the representations derived by the i-th expert for the j-th expert. Furthermore, we can derive the normalized relational matrix $\mathbf{w}'$ for the j-th expert utilizing the weighting function :

$$F_{\text{normalized}}(\mathbf{w}, w(j,i)) = \frac{\exp\{\mathbf{w}(j,i)\}}{\sum_{c=1}^{|\mathbf{w}(j)|}\{w(j,c)\}}, \tag{5}$$

where $|\mathbf{w}(j)|$ denotes the number of elements for the j-the row of $\mathbf{w}$ and $w'(j,i) = F_{\text{normalized}}(\mathbf{w}, w(j,i))$. Then, the normalized graph relation matrix $\mathbf{w}'$ can be used to regulate all previously learnt features :

$$\bar{\mathbf{z}}^j = \left\{\sum_{c=3}^{j-1}\left\{w'(j,c)\tilde{\mathbf{z}}^c\right\}\right\}, \tag{6}$$

and each $\bar{\mathbf{z}}^c$ can be further decomposed, resulting in :

$$\bar{\mathbf{z}}^j = \left\{\sum_{c=3}^{j-1}\left\{w'(j,c)\sum_{c'=3}^{c-1}\{w'(c,c')\tilde{\mathbf{z}}^{c'}\}\right\}\right\}, \tag{7}$$

where $\{w'(c,c') \,|\, c = 3, \cdots, j-1, c < j, c' = 1, \cdots, |\mathbf{w}'(c)|\}$ are the elements from a relation matrix which are frozen to preserve the previously learned knowledge structure.

3.5 Expert Collaboration Mechanism

In order to have a better collaboration between the previously learned experts and the currently updating expert during the training, we introduce a new expert collaboration mechanism that incrementally creates new trainable adaptive weight vectors $\widehat{\mathbf{w}}^j = \{\widehat{\mathbf{w}}^j_{\text{previous}}, \widehat{\mathbf{w}}^j_{\text{current}}\}$ for each j-th expert to regulate the connection of the entire previously learned knowledge together with the new information, updating during the new task learning. Specifically, we propose to use the softmax function to normalize the adaptive weights $\widehat{\mathbf{w}}^j$, resulting in $\bar{\mathbf{w}}^j = \{\bar{\mathbf{w}}^j_{\text{previous}}, \bar{\mathbf{w}}^j_{\text{current}}\}$. The augmented representation $\tilde{\mathbf{z}}^j$ is expressed as :

$$\tilde{\mathbf{z}}^j = \bar{\mathbf{w}}^j_{\text{previous}}\bar{\mathbf{z}}^j \cup \bar{\mathbf{w}}^j_{\text{current}}\hat{\mathbf{z}}^j . \tag{8}$$

When optimizing the j-th expert during the training, we also update the associated adaptive parameters by :

$$\begin{aligned}\mathbf{w}(j,c) &= \mathbf{w}(j,c) - \eta_r \nabla_{\mathbf{w}(j,c)} \mathcal{L}_{\text{CE}}(E_j),\\ \widehat{\mathbf{w}}^j &= \widehat{\mathbf{w}}^j - \eta_r \nabla_{\widehat{\mathbf{w}}^j} \mathcal{L}_{\text{CE}}(E_j),\\ c &= 1, \cdots, |\mathbf{w}(j)|,\end{aligned} \tag{9}$$

where η_r is the learning rate. Once the j-th expert finishes its training process, we freeze all previously learned elements $\{\mathbf{w}(j') \mid j' = 3, \cdots, j\}$ of the relation matrix and learned adaptive weights $\widehat{\mathbf{w}}^j$, where $\mathbf{w}(j')$ denotes all elements from the j'-th row of $\mathbf{w}$. Then we dynamically expand the dimension of the relation matrix, expressed as $\mathbf{w} \in \mathbb{R}^{(j+1,j+1)}$. During a new task learning, we only optimize the newly added elements $\mathbf{w}(j+1)$ to promote the transfer learning. By integrating the proposed recursive expansion process, the EEGA and the ECM into a unified optimization framework, we create an expandable knowledge graph for each new expert, which maximizes the benefit of the positive knowledge transfer.

3.6 Knowledge-Aware Memory Optimization

The continual learning model is restricted to observing only a limited number of samples during each training session. In order to retain essential information, it is imperative to implement a compact memory buffer capable of storing numerous historical data samples [4, 12]. Numerous studies have explored a straightforward yet effective sample selection technique known as reservoir sampling [61], which randomly selects and retains past samples within the memory buffer. Nevertheless, reservoir sampling does not effectively identify the most critical data samples that would enhance performance. In this paper, we present an innovative memory optimization strategy termed Knowledge-Aware Memory Optimization (KAMO), which leverages all previously acquired knowledge to guide the memory optimization process. Specifically, we consider that the proposed mixture framework has already acquired j experts, allowing us to construct two sets of feature vectors for a given sample $\mathbf{x}$ as follows :

$$\begin{aligned}\mathbf{Z}^j(\mathbf{x}) &= \{f_{\phi_c}(f_{\zeta_c}(f_\theta(\mathbf{x})) \mid c = 1, \cdots, j-1\},\\ \bar{\mathbf{Z}}^j(\mathbf{x}) &= \{f_{\phi_j}(f_{\zeta_j}(f_\theta(\mathbf{x})), \cdots, f_{\phi_j}(f_{\zeta_j}(f_\theta(\mathbf{x}))\}.\end{aligned} \tag{10}$$

where $\mathbf{Z}^j(\mathbf{x})$ and $\bar{\mathbf{Z}}^j(\mathbf{x})$ represent the compressed knowledge of $\mathbf{x}$ using all previous experts $\{E_1, \cdots, E_{j-1}\}$ and the current expert (E_j), respectively. In order to easily evaluate the discrepancy between $\mathbf{Z}^j(\mathbf{x})$ and $\bar{\mathbf{Z}}^j(\mathbf{x})$, we allow $\bar{\mathbf{Z}}^j(\mathbf{x})$ to have the same length with $\mathbf{Z}^j(\mathbf{x})$. A significant divergence between $\mathbf{Z}^j(\mathbf{x})$ and $\bar{\mathbf{Z}}^j(\mathbf{x})$ indicates that the data sample $\mathbf{x}$ possesses sufficiently distinct information relative to the entire acquired knowledge and should therefore be incorporated into the memory buffer. To achieve this objective, we propose assessing the discrepancy between $\mathbf{Z}^j(\mathbf{x})$ and $\bar{\mathbf{Z}}^j(\mathbf{x})$ utilizing the Maximum Mean Discrepancy (MMD) criterion [58], which can effectively measure the difference between two empiric data distributions, making it well-suited for evaluating the difference between $\mathbf{Z}^j(\mathbf{x})$ and $\bar{\mathbf{Z}}^j(\mathbf{x})$. We provide the discussion about other distance choices in **Appendix-B** from SM.

Specifically, let $\mathbf{P}_{\mathbf{Z}^j(\mathbf{x})}$ and $\mathbf{P}_{\bar{\mathbf{Z}}^j(\mathbf{x})}$ denote the distribution of $\mathbf{Z}^j(\mathbf{x})$ and $\bar{\mathbf{Z}}^j(\mathbf{x})$, respectively. Let us define $\{f' \in \mathcal{F}' \mid f' : \mathcal{X} \to \mathbb{R}\}$ as a function where $\mathcal{F}'$ denotes a class of functions. The MMD criterion between two distributions $\mathbf{P}_{\mathbf{Z}^j(\mathbf{x})}$ and $\mathbf{P}_{\bar{\mathbf{Z}}^j}(\mathbf{x})$ is expressed as [58] :

$$\begin{aligned}&\mathcal{L}_{\text{MMD}}(\mathbf{P}_{\bar{\mathbf{Z}}^j(\mathbf{x})}, \mathbf{P}_{\mathbf{Z}^j(\mathbf{x})}) =\\ &\sup_{f' \in \mathcal{F}'} \left(\mathbb{E}_{\mathbf{z} \sim \mathbf{P}_{\mathbf{Z}^j(\mathbf{x})}}\left[f'(\mathbf{z})\right] - \mathbb{E}_{\mathbf{z}' \sim \mathbf{P}_{\mathbf{Z}^j(\mathbf{x})}}\left[f'(\mathbf{z}')\right]\right),\end{aligned} \tag{11}$$

where sup denotes the least upper bound of a set of numbers. If $\mathcal{L}_{\text{MMD}}(\mathbf{P}_{\bar{\mathbf{Z}}^j(\mathbf{x})}, \mathbf{P}_{\mathbf{Z}^j(\mathbf{x})}) = 0$, then we have $\mathbf{P}_{\bar{\mathbf{Z}}^j(\mathbf{x})} = \mathbf{P}_{\mathbf{Z}^j(\mathbf{x})}$. The function class $\mathcal{F}$ is usually a unit ball in a Reproducing Kernel Hilbert Space (RKHS) with a positive definite kernel $f_k(\mathbf{z}, \mathbf{z}')$. Eq. (11) is usually challenging to calculate and therefore we estimate the MMD on the embedding space [36], expressed as :

$$\mathcal{L}^2_{\text{MMD}}(\mathbf{P}_{\bar{\mathbf{Z}}^j(\mathbf{x})}, \mathbf{P}_{\mathbf{Z}^j(\mathbf{x})}) = ||\boldsymbol{\mu}_{\mathbf{P}_{\bar{\mathbf{Z}}^j(\mathbf{x})}} - \boldsymbol{\mu}_{\mathbf{P}_{\mathbf{Z}^j(\mathbf{x})}}||^2, \tag{12}$$

where $\boldsymbol{\mu}_{\mathbf{P}_{\bar{\mathbf{Z}}^j(\mathbf{x})}}$ and $\boldsymbol{\mu}_{\mathbf{P}_{\mathbf{Z}^j(\mathbf{x})}}$ denote the mean embeddings of $\mathbf{P}_{\bar{\mathbf{Z}}^j(\mathbf{x})}$ and $\mathbf{P}_{\mathbf{Z}^j(\mathbf{x})}$. In practice, when $\mathbf{P}_{\bar{\mathbf{Z}}^j(\mathbf{x})}$ and $\mathbf{P}_{\bar{\mathbf{Z}}^j(\mathbf{x})}$ are formed by using the same number of data samples, we can employ an unbiased empirical estimate, defined as :

$$\mathcal{L}'_{\text{MMD}}(\mathbf{P}_{\bar{\mathbf{Z}}^j(\mathbf{x})}, \mathbf{P}_{\mathbf{Z}^j(\mathbf{x})}) = \frac{1}{n'(n'-1)} \sum\nolimits_{c' \neq c}^{n'} h(c_1, c_2), \tag{13}$$

where $h(c_1, c_2)$ is defined as :

$$\begin{aligned}h(c_1, c_2) &= f_k(\bar{\mathbf{Z}}^j(\mathbf{x})[c_1], \bar{\mathbf{Z}}^j(\mathbf{x})[c_2])\\ &+ f_k(\mathbf{Z}^j(\mathbf{x})[c_1], \mathbf{Z}^j(\mathbf{x})[c_2]) - f_k(\bar{\mathbf{Z}}^j(\mathbf{x})[c_1], \mathbf{Z}^j(\mathbf{x})[c_2])\\ &- f_k(\mathbf{Z}^j(\mathbf{x})[c_2], \bar{\mathbf{Z}}^j(\mathbf{x})[c_1]),\end{aligned} \tag{14}$$

where $\bar{\mathbf{Z}}^j(\mathbf{x})[c_1]$ and $\mathbf{Z}^j(\mathbf{x})[c_1]$ denote the c_1-th sample drawn from $\mathbf{P}_{\bar{\mathbf{Z}}^j(\mathbf{x})}$ and $\mathbf{P}_{\mathbf{Z}^j(\mathbf{x})}$, respectively. Eq. (13) can evaluate the statistical distinction between $\mathbf{P}_{\bar{\mathbf{Z}}^j(\mathbf{x})}$ and $\mathbf{P}_{\mathbf{Z}^j(\mathbf{x})}$. Based on the MMD criterion, we can calculate the discrepancy score for a data sample $\mathbf{x}$ by considering all previously learned experts, expressed as :

$$F_{\text{dis}}(\mathbf{x}) = \mathcal{L}'_{\text{MMD}}(\mathbf{P}_{\bar{\mathbf{Z}}^j(\mathbf{x})}, \mathbf{P}_{\mathbf{Z}^j(\mathbf{x})}), \tag{15}$$

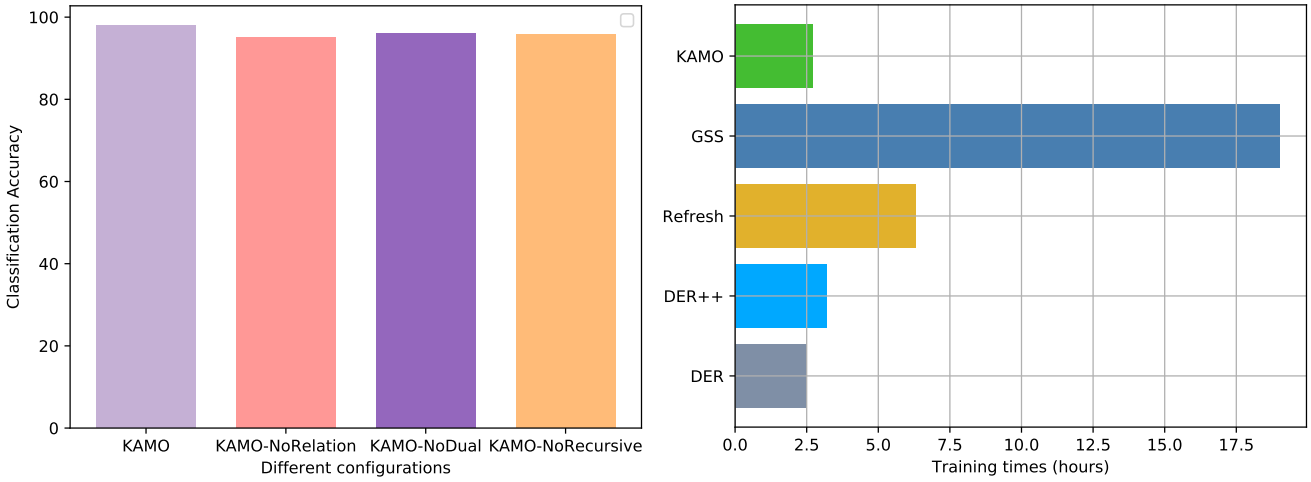
where $F_{\text{dis}}(\cdot, \cdot)$ is a distance measure function for evaluating the similarity between two data distributions. If the score calculated by Eq. (15) is large, it means that the data $\mathbf{x}$ is sufficiently different from all previously learned information and it should be preserved in the memory buffer, which helps the current expert in order to enrich the known information.

Let $\mathcal{M}_t$ be a memory buffer updated at the t-th training time and $|\mathcal{M}|^{max}$ is the maximum memory size. When the memory buffer is full $|\mathcal{M}_t| = |\mathcal{M}|^{max}$, we check wether we add a new data sample $\mathbf{x}$ into the memory buffer at the t-th training iteration :

$$\min_{m=1}^{|\mathcal{M}_t|} \{F_{dis}(\mathcal{M}_t[m])\} < F_{\text{dis}}(\mathbf{x}), \tag{16}$$

where $\mathcal{M}_t[m]$ is the m-th memorized sample from $\mathcal{M}_t$. If Eq. (16) is satisfied, then we replace the memorized sample $\mathcal{M}_t[m^\star]$ using the new sample $\mathbf{x}$:

$$\begin{aligned}\mathcal{M}_t[m^\star] &= \mathbf{x},\\ m^\star &= \operatorname*{argmax}_{m=1,\cdots,|\mathcal{M}_t|} \{F_{\text{dis}}(\mathcal{M}_t[m])\}.\end{aligned} \tag{17}$$



(a) Different configurations.

(b) The training times.

Figure 2: Ablation study results. (a) The different configurations for the proposed KAMO framework on the CIFAR100. (b) The training times for various models under the CIFAR10.

3.7 Expert Learning of Compact Knowledge

In a more realistic learning environment, the task information and labels are usually unavailable in the inference phase. It is impossible to use any auxiliary information to select an appropriate expert for a given sample. In this paper, we propose a new sample discrepancy mechanism that can automatically choose the appropriate expert for a given single sample or a batch of samples. Since the feature extractor $f_{\zeta_j}(f_\theta(\mathbf{x}))$ of each expert (E_j) can provide a strong and semantically rich representation for a given sample $\mathbf{x}$, we estimate the selection score on the low-dimensional feature vectors instead of the high-dimensional images, which reduces the computational costs and storage space requirements. Specifically, we introduce a probabilistic generative model $p_{\omega_j}(\tilde{\mathbf{z}}, \mathbf{z}^v) = p_{\omega_j}(\tilde{\mathbf{z}} \,|\, \mathbf{z}^v)p(\mathbf{z}^v)$ with the parameter set ω_j for the j-th expert (E_j), which has an observed variable $\tilde{\mathbf{z}}$ over the space $\mathcal{Z}'$ and a latent variable $\mathbf{z}^v$ over the space $\mathcal{Z}^v$. Optimizing the generative model $p_{\omega_j}(\tilde{\mathbf{z}}, \mathbf{z}^v)$ corresponds to maximizing the marginal sample log-likelihood, expressed as $\int p_{\omega_j}(\tilde{\mathbf{z}} \,|\, \mathbf{z}^v)p(\mathbf{z}^v)\, \mathrm{d}\mathbf{z}^v$, which is computationally intractable. To solve this issue, we consider maximizing a lower bound to the sample log-likelihood, given by the negative reconstruction loss compensated by the Kullback–Leibler (KL) divergence, as in VAEs [33] :

$$\begin{aligned}\mathcal{L}_v(E_j, \mathbf{x}) = -\mathbb{E}_{\mathbf{z}\sim q^e_{\omega_j}(\mathbf{z}^v \,|\, f_{\zeta_j}(f_\theta(\mathbf{x})))}\left[\log p_{\omega^d_j}(\tilde{\mathbf{z}} \,|\, \mathbf{z}^v)\right] \\ + D_{KL}\left[q_{\omega^e_j}(\mathbf{z}^v \,|\, f_{\zeta_j}(f_\theta(\mathbf{x}))) \,||\, p(\mathbf{z})\right],\end{aligned} \tag{18}$$

where the model $p_{\omega_j}(\tilde{\mathbf{z}} \,|\, \mathbf{z}^v)$ is divided into an inference model $q_{\omega^e_j}(\mathbf{z}^v \,|\, f_{\zeta_j}(f_\theta(\mathbf{x})))$ and a decoder $p_{\omega^d_j}(\tilde{\mathbf{z}} \,|\, \mathbf{z}^v)$ with the parameter sets $\{\omega^e_j, \omega^d_j\}$. $p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$ is a prior distribution, considered as Gaussian.

In the optimization process of the j-th expert (E_j), we also update the parameters $\{\omega^e_j, \omega^d_j\}$ on the samples from the memory buffer using Eq. (18), which encourages to learn and recognizing compact knowledge derived from the j-th expert. In the testing phase, the sample log-likelihood is evaluated as the expert selection score for a given testing sample $\mathbf{x}_{\text{test}}$ as :

$$s^\star = \underset{i=1,\cdots,j}{\text{argmax}}\{-\mathcal{L}_v(E_i, \mathbf{x}_{\text{test}})\}, \tag{19}$$

where $s^\star$ is the index of the selected expert for the given testing sample $\mathbf{x}_{test}$ and we employ the classifier to make the prediction $f_{\phi_{s^\star}}(f_{\zeta_{s^\star}}(f_\theta(\mathbf{x}_{\text{test}})))$.

3.8 Framework Implementation

The overall learning process for the proposed framework is shown in Figure 1. We provide the detailed learning process as :

Step 1 (The expert construction process). In the initial learning procedure, we build the first expert $\{f_{\zeta_1}, f_{\theta_1}, q_{\omega^e_1}, q_{\omega^d_1}\}$. In the following task learning (the j-th task learning), we build a new expert $\{f_{\zeta_j}, f_{\theta_j}, q_{\omega^e_j}, q_{\omega^d_j}\}$ with the associated adaptive weights $\widehat{\mathbf{w}}^j$. We also expand the dimension of the relation matrix $\mathbf{w} \in \mathbb{R}^{(j,j)}$ while the augmented representation is formed using Eq. (8).

Step 2 (The memory optimization). If the memory buffer is not full $|\mathcal{M}_t| < |\mathcal{M}_t|^{max}$, we continually add new data samples into the memory buffer; otherwise, we check the sample selection criterion. If $|\mathcal{M}_t| = |\mathcal{M}_t|^{max}$, we add the new sample into the memory buffer using Eq. (17).

Step 3 (The parameter optimization). We update the parameter set $\{f_{\zeta_j}, f_{\theta_j}, q_{\omega^e_j}, q_{\omega^d_j}\}$ of the current expert (E_j) using the new data batch $\mathbf{X}_t$ and memorized samples using Eq. (3) and Eq. (18), respectively. We also update the relation matrix and the adaptive weight using Eq. (9).

Table 1: The average accuracy calculated by various models on standard continual learning benchmarks, as averages of 10 runs. The results of baselines are taken from [7] and [46, 64].

Methods	CIFAR10	TinyImageNet	CIFAR100
ER [50]	93.61 ± 0.27	48.64 ± 0.46	73.37 ± 0.43
GEM [39]	92.16 ± 0.69	-	-
A-GEM [10]	89.48 ± 1.45	25.33 ± 0.49	48.06 ± 0.57
iCaRL [48]	88.22 ± 2.62	31.55 ± 3.27	-
FDR [6]	93.29 ± 0.59	49.88 ± 0.71	-
GSS [3]	91.02 ± 1.57	-	57.50 ± 1.93
HAL [9]	84.54 ± 2.36	-	42.94 ± 1.80
DER [7]	93.40 ± 0.39	51.78 ± 0.88	-
ICL w Pure-MM [46]	99.68	-	96.35
DualPrompt [65]	98.12	-	93.76
L2P [66]	96.78	-	93.92
DER++ [7]	93.88 ± 0.50	51.91 ± 0.68	75.64 ± 0.60
DER+++refresh [64]	94.64 ± 0.38	54.06 ± 0.79	77.71 ± 0.85
KAMO	**99.72** ± 0.46	**92.05** ± 0.42	**97.86** ± 0.37

Table 2: The average accuracy calculated by various models on complex continual learning benchmarks. The results of baselines are taken from [41].

Methods	ImageNet-R	CUB200	Cars
DualPrompt [65]	71.00	79.50	40.10
RanPAC [41]	77.90	90.30	77.50
L2P [66]	72.40	65.20	38.20
CODA-Prompt [55]	75.50	79.50	43.20
ADaM [71]	72.30	87.10	41.40
KAMD	**78.92**	**91.05**	**79.02**

Table 3: The training time (hours) of various models.

Methods	Split CIFAR10	Split TinyImageNet	Split CIFAR100
DER [7]	2.60	20.74	5.47
DER++ [7]	3.37	24.52	6.23
GEM [39]	6.53	42.20	30.21
KAMO	5.12	31.92	10.24

4 Experiment

4.1 Experiment Setting

Dataset. In the experiments, we consider several standard datasets used to evaluate the model's performance. Specifically, we adopt two popular continual learning paradigms, called Class Incremental Learning (Class-IL) [25, 59] and Task Incremental Learning (Task-IL). The main difference between Class-IL and Task-IL is that the task identifications are known and provided during the inference phase for the Task-IL setting. In addition, we divide each dataset into several non-overlapping subsets, with each subset containing data samples from several adjacent categories. As a result, we divide CIFAR10 [34], CIFAR100 [34], and TinyImageNet [35] into 5, 10, and 10 tasks, resulting in the **Split CIFAR10**, **Split CIFAR100**, and **Split TinyImageNet**, respectively. The number of classes for each task for the Split CIFAR10, Split CIFAR100, and Split TinyImageNet is 2, 10, and 20, respectively.

Baselines. We compare the proposed approach with several popular baselines, including Experience Replay (ER) [50], Gradient Episodic Memory (GEM) [10], Averaged GEM (A-GEM) [11], iCaRL [48], Dark Experience Replay (DER) [7] and DER+++refresh [64], Greedy Sample Selection (GSS) [3]. In addition, we also consider the recent state-of-the-art such as Interactive Continual Learning with Pure-MM (ICL w Pure-MM) as the backbone [46], and Loss DEcoupling (LODE) [38].

The hyperparameter configuration and GPU hardware. For all experiments, we employ Adam [32] as the optimization algorithm, which is used to train various models. We consider the learning rate of 0.0001 and the default values for the other hyperparameters of Adam. We set the number of training epochs as 100 for each task learning. The experiments are performed using a Tesla V100 GPU and we adopt the operating system as Ubuntu 18.04.5.

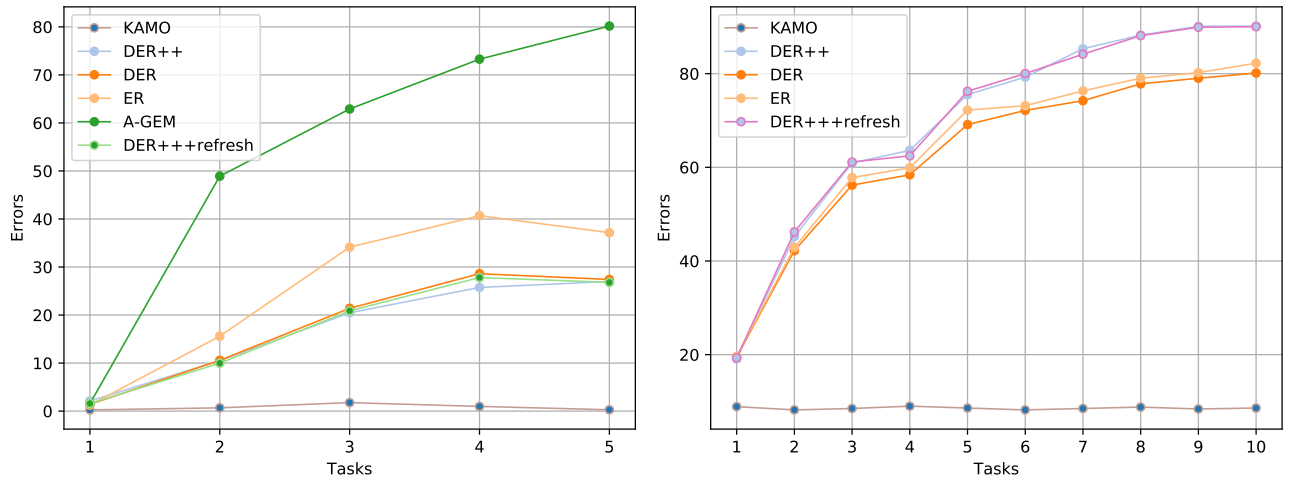
Network architecture and training details. According to the setting from [7], most of the methods used in the experiments, including ER, DER and GSS, adopt the ResNet18 [23] as the backbone for the Split CIFAR10, Split CIFAR100 and Split TinyImageNet, respectively. Recent CL models have explored a more powerful backbone, a pre-trained ViT [17], to address network forgetting in continual learning [46]. For a fair comparison, this paper also compares the proposed approach with the current state-of-the-art (ICL w Pure-MM) [46] that employs a pre-trained ViT as one of the components in the continual learning system. For all methods, the number of training epochs for Split CIFAR10, Split TinyImageNet and Split CIFAR100 is of 50, 100 and 100, respectively.

4.2 Comparisons on the Standard Benchmarks

In the experiment, we compare the proposed approach with different types of continual learning methods, including the memory-based, regularization-based and dynamic expansion methods. According to the setting from [7], we set the maximum memory size as 500 for all memory-based methods. We train various models on the Split CIFAR10, Split CIFAR100 and Split TinyImageNet, respectively, and evaluate the model's performance using the Class-IL and Task-IL performance criteria. The average results of various models on the Split CIFAR10, the Split CIFAR100, and Split TinyImageNet are reported in Tab. 1.

4.3 The Results on Complex Datasets

In this section, we evaluate the performance of various models on datasets with complex images. Specifically, we consider employing the ImageNet-R [65], the Caltech-UCSD Birds (CUB200) [62] and the Car196 [7], each of them containing samples from diverse categories. The classification results of various models on the complex datasets are reported in Table 2, which shows that the prompt-based continual learning methods achieve good performance on complex datasets. In addition, the proposed KAMD achieves the best performance compared to the state-of-the-art methods, demonstrating its effectiveness on dealing with complex data in continual learning.



(a) CIFAR10. (b) TinyImageNet.

Figure 3: The forgetting curve produced by various models.

4.4 Ablation Study

In this section, we construct a series of experiments to evaluate the performance of the proposed framework under different configurations.

The effect of the proposed graph structure. In order to investigate the effectiveness of each proposed module in continual learning, we consider creating several baselines. The first baseline is the proposed KAMO framework that does not use the recursive expansion process, called the KAMO-NoRecursive while the second baseline is the proposed KAMO framework that does not use the relation matrix to regulate all previously learned representations, namely the KAMO-NoRelation. The third baseline is the proposed KAMO framework that does not use the dual attention mechanism, namely the KAMO-NoDual. We employ the same hyperparameter configuration to train the proposed KAMO, the KAMO-Recursive, the KAMO-NoRelation, and the KAMO-NoDual on the Split CIFAR10. We provide the classification results in Figure 2a, which show that combining all proposed mechanisms into a unified framework achieves the best performance.

The computational complexity. We investigate the computational complexity of the proposed framework and other baselines. The training times (hours) of various models on the CIFAR10 are presented in Figure 2b, where 'Refresh' denotes DER+++Refresh. From the results, we observe that the proposed approach enjoys faster training compared to other baselines, including GSS and DER+++Refresh. The reason for this result is because the proposed approach only updates the current expert with a few parameters while freezing all previously learned experts, which is computationally efficient.

The forgetting analysis. We also investigate the forgetting effects of various models in continual learning. Specifically, we train various models on the CIFAR-10 and calculate the average classification error on all previously learned tasks after each task switch. We provide the results in Figure 3. The empirical results show that the proposed approach almost does not suffer from forgetting effects as the number of tasks increases. In contrast, other baseline models gradually suffer from the increased classification errors over time, leading to network forgetting.

The computation costs. We investigate the model's complexity by evaluating the computation costs during the whole training process. Specifically, we compare the three most popular baselines, including DER, DER++, and DEM, and the results are reported in Tab. 3. From the results, we can observe that the DER and DER++ [7] enjoy a fast training procedure due to their simple network architecture and optimization algorithm. Compared to the GEM [39], the proposed KAMO framework requires less training time. The main reason for these results is that the proposed KAMO framework only updates one expert with a few parameters at each new task learning, which avoids considerable computational costs.

5 Conclusion

This paper proposes a novel dynamic expansion framework, entitled the Knowledge-Aware Memory Optimization (KAMO), to deal with network forgetting in continual learning. We introduce an expandable expert graph attention mechanism to regulate all previously learned information in order to promote positive knowledge transfer learning when learning new information. A novel dual attention mechanism is proposed to promote the cooperation between all previously learned and the currently updated experts during training, further improving the model's performance. The empirical results on a series of datasets show that the proposed approach achieves state-of-the-art performance.

6 Acknowledgment

This paper is supported by the Sichuan Provincial Natural Science Foundation Project (No. 2025ZNSFSC0510) and the Open Fund of Key Laboratory of Large-scale Electromagnetic Industrial Software of Ministry of Education (No.EMCAE202504).

References

[1] A. Achille, T. Eccles, L. Matthey, C. Burgess, N. Watters, A. Lerchner, and I. Higgins. 2018. Life-long disentangled representation learning with cross-domain latent homologies. In *Advances in Neural Information Processing Systems (NeurIPS)*. 9873–9883.

[2] Hongjoon Ahn, Sungmin Cha, Donggyu Lee, and Taesup Moon. 2019. Uncertainty-based Continual Learning with Adaptive Regularization. In *Advances in Neural Information Processing Systems*. 4394–4404.

[3] R. Aljundi, M. Lin, B. Goujaud, and Y. Bengio. 2019. Gradient based sample selection for online continual learning. In *Advances in Neural Information Processing Systems (NeurIPS)*. 11817–11826.

[4] Jihwan Bang, Heesu Kim, YoungJoon Yoo, Jung-Woo Ha, and Jonghyun Choi. 2021. Rainbow Memory: Continual Learning with a Memory of Diverse Samples. In *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 8218–8227.

[5] Jihwan Bang, Hyunseo Koh, Seulki Park, Hwanjun Song, Jung-Woo Ha, and Jonghyun Choi. 2022. Online Continual Learning on a Contaminated Data Stream With Blurry Task Boundaries. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 9275–9284.

[6] Ari S Benjamin, David Rolnick, and Konrad Kording. 2018. Measuring and regularizing networks in function space. *arXiv preprint arXiv:1805.08289* (2018).

[7] Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. 2020. Dark experience for general continual learning: a strong, simple baseline. *Advances in Neural Information Processing Systems (NeurIPS)* 33 (2020), 15920–15930.

[8] Hyuntak Cha, Jaeho Lee, and Jinwoo Shin. 2021. Co2l: Contrastive continual learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 9516–9525.

[9] Arslan Chaudhry, Albert Gordo, Puneet Dokania, Philip Torr, and David Lopez-Paz. 2021. Using hindsight to anchor past knowledge in continual learning. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 35. 6993–7001.

[10] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. 2019. Efficient lifelong learning with A-GEM. In *International Conference on Learning Representations (ICLR), arXiv preprint arXiv:1812.00420*.

[11] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. 2019. Efficient lifelong learning with A-GEM. In *Int. Conf. on Learning Representations (ICLR), arXiv preprint arXiv:1812.00420*.

[12] A. Chaudhry, M. Rohrbach, M. Elhoseiny, T. Ajanthan, P. Dokania, P. H. S. Torr, and M.'A. Ranzato. 2019. On Tiny Episodic Memories in Continual Learning. *arXiv preprint arXiv:1902.10486* (2019).

[13] C. Cortes, X. Gonzalvo, V. Kuznetsov, M. Mohri, and S. Yang. 2017. Adanet: Adaptive structural learning of artificial neural networks. In *Proc. of Int. Conf. on Machine Learning (ICML), vol. PMLR 70*. 874–883.

[14] Danruo Deng, Guangyong Chen, Jianye Hao, Qiong Wang, and Pheng-Ann Heng. 2021. Flattening Sharpness for Dynamic Gradient Projection Memory Benefits Continual Learning. *Advances in Neural Information Processing Systems* 34 (2021), 18710–18721.

[15] Mohammad Mahdi Derakhshani, Xiantong Zhen, Ling Shao, and Cees Snoek. 2021. Kernel Continual Learning. In *International Conference on Machine Learning*. PMLR, 2621–2631.

[16] Shibhansh Dohare, J Fernando Hernandez-Garcia, Qingfeng Lan, Parash Rahman, A Rupam Mahmood, and Richard S Sutton. 2024. Loss of plasticity in deep continual learning. *Nature* 632, 8026 (2024), 768–774.

[17] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020).

[18] Arthur Douillard, Alexandre Ramé, Guillaume Couairon, and Matthieu Cord. 2022. DyTox: Transformers for continual learning with dynamic token expansion. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9285–9295.

[19] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. 2014. Generative adversarial nets. In *Proc. Advances in Neural Inf. Proc. Systems (NIPS)*. 2672–2680.

[20] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. *International Journal of Computer Vision* 129 (2021), 1789–1819.

[21] Yanan Gu, Xu Yang, Kun Wei, and Cheng Deng. 2022. Not Just Selection, but Exploration: Online Class-Incremental Continual Learning via Dual View Consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 7442–7451.

[22] Yiduo Guo, Bing Liu, and Dongyan Zhao. 2022. Online continual learning through mutual information maximization. In *International Conference on Machine Learning*. PMLR, 8109–8126.

[23] K. He, X. Zhang, S. Ren, and J. Sun. 2016. Deep residual learning for image recognition. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 770–778.

[24] G. Hinton, O. Vinyals, and J. Dean. 2014. Distilling the knowledge in a neural network. In *Proc. NIPS Deep Learning Workshop, arXiv preprint arXiv:1503.02531*.

[25] Yen-Chang Hsu, Yen-Cheng Liu, Anita Ramasamy, and Zsolt Kira. 2018. Re-evaluating continual learning scenarios: A categorization and case for strong baselines. *arXiv preprint arXiv:1810.12488* (2018).

[26] Ching-Yi Hung, Cheng-Hao Tu, Cheng-En Wu, Chien-Hung Chen, Yi-Ming Chan, and Chu-Song Chen. 2019. Compacting, Picking and Growing for Unforgetting Continual Learning. In *Advances in Neural Information Processing Systems*. 13647–13657.

[27] Fushuo Huo, Wenchao Xu, Jingcai Guo, Haozhao Wang, and Yunfeng Fan. 2024. Non-exemplar Online Class-Incremental Continual Learning via Dual-Prototype Self-Augment and Refinement. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 12698–12707.

[28] Saurav Jha, Dong Gong, He Zhao, and Lina Yao. 2024. NPCL: Neural processes for uncertainty-aware continual learning. *Advances in Neural Information Processing Systems* 36 (2024).

[29] Haeyong Kang, Rusty John Lloyd Mina, Sultan Rizky Hikmawan Madjid, Jaehong Yoon, Mark Hasegawa-Johnson, Sung Ju Hwang, and Chang D Yoo. 2022. Forget-free Continual Learning with Winning Subnetworks. In *International Conference on Machine Learning*. PMLR, 10734–10750.

[30] Ronald Kemker, Marc McClure, Angelina Abitino, Tyler Hayes, and Christopher Kanan. 2018. Measuring catastrophic forgetting in neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

[31] Junsu Kim, Hoseong Cho, Jihyeon Kim, Yihalem Yimolal Tiruneh, and Seungryul Baek. 2024. Sddgr: Stable diffusion-based deep generative replay for class incremental object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 28772–28781.

[32] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proc. Int. Conf. on Learning Representations (ICLR), arXiv preprint arXiv:1412.6980*.

[33] D. P. Kingma and M. Welling. 2013. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114* (2013).

[34] Alex Krizhevsky and Geoffrey Hinton. 2009. *Learning multiple layers of features from tiny images*. Technical Report. Univ. of Toronto.

[35] Ya Le and Xuan Yang. 2015. *Tiny imageNet visual recognition challenge*. Technical Report. Univ. of Stanford. 1–6 pages.

[36] Chun-Liang Li, Wei-Cheng Chang, Yu Cheng, Yiming Yang, and Barnabás Póczos. 2017. MMD GAN: Towards deeper understanding of moment matching network. In *Advances in Neural Information Processing Systems (NIPS)*, Vol. 30. 2200–2210.

[37] Z. Li and D. Hoiem. 2017. Learning without forgetting. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 40, 12 (2017), 2935–2947.

[38] Yan-Shuo Liang and Wu-Jun Li. 2023. Loss decoupling for task-agnostic continual learning. *Advances in Neural Information Processing Systems (NeurIPS)* 36 (2023), 11151–11167.

[39] David Lopez-Paz and Marc'Aurelio Ranzato. 2017. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*. 6467–6476.

[40] James Martens and Roger B. Grosse. 2015. Optimizing Neural Networks with Kronecker-factored Approximate Curvature. In *Proceedings of the 32nd International Conference on Machine Learning (ICML), JMLR*, Vol. 37. 2408–2417.

[41] Mark D. McDonnell, Dong Gong, Amin Parvaneh, Ehsan Abbasnejad, and Anton van den Hengel. 2023. RanPAC: Random Projections and Pre-trained Models for Continual Learning. In *Advances in Neural Information Processing Systems 36, article no. 526*. 12022–12053.

[42] Martin Menabue, Emanuele Frascaroli, Matteo Boschini, Enver Sangineto, Lorenzo Bonicelli, Angelo Porrello, and Simone Calderara. 2024. Semantic residual prompts for continual learning. In *European Conference on Computer Vision (ECCV), vol. LNCS 15119*. 1–18.

[43] Cuong V Nguyen, Yingzhen Li, Thang D Bui, and Richard E Turner. 2018. Variational continual learning. In *Proc. of Int. Conf. on Learning Representations (ICLR), arXiv preprint arXiv:1710.10628*.

[44] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter. 2019. Continual lifelong learning with neural networks: A review. *Neural Networks* 113 (2019), 54–71.

[45] R. Polikar, L. Upda, S. S. Upda, and Vasant Honavar. 2001. Learn++: An incremental learning algorithm for supervised neural networks. *IEEE Trans. on Systems Man and Cybernetics, Part C* 31, 4 (2001), 497–508.

[46] Biqing Qi, Xinquan Chen, Junqi Gao, Dong Li, Jianxing Liu, Ligang Wu, and Bowen Zhou. 2024. Interactive continual learning: Fast and slow thinking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12882–12892.

[47] J. Ramapuram, M. Gregorova, and A. Kalousis. 2017. Lifelong generative modeling. In *Proc. Int. Conf. on Learning Representations (ICLR), arXiv preprint arXiv:1705.09847*.

[48] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. 2017. iCaRL: Incremental classifier and representation learning. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2001–2010.

[49] B. Ren, H. Wang, J. Li, and H. Gao. 2017. Life-long learning based on dynamic combination model. *Applied Soft Computing* 56 (2017), 398–404.

[50] Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, and Gerald Tesauro. 2098. Learning to learn without forgetting by maximizing transfer and minimizing interference. In *International Conference on Learning Representations (ICLR), arXiv preprint arXiv:1810.11910.*

[51] Hippolyt Ritter, Aleksandar Botev, and David Barber. 2018. Online Structured Laplace Approximations for Overcoming Catastrophic Forgetting. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 31. 3742–3752.

[52] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. 2016. Progressive neural networks. *arXiv preprint arXiv:1606.04671* (2016).

[53] Yujun Shi, Li Yuan, Yunpeng Chen, and Jiashi Feng. 2021. Continual learning via bit-level information preserving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 16674–16683.

[54] H. Shin, J. K. Lee, J. Kim, and J. Kim. 2017. Continual learning with deep generative replay. In *Advances in Neural Inf. Proc. Systems (NIPS).* 2990–2999.

[55] James Seale Smith, Leonid Karlinsky, Vyshnavi Gutta, Paola Cascante-Bonilla, Donghyun Kim, Assaf Arbelle, Rameswar Panda, Rogério Feris, and Zsolt Kira. 2023. CODA-Prompt: COntinual Decomposed Attention-Based Prompting for Rehearsal-Free Continual Learning. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR.* 11909–11919.

[56] Michalis K Titsias, Jonathan Schwarz, Alexander G de G Matthews, Razvan Pascanu, and Yee Whye Teh. 2019. Functional Regularisation for Continual Learning with Gaussian Processes. In *Proc. Int. Conf. on Learning Represenations (ICLR), arXiv preprint arXiv:1901.11356.*

[57] Rishabh Tiwari, Krishnateja Killamsetty, Rishabh Iyer, and Pradeep Shenoy. 2022. GCR: Gradient Coreset Based Replay Buffer Selection for Continual Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).* 99–108.

[58] Ilya O Tolstikhin, Bharath K Sriperumbudur, and Bernhard Schölkopf. 2016. Minimax estimation of maximum mean discrepancy with radial kernels. In *Advances in Neural Information Processing Systems (NIPS)*, Vol. 29. 1930–1938.

[59] Gido M van de Ven and Andreas S Tolias. 2018. Three continual learning scenarios. In *NeurIPS Continual Learning Workshop*, Vol. 1.

[60] Vinay Kumar Verma, Kevin J Liang, Nikhil Mehta, Piyush Rai, and Lawrence Carin. 2021. Efficient feature transformations for discriminative and generative continual learning. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).* 13865–13875.

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

[62] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. 2011. *Caltech-UCSD Birds-200-2011.* Technical Report CNS-TR-2011-001. California Institute of Technology.

[63] Shipeng Wang, Xiaorong Li, Jian Sun, and Zongben Xu. 2021. Training networks in null space of feature covariance for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 184–193.

[64] Zhenyi Wang, Yan Li, Li Shen, and Heng Huang. 2024. A Unified and General Framework for Continual Learning. In *International Conference on Learning Representations (ICLR), arXiv preprint arXiv:2403.13249.*

[65] Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer G. Dy, and Tomas Pfister. 2022. DualPrompt: Complementary Prompting for Rehearsal-Free Continual Learning. In *Proc. Eucopean Conference on Computer Vision (ECCV), vol. LNCS 13686.* 631–648.

[66] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. 2022. Learning to prompt for continual learning. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 139–149.

[67] Yeming Wen, Dustin Tran, and Jimmy Ba. 2020. BatchEnsemble: an Alternative Approach to Efficient Ensemble and Lifelong Learning. In *Proc. Int. Conf. on Learning Representations (ICLR), arXiv preprint arXiv:2002.06715.*

[68] Mengqi Xue, Haofei Zhang, Jie Song, and Mingli Song. 2022. Meta-attention for ViT-backed continual learning. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 150–159.

[69] Jiazuo Yu, Yunzhi Zhuge, Lu Zhang, Ping Hu, Dong Wang, Huchuan Lu, and You He. 2024. Boosting continual learning of vision-language models via mixture-of-experts adapters. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 23219–23230.

[70] M. Zhai, L. Chen, F. Tung, J He, M. Nawhal, and G. Mori. 2019. Lifelong GAN: Continual Learning for Conditional Image Generation. In *Proc. of the IEEE/CVF Int. Conf. on Computer Vision (ICCV).* 2759–2768.

[71] Da-Wei Zhou, Han-Jia Ye, De-Chuan Zhan, and Ziwei Liu. [n. d.]. Revisiting Class-Incremental Learning with Pre-Trained Models: Generalizability and Adaptivity are All You Need. In *International Conference on Learning Representations (ICLR), arXiv preprint arXiv:2303.07338.*

[72] Guanyu Zhou, Kihyuk Sohn, and Honglak Lee. 2012. Online incremental feature learning with denoising autoencoders. In *Proc. Int. Conf. on Artificial Intelligence and Statistics (AISTATS), vol. PMLR 22.* 1453–1461.