

## **TASK-AWARE MULTI-EXPERT ARCHITECTURES FOR LIFELONG DEEP LEARNING**

Jianyu Wang<sup>1</sup>, Jacob Nean-Hua Sheikh<sup>2</sup>, Cat P. Le<sup>3</sup>, and Hoda Bidkhori<sup>1</sup>

<sup>1</sup>Dept. of Computational and Data Sciences, George Mason University, Fairfax, VA, USA

<sup>2</sup>Dept. of Computer Science, George Mason University, Fairfax, VA, USA

<sup>3</sup>Dept. of Electrical and Computer Engineering, Duke University, Durham, NC, USA

### **ABSTRACT**

Lifelong deep learning (LDL) enables neural networks to learn continuously across tasks while preserving prior knowledge. We propose Task-Aware Multi-Expert (TAME), a novel algorithm that leverages task similarity to guide expert model selection and knowledge transfer. TAME maintains a pool of pretrained neural networks, activating the most relevant expert for each new task. A shared dense layer integrates the selected expert’s features to generate predictions. To mitigate catastrophic forgetting, TAME employs a replay buffer storing representative samples and embeddings from past tasks, enabling continual reference. An attention mechanism further prioritizes the most relevant stored knowledge for each task. Overall, the TAME algorithm supports flexible and adaptive learning across diverse scenarios. Experiments on CIFAR-100-derived classification tasks show that TAME improves classification accuracy while maintaining performance on earlier tasks, demonstrating its effectiveness in balancing adaptation and retention in evolving task sequences.

### **1 INTRODUCTION**

Lifelong deep learning is a rapidly evolving area focused on empowering deep neural networks to continually learn new tasks while retaining previously acquired knowledge and adapting to novel situations. Unlike traditional ensemble learning, which aims to improve accuracy by combining multiple models for the same task, and multi-task learning (MTL), which learns multiple tasks jointly, lifelong deep learning emphasizes continual adaptation and knowledge retention as tasks arrive sequentially. A key challenge in this domain is catastrophic forgetting, where models lose performance on earlier tasks when trained sequentially—an issue particularly evident in online or real-time systems. For instance, a traffic sign detection system may degrade in recognizing previously learned signs when updated with new ones. While current methods often employ replay mechanisms or regularization strategies, these solutions may struggle to scale in complex environments or fully exploit shared knowledge across tasks. To overcome these limitations, we propose a novel framework that dynamically integrates relevant insights from multiple models, enhancing both knowledge retention and task adaptability. This is especially advantageous in simulation-based systems, where ongoing adaptation to changing conditions is essential.

In this paper, we present TAME (Task-Aware Multi-Expert), a lifelong learning algorithm designed to enable multiple deep learning models to incrementally and collaboratively learn over time. Central to our approach is the use of task similarity to inform expert selection and guide knowledge transfer. By identifying the relationships between the current task and previously encountered ones, TAME enables selective reuse of relevant knowledge, thereby improving learning efficiency and mitigating redundancy. For example, when a new task closely resembles a prior one, the most appropriate expert model is selected to extract transferable features, promoting effective adaptation while reducing forgetting. To further address the challenge of catastrophic forgetting, TAME incorporates a finite-capacity replay buffer that stores representative samples and their corresponding feature embeddings from earlier tasks. This allows the system to revisit and reinforce prior knowledge during training. We also introduce a scaled dot-product

attention mechanism that dynamically prioritizes the most relevant past experiences in the buffer. To support long-range dependency learning, the replay buffer retains representative samples from all previous tasks using a balanced replacement strategy, ensuring early task information remains accessible. While attention highlights relevant memories, the buffer preserves them—enabling sustained generalization and temporal continuity. The key contributions of this work are as follows:

- We propose a Task-Aware Multi-Expert lifelong learning algorithm designed to improve adaptability and knowledge retention across sequential tasks. The model architecture consists of multiple pretrained neural networks interconnected with a shared dense layer (*SDL*) for the final output. When a new task arrives, TAME uses task similarity—measured via *Fréchet Inception Distance (FID)* or *Cosine Similarity*—to select the most appropriate model. This model extracts feature representations from the new task, which are then stored in a replay buffer and passed to the *SDL* for learning while preserving the expert models to prevent forgetting,
- We further enhance the algorithm to incorporate a scaled dot-product attention mechanism when our model retrieves knowledge from the replay buffer. Instead of treating all stored features equally, we compute attention scores between the current task and stored past tasks to dynamically prioritize more relevant experiences. This targeted weighting of memory improves feature reuse and further enhances performance and knowledge retention.
- We design a tailored experimental setup using multiple expert architectures connected to a shared dense layer, applied to binary classification tasks derived from the CIFAR-100 dataset. Our framework supports controlled evaluation across multiple task sequences, enabling consistent assessment of task similarity-based strategies in high-dimensional image domains. We have also conducted an experiment to compare TAME and a classical multi-task learning approach, the Shared-Bottom model. Our results show that our proposed algorithm, compared with baselines that do not utilize task similarity, performs better in terms of Average Forgetting and Average AUROC score.

TAME can be adapted to digital twins and traffic simulation by enabling real-time adaptation to evolving conditions. In smart manufacturing, it allows digital twins to incrementally learn new equipment behaviors—such as emerging faults or wear patterns—by selecting relevant expert models based on sensor data similarity, supporting predictive maintenance without retraining. In traffic simulation, TAME can be adapted to changing dynamics across regions or times of day by reusing expert models trained on similar patterns. TAME is also compatible with datasets from stochastic simulations, where each scenario is treated as a task, making it well-suited for simulation-informed digital twins and other real-time environments.

## 2 RELATED WORK

*Lifelong deep learning*, also referred to as *continual deep learning*, enables models to acquire new knowledge over time while preserving previously learned information. Unlike traditional deep learning models trained on fixed datasets, LDL models must handle a sequence of tasks and adapt dynamically. A fundamental challenge in this setting is *catastrophic forgetting*, where learning a new task disrupts previously acquired knowledge (Parisi et al. 2021). To address this issue, five primary classes of methods have been proposed: regularization-based, replay-based, architectural expansion, optimization, and representation methods (Wang 2024). Regularization-based methods constrain weight updates to retain previously important parameters (e.g. Elastic Weight Consolidation (Kirkpatrick et al. 2017)). Replay-based methods intelligently store examples from earlier tasks and interleave them with new data during training (e.g. Experience Replay (Shin et al. 2017)). Architectural expansion approaches, such as Dynamically Expandable Networks (Yoon et al. 2018), allocate new capacity for new tasks. Representation methods explore how the feature vectors generated by networks might be shared across tasks. Optimization methods explore the objective function used to train the network, and how this function might best prevent catastrophic forgetting.

*Task Similarity* is a quantitative measure of how similar two different tasks are by some specific metric. Task similarity plays an important role across machine learning literature, such as in few-shot learning and meta-learning (Le et al. 2022; Le et al. 2022; Kötter et al. 2024), multi-task learning (Shui et al. 2019), and causal inference (Aloui et al. 2023). Task similarity-based approaches have been shown to improve knowledge transfer between tasks in deep learning applications (Upadhyay et al. 2024). Task similarity can be measured in a variety of ways, depending on the specific domain. In this paper, we use Fréchet Inception Distance (FID), and Cosine Similarity (Heusel et al. 2017). Recent studies have explored the role of task similarity in lifelong learning, particularly focusing on similarities in weight differences between teacher and student networks. For instance, the teacher-student continual learning framework proposed in Lee et al. (2021) employs a two-layer nonlinear neural network, where task similarity is measured based on the models’ weights. In this setup, the distance between tasks is quantified by comparing the weight differences between the teacher and student models. Their findings reveal that catastrophic forgetting is most pronounced when tasks exhibit intermediate similarity. Additionally, the weight regularization-based task similarity framework proposed in Hiratani (2024) demonstrates an improvement in knowledge retention in low-dimensional latent structures. Notably, when readout similarity (i.e., similarity in weights from hidden to output layers) is low, even high feature similarity (i.e., similarity in weights from input to hidden layers) can lead to interference during learning—causing the network to misuse shared representations across tasks. While these approaches provide valuable insight for shallow neural networks, they fall short in capturing the complexities of deeper architectures and do not generalize well in real-world applications where multiple models collaborate simultaneously at large scales.

In this work, our primary objective is to demonstrate the impact of leveraging task similarity in lifelong deep learning. Our task-aware multi-expert framework employs feature-based similarity measures along with attention-guided replay to dynamically select and reuse relevant expert models. We show that incorporating task similarity significantly enhances adaptive knowledge retention across tasks in lifelong learning settings.

### 3 PROPOSED METHOD

In this section, we introduce the Task-Aware Multi-Expert (TAME) lifelong learning algorithm and its attention-enhanced algorithm (AE-TAME). Here, the proposed architecture consists of  $n$  Convolutional Neural Networks (*CNNs*), each interconnected via a shared fully connected component, referred to as the *Shared Dense Layer* (*SDL*). A detailed overview of the proposed framework is provided in Figure 1.

#### 3.1 Task-Aware Multi-Expert Lifelong Learning Algorithm

We present the algorithm in the context of image classification. TAME is designed to address the challenge of lifelong learning by selectively reusing pretrained expert models based on task similarity. Its architecture consists of three core components:

- A pool of expert models ( $CNN_i$ ), each pretrained on a distinct source task.
- A shared dense layer (*SDL*) that receives extracted features from the selected expert and performs task-specific classification.
- A fixed-capacity replay buffer that stores representative samples and their embeddings from previously seen tasks to mitigate catastrophic forgetting.

The  $i^{th}$  *CNN*,  $CNN_i, 0 < i \leq n$ , is initially pretrained on its corresponding initial task  $i$ . Following this pretraining phase, the model enters a lifelong learning phase, where it is sequentially exposed to new tasks.

When a new task  $T_t$  arrives, the algorithm computes its similarity to each initial task  $\hat{T}_i$  and selects the expert  $CNN_i$  associated with the most similar initial task. The selected *CNN* generates a task-specific feature representation  $\beta(T_t, i^*)$ , which is used for downstream prediction and stored in a fixed-capacity replay buffer for future reference. Unlike traditional ensemble methods, only the shared dense layer (*SDL*)

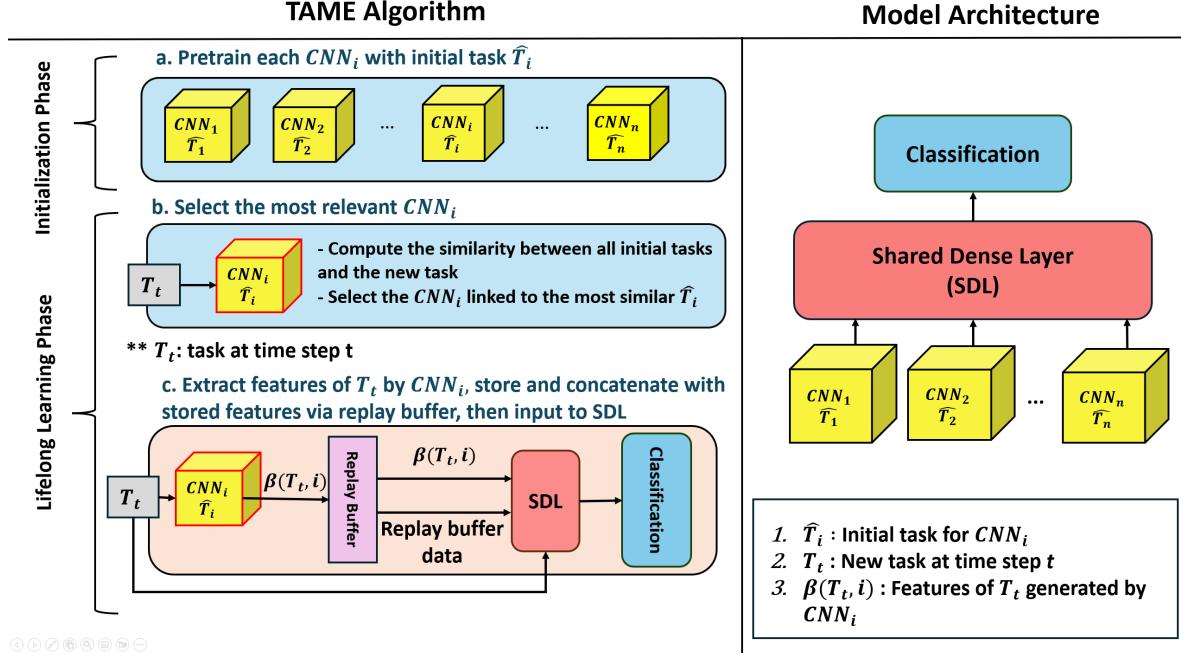


Figure 1: **Overview of TAME.** **Left: TAME algorithm procedure:** a) Each  $CNN$  is pretrained on an initial task. b) When a new task arrives, its similarity to initial tasks is computed. The most similar  $CNN$  is selected. c) The selected  $CNN$  extracts features of the new task, stores them in the replay buffer, and—if past data is available—feeds both new and stored features to the  $SDL$  for classification. **Right: Model Architecture:** Multiple  $CNN$ s connect to a shared dense layer ( $SDL$ ), which performs the final classification.

is updated during lifelong learning, while the expert  $CNN$ s remain fixed, allowing the system to accumulate and preserve expert knowledge efficiently. The algorithmic details are described as follows:

1. **Initialization Phase.** During this phase, we initialize  $n$  expert  $CNN$ s and a Shared Dense Layer ( $SDL$ ) with random parameters. A replay buffer is also initialized as a dictionary indexed by  $CNN_i$ , where  $i = 1, \dots, n$ , with each entry storing image samples, labels, and feature representation. Each  $CNN_i$  is independently pretrained on a distinct initial task  $\hat{T}_i$ , and its parameters are frozen afterward to preserve learned representations.
2. **Lifelong Learning Phase.** When a new task  $T_t$  arrives at time step  $t$ , the algorithm executes the following steps:
  - *Step 1: Task Similarity-Based Model Selection.* TAME computes the similarity between the incoming task  $T_t$  and all initial tasks  $\hat{T}_1, \dots, \hat{T}_n$ . This is done using metrics such as: *Fréchet Inception Distance (FID)* and *Cosine Similarity*. FID quantifies the distance between the feature distributions of the current task and the previously learned initial tasks, where lower values indicate higher similarity. Cosine similarity, in contrast, measures the cosine of the angle between two feature vectors, with higher values indicating greater similarity. The expert model  $CNN_{i^*}$  associated with the most similar source task is selected. The chosen model is used to generate a feature embedding  $\beta(T_t, i^*)$  for the new task. Full definitions and implementation details of both measures are provided in Appendix A.1 and Appendix A.2.
  - *Step 2: Replay Learning.* We employ a replay buffer as a memory module with a fixed capacity. It removes the oldest entries once full to retain recent information. As  $T_t$  arrives at time  $t$ , its feature representation  $\beta(T_t, i^*)$ , raw images  $Img(T_t)$  and task label  $L_t$  are stored under the selected  $CNN_{i^*}$ . If replay data is available, the buffer retrieves all previous feature representations

$\beta(T_j, i^*)$  using  $CNN_i^*$ , and their raw images  $Img(T_j)$ ,  $1 \leq j \leq t-1$ . It concatenates them with  $\beta(T_t, i^*)$  and  $Img(T_t)$ , respectively to form combined feature vector  $\theta_i$  and  $Img$ :

$$\theta_i = \begin{bmatrix} \beta(T_t, i^*) \\ \beta(T_j, i^*) , 1 \leq j \leq t-1 \end{bmatrix}, Img = \begin{bmatrix} Img(T_t) \\ Img(T_j) \end{bmatrix},$$

which—along with the label of all stored task  $L_j$ ,  $1 \leq j \leq t-1$  and the new task label  $L_t$ —is used to update the parameters of the *SDL* via backpropagation. If  $T_t$  is the first task for  $CNN_{i^*}$ , only  $Img(T_t)$ , label  $L_t$  and feature  $\beta(T_t, i^*)$  are used, refer to Algorithm 1.

---

**Algorithm 1:** Task-aware multi-expert lifelong deep learning algorithm (TAME).

---

**Input:** Initialization tasks:  $\hat{T}_1, \dots, \hat{T}_n$ ; Lifelong tasks:  $T_1, \dots, T_t$   
**Output:** Classification decision  $O$

**Function** Initialization:

- Initialize a replay buffer
- for**  $\hat{T}_i$ ,  $1 \leq i \leq n$  **do**
  - Train  $CNN_i$  with  $\hat{T}_i$
  - Freeze  $CNN_i$
- return**  $CNN$

**Function** LifelongLearning:

- for** incoming task  $T_t$  **do**
  - Compute similarity  $d(T_t, \hat{T}_i)$  for all  $i$ ,  $1 \leq i \leq n$  ▷ Step 1: Task Similarity-Based Selection
  - Select  $CNN_{i^*}$  with highest similarity to generate feature  $\beta(T_t, i^*)$
  - Store the raw images of  $T_t$ , denoted as  $Img(T_t)$ , new task label  $L_t$ , and  $\beta(T_t, i^*)$  in replay buffer
  - if** replay data is available ▷ Step 2: Replay-Based Learning
  - then**
    - Retrieve stored features  $\beta(T_j, i^*)$  from the replay buffer for  $1 \leq j \leq t-1$
    - Obtain combined feature vector  $\theta_i$  and combined image vector  $Img$
    - Pass  $Img$ , new task label  $L_t$ , past task labels  $L_j$ ,  $1 \leq j \leq t-1$ , and  $\theta_i$  to *SDL*
  - else**
    - Pass  $Img(T_t)$ , new task label  $L_t$ , and  $\beta(T_t, i^*)$  to *SDL*
- return**  $O$

---

To evaluate the importance of task similarity, we design a baseline model that shares the same architecture and replay mechanism as TAME algorithm. However, rather than selecting the most similar  $CNN_{i^*}$ , the baseline randomly assigns each incoming task  $T_t$  to one of the  $n$  *CNNs*, regardless of similarity.

### 3.2 Attention-Enhanced Task-Aware Multi-Expert Lifelong Learning Algorithm

We implement an attention mechanism in the TAME algorithm, while preserving the original architecture and replay buffer, referring to the enhanced model as AE-TAME. Inspired by scaled dot-product attention (Vaswani et al. 2017), AE-TAME dynamically assigns weights to stored task features based on their relevance to the current task, enabling the model to focus on more informative representations. The attention-enhanced features are then passed to the Shared Dense Layer (*SDL*) for final prediction.

Attention works by comparing a *Query*—the current task features—to a set of features *Keys* representing past tasks, and weighting the corresponding *Values* based on attention weight.

Our algorithm still consists of two phases. The *Initialization Phase* and *Lifelong learning phase*. The *Initialization Phase* remains unchanged. In the *Lifelong learning phase*, Step 1—*Task Similarity-Based*

*Model Selection*—is also unchanged. The change happen in Step 2—*Replay Learning*. This step incorporates an attention mechanism when the selected  $CNN_{i^*}$  retrieves stored feature in the replay buffer to enhance knowledge retention and task adaptability. In this step, we define the standard attention components as follows:

- $Q_t$  (*Query*): The feature representation  $\beta(T_t, i^*)$  generated by  $CNN_{i^*}$ .
- $K_j$  (*Key*): The feature representations stored in the replay buffer for  $CNN_{i^*}$  for previous lifelong learning tasks,  $1 \leq j \leq t-1$ .
- $V_t$  (*Value*): A weighted sum of the stored task features, computed using attention weights  $\alpha_j$ :

$$V_t = \sum_{j=1}^{t-1} \alpha_j \cdot \beta(T_j, i^*).$$

where each  $\alpha_j$  is derived from the scaled dot-product attention weight. For the details, please refer to Appendix A.3.

When a new task  $T_t$  arrives at time step  $t$ , the selected  $CNN_{i^*}$  first generates its feature representation  $\beta(T_t, i^*)$ . The attention mechanism then computes a weighted sum  $V_t$  of the stored features  $\beta(T_j, i^*)$ ,  $1 \leq j \leq t-1$ , retrieved from the replay buffer. These are combined to form the attention-enhanced representation  $\gamma_t$ :

$$\gamma_t = [\beta(T_t, i^*), V_t].$$

Finally,  $\gamma_t$ , the combined image vector  $Img$ , the current task label  $L_t$ , and the stored task labels  $L_j$ ,  $1 \leq j \leq t-1$ , are passed to the Shared Dense Layers (*SDL*) for prediction, as detailed in Algorithm 2.

**Baseline for Evaluation of AE-TAME.** We design a baseline model that shares the same architecture, replay mechanism, and attention mechanism as the AE-TAME algorithm, referred to as the Attention-Enhanced Baseline. The only difference is that it does not use task similarity measures. This baseline randomly assigns each incoming task  $T_t$  to one of the  $n$  *CNN*, regardless of similarity.

## 4 EXPERIMENTAL RESULTS

In this section, we demonstrate the effectiveness of TAME and AE-TAME in comparison to baseline algorithms. As discussed in the previous section, the baseline algorithms maintain the same architectural structure. The key difference lies in the expert selection step, where the baseline methods select a *CNN* at random rather than leveraging task similarity for informed model selection.

### 4.1 Experimental Setup and Evaluation Metrics

In this subsection, we present the experimental setup and evaluation metrics.

- **Dataset and Task Formulation.** In machine learning, a "task" refers to a well-defined objective that the learning algorithm aims to optimize based on input-output mappings derived from data. For instance, in an "image classification task", the goal is to categorize images into predefined classes based on their visual features. In our experiment, each task is a binary classification problem. They are derived from the CIFAR-100 dataset, which contains 60,000 color images across 100 categories. Each initial task (for pretraining) consists of 400 images (200 per class), trained for 5 epochs. During lifelong learning phase, 10 new binary tasks are introduced incrementally, each containing 200 images. We evaluate performance across 5 distinct sequences (Sequence 1–5), each comprising 10 unseen binary tasks.
- **Model Architecture.** For TAME, AE-TAME, and the baselines, our model architecture is the same and consists of five convolutional neural networks (*CNN*), each comprising three convolutional

---

**Algorithm 2:** Attention-enhanced TAME algorithm (AE-TAME).

---

**Input:** Initialization tasks:  $\hat{T}_1, \dots, \hat{T}_n$ ; Lifelong tasks:  $T_1, \dots, T_t$   
**Output:** Classification decision  $O$

**Function** Initialization:

```

for  $\hat{T}_i, 1 \leq i \leq n$  do
    | Train  $CNN_i$  with  $\hat{T}_i$ 
    | Freeze  $CNN_i$ 
return  $CNN$ 

```

**Function** LifelongLearning:

```

for incoming task  $T_t$  do
    | Compute similarity  $d(T_t, \hat{T}_i)$  for all  $i, 1 \leq i \leq n$  ;            $\triangleright$  Step 1: Task Similarity-Based Selection
    | Select  $CNN_{i^*}$  with highest similarity score to generate feature  $\beta(T_t, i^*)$ 
    | Store the raw images of  $T_t$ , denoted as  $Img(T_t)$ , new task label  $L_t$ ,  $\beta(T_t, i^*)$  in replay buffer
    | if replay data exists ;
        | | then
            | | | Retrieve stored features  $\beta(T_j, i^*)$  from the replay buffer for  $1 \leq j \leq t-1$ 
            | | | Compute attention weights  $\alpha_j$ 
            | | | Compute  $V_t = \sum \alpha_j \cdot \beta(T_j, i^*)$ 
            | | | Obtain combined attention-enhanced representation  $\gamma_t$ 
            | | | Pass combined image vector  $Img$ , new task label  $L_t$ , stored task labels  $L_j$ , and  $\gamma_t$  to  $SDL$ 
    | | else
        | | | Pass  $Img(T_t)$ , new task label  $L_t$ , and  $\beta(T_t, i^*)$  to  $SDL$ 
return  $O$ 

```

---

layers with ReLU activations and max-pooling, followed by two fully connected layers. Each  $CNN$  contains approximately 180K parameters. These  $CNNs$  are connected to a Shared Dense Layer ( $SDL$ ), which includes five fully connected layers with ReLU activations and a sigmoid output for binary classification, totaling around 720K parameters. A replay buffer with a fixed maximum capacity of 1,000 images is employed to retain past task information for lifelong learning phase.

- **Lifelong Learning Phase.** For each task, only the  $SDL$  is updated using Binary Cross Entropy (BCE) loss over three training epochs, with the Adam optimizer (learning rate = 0.001, batch size = 32). Task similarity is computed using either Fréchet Inception Distance (FID) or Cosine Similarity. Based on the similarity score, the most appropriate  $CNN$  is selected to process the new task. A baseline variant assigns the new tasks to a randomly selected  $CNN$  without using similarity measures.
- **Evaluation Metrics.** We evaluate and benchmark the performance of our algorithms using two standard metrics: For each task sequence, *Average AUROC* (*A-AUROC*) is computed by averaging the AUROC (Aread Under the Receiver Operating Characteristic Curve) scores of all tasks in the sequence, reflecting the model’s overall ability to distinguish between classes. *Average Forgetting* (*AF*) measures knowledge degradation by averaging the difference between each task’s highest and final recorded accuracy, as detailed in Appendix A.4 and Appendix A.5.
- **Runtime and Computational Setup.** All experiments were conducted on a GPU cluster with a single GPU featuring 14 GB of memory. Under the described experimental setup, TAME and AE-TAME each required approximately 40 minutes on average to complete  $CNN$  pretraining, model training, and evaluation per trial. Although the baseline models ran slightly faster, the difference in runtime was not significant.

Table 1: Performance of all algorithms using FID similarity in terms of average forgetting and average AUROC across five task sequences.

Sequence	Baseline		AE-Baseline		TAME		AE-TAME	
	AF	A-AUROC	AF	A-AUROC	AF	A-AUROC	AF	A-AUROC
Seq 1.	0.0815	0.4869	0.099	0.4871	0.067	0.5080	<b>0.0185</b>	<b>0.5580</b>
Seq 2.	0.0964	0.4982	0.0625	0.4704	<b>0.0051</b>	0.5114	0.028	<b>0.5643</b>
Seq 3.	0.072	<b>0.5350</b>	0.2235	0.4443	0.06	0.4521	<b>0.052</b>	0.5167
Seq 4.	0.075	0.4975	0.0675	0.4369	0.072	0.4017	<b>0.019</b>	<b>0.4993</b>
Seq 5.	0.0777	0.4975	0.0593	0.4512	<b>0.0536</b>	<b>0.5622</b>	0.052	0.5022

#### 4.2 Performance and Comparison of TAME and AE-TAME and Baselines

Table 1 compares all algorithms utilizing the FID similarity measure. AE-TAME outperforms both Baseline and AE-Baseline in terms of Average Forgetting (AF) across all five sequences, and achieves higher Average AUROC (A-AUROC) in four out of five cases, confirming its strong performance when leveraging FID-based task similarity. TAME also outperforms Baseline and AE-Baseline in five out of five sequences for AF and in three out of five sequences for A-AUROC, demonstrating that task similarity—even without the attention mechanism—significantly improves knowledge retention and classification accuracy. Furthermore, AE-TAME surpasses TAME in three out of five sequences in terms of AF and in four out of five sequences for A-AUROC. These results collectively highlight the benefits of incorporating task similarity into the model, and further show that performance is additionally enhanced when the attention mechanism is employed.

Table 2 compares all algorithms utilizing the Cosine Similarity measure. The results show that AE-TAME consistently achieves lower forgetting than both Baseline and AE-Baseline across four out of five sequences and outperforms them in A-AUROC also in four out of five cases. TAME exceeds the performance of Baseline and AE-Baseline in three out of five sequences for AF and in two out of five for A-AUROC. When comparing AE-TAME to TAME, AE-TAME demonstrates lower forgetting in three out of five sequences and achieves higher A-AUROC in five out of five. These findings indicate that while both task similarity metrics are beneficial, Cosine Similarity proves particularly effective when combined with attention, especially in enhancing AUROC. Overall, AE-TAME remains the top-performing model across both task similarity metrics and evaluation criteria.

Table 2: Performance of all algorithms using cosine similarity in terms of average forgetting and average AUROC across five task sequences.

Sequence	Baseline		AE-Baseline		TAME		AE-TAME	
	AF	A-AUROC	AF	A-AUROC	AF	A-AUROC	AF	A-AUROC
Seq 1.	0.0815	0.4869	0.099	0.4871	<b>0.0195</b>	0.5345	0.026	<b>0.5557</b>
Seq 2.	0.0964	0.4982	0.0625	0.4704	0.0723	0.5028	<b>0.0445</b>	<b>0.5807</b>
Seq 3.	0.072	0.5350	0.2235	0.4443	<b>0.027</b>	0.5176	0.076	<b>0.5583</b>
Seq 4.	0.075	<b>0.4975</b>	0.0675	0.4369	0.083	0.4268	<b>0.0095</b>	0.4591
Seq 5.	0.0777	0.4975	0.0593	0.4512	0.1012	0.4268	<b>0.0523</b>	<b>0.5785</b>

##### 4.2.1 Additional Illustration for Accuracy

For additional illustration, Figures 2 visualize the performance of TAME and AE-TAME across one particular sequence of 10 binary classification tasks derived from CIFAR-100, showing task-specific accuracy. In Figure 2a, TAME utilizing FID and TAME utilizing Cosine Similarity show stronger and more stable accuracy trends than the baseline. TAME utilizing FID achieves higher accuracy in later tasks, while TAME utilizing Cosine Similarity is competitive in early and middle tasks. The baseline shows lower and

more volatile accuracy. In Figure 2b, AE-TAME using FID consistently outperforms the baseline across most tasks, while AE-TAME using Cosine Similarity achieves the highest accuracy, peaking around 70%. The AE baseline remains inferior, confirming the advantage of combining attention mechanisms with task similarity for robust and consistent accuracy across tasks.

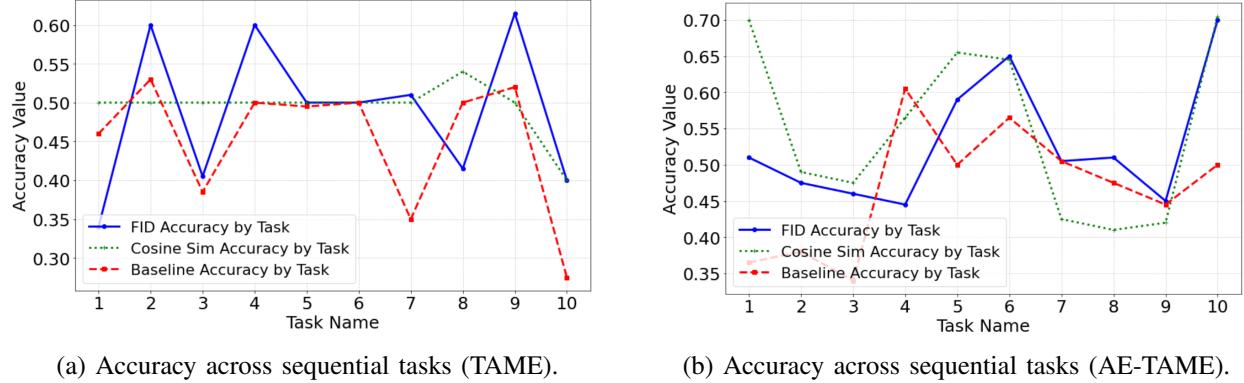


Figure 2: Accuracy performance of TAME and AE-TAME across tasks for one sequence from CIFAR-100.

### 4.3 Comparison with the Shared-Bottom model

In this subsection, we compare the average forgetting performance of TAME and AE-TAME with a traditional multi-task learning baseline—the Shared-Bottom model (Zhang and Yang 2022). The Shared-Bottom model uses a shared *CNN* encoder and task-specific binary classification heads, with a structure and parameter size designed to match that of TAME. This configuration ensures a fair comparison in terms of model capacity and architectural complexity. We adopt the same experimental protocol as with TAME. The model is optimized using Adam with a learning rate of 0.001, trained for 3 epochs per task with a batch size of 32, and evaluated after each task to compute average forgetting.

As shown in Table 3, both AE-TAME and TAME completely outperforms the Shared-Bottom model across all sequences. The Shared-Bottom model, while conceptually efficient, suffers from greater forgetting due to its inability to isolate task-specific representations during continual updates.

Table 3: Average forgetting of TAME, AE-TAME, and shared-bottom model across five task sequences.

Sequence	TAME (FID)	TAME (Cosine)	AE-TAME (FID)	AE-TAME (Cosine)	Shared-Bottom
Seq 1.	0.0670	0.0195	<b>0.0185</b>	0.0260	0.2111
Seq 2.	<b>0.0051</b>	0.0723	0.0280	0.0445	0.1972
Seq 3.	0.0600	<b>0.0270</b>	0.0520	0.0760	0.2083
Seq 4.	0.0720	0.0830	0.0190	<b>0.0095</b>	0.1947
Seq 5.	0.0536	0.1012	<b>0.0520</b>	0.0523	0.1573

## 5 CONCLUSION

We presented a Task-Aware Multi-Expert lifelong learning framework that leverages task similarity and a multi-expert *CNN* architecture to facilitate incremental and collaborative learning. By dynamically selecting the most relevant expert model based on task similarity metrics, our approach enhances both prediction accuracy and knowledge retention. Additionally, the integration of an attention mechanism in AE-TAME further improves the reuse of past knowledge by focusing on the most relevant task representations.

While this work focuses on image-based tasks and a single data modality, it can be adapted to a broad range of real-world applications involving multimodal data. TAME’s modular design makes it well suited

for such extensions, enabling specialized expert models to handle different modalities. In future work, we aim to improve the scalability of the framework and incorporate task similarity to enhance other lifelong learning algorithms.

## ACKNOWLEDGMENT

This research was supported by the ONR grant N000142312629.

## A APPENDIX

### A.1 Fréchet Inception Distance (FID)

Given two tasks—Task 1 (new task) and Task 2 (initial task)—the Fréchet Inception Distance (FID) quantifies the similarity between their feature distributions by measuring the distance between two multivariate Gaussians (Heusel et al. 2017). It is computed as:

$$FID = \|\mu_1 - \mu_2\|^2 + \text{Tr}(\Sigma_1 + \Sigma_2 - 2(\Sigma_1 \Sigma_2)^{1/2}),$$

where:

- $\mu_1, \mu_2$  are the mean feature vectors of Task 1 and Task 2, respectively. For a given task  $T$ , the mean vector  $\mu_T$  is computed as:

$$\mu_T = \frac{1}{N} \sum_{i=1}^N f(x_i),$$

where  $f(x_i)$  is the extracted feature vector for image  $x_i$  and  $N$  is the total number of samples in the task.

- $\Sigma_1, \Sigma_2$  are the covariance matrices of the feature distributions of Task 1 and Task 2, respectively. Each covariance matrix  $\Sigma_T$  is calculated as:

$$\Sigma_T = \frac{1}{N-1} \sum_{i=1}^N (f(x_i) - \mu_T)(f(x_i) - \mu_T)^\top,$$

Lower FID values indicate greater similarity between the tasks, suggesting that their feature distributions are closely aligned.

### A.2 Cosine Similarity

Cosine similarity quantifies the orientation between two feature vectors by measuring the cosine of the angle between them. It is computed as:

$$\text{Cosine Similarity} = \frac{A \cdot B}{\|A\| \|B\|},$$

where:

- $A$ : Feature vector of the new task.
- $B$ : Feature vector of the initial task.

Higher Cosine Similarity values indicate greater alignment between the tasks, suggesting stronger task similarity.

### A.3 Attention Weight

The attention weight  $\alpha_j$  quantifies the relevance of the  $j^{\text{th}}$  past task relative to the current task  $T_t$  and is used to prioritize informative feature representations during replay-based learning. Following the scaled

dot-product attention mechanism (Vaswani et al. 2017), we normalize similarity scores between the current task and previously encountered tasks using their feature representation.

The attention weight  $\alpha_j$  is computed as:

$$\alpha_j = \frac{\exp\left(\frac{Q_t \cdot K_j^\top}{\sqrt{d_k}}\right)}{\sum_{l=1}^{t-1} \exp\left(\frac{Q_t \cdot K_l^\top}{\sqrt{d_k}}\right)} = \frac{\exp\left(\frac{\beta(T_t, i^*) \cdot \beta(T_j, i^*)^\top}{\sqrt{d_k}}\right)}{\sum_{l=1}^{t-1} \exp\left(\frac{\beta(T_t, i^*) \cdot \beta(T_l, i^*)^\top}{\sqrt{d_k}}\right)},$$

where:

- $\beta(T_t, i^*)$ : Feature vector (query  $Q_t$ ) for the current task.
- $\beta(T_j, i^*)$ : Feature vector (key  $K_j$ ) for the  $j^{\text{th}}$  past task.
- $d_k$ : Dimensionality of the feature vectors (for scaling).

This attention formulation allows the model to emphasize past experiences that are most relevant to the current task, enhancing prediction quality and knowledge retention.

#### A.4 AUROC

AUROC score, which measures the model’s ability to distinguish between classes by calculating the area under the ROC curve. In our work, AUROC is computed for each task in a sequence, and we compute the Average AUROC by averaging these scores across all tasks

#### A.5 Forgetting

Forgetting intuitively represents the degradation in a model’s performance on previously learned tasks after it is trained on a new task (Chaudhry, Dokania, Torr, and Vedaldi 2018). Suppose we have a sequence of  $T$  tasks. For a task  $T_i$ , let  $a_i^i$  denote its accuracy immediately after being learned, and  $a_t^i$  denote its accuracy after learning task  $T_t$  (for  $t > i$ ). Then, the forgetting for task  $T_i$  at time  $t$  is defined as:

$$F_t^i = \max(a^i) - a_t^i,$$

where  $\max(a^i)$  is the highest accuracy achieved on task  $T_i$  over time.

The overall forgetting at time  $t$  is then averaged over all previously learned tasks:

$$\text{Average Forgetting} = \frac{1}{t-1} \sum_{i=1}^{t-1} [\max(a^i) - a_t^i].$$

## REFERENCES

Aloui, A., J. Dong, C. P. Le, and V. Tarokh. 2023. “Transfer learning for individual treatment effect estimation”. In *Proceedings of the 39th Conference on Uncertainty in Artificial Intelligence (UAI)*, Volume 216 of *Proceedings of Machine Learning Research*, 56–66. Pittsburgh, Pennsylvania, USA: PMLR.

Chaudhry, A., P. K. Dokania, P. H. S. Torr, and A. Vedaldi. 2018. “Riemannian walk for incremental learning: Understanding forgetting and intransigence”. In *Computer Vision – ECCV 2018: 15th European Conference*, 532–547. Munich, Germany: Springer.

Heusel, M., H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. 2017. “GANs trained by a two time-scale update rule converge to a local Nash equilibrium”. In *Advances in Neural Information Processing Systems 30 (NeurIPS)*, 6626–6637. Long Beach, California, USA: Curran Associates, Inc.

Hiratani, N. 2024. “Disentangling and mitigating the impact of task similarity for continual learning”. *arXiv preprint arXiv:2405.20236*.

Kirkpatrick, J., R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, and A. A. e. a. Rusu. 2017. “Overcoming catastrophic forgetting in neural networks”. *Proceedings of the National Academy of Sciences* 114(13):3521–3526.

Kötter, A., S. Allenspach, C. Grebner, H. Matter, J. A. Hiss, G. Schneider *et al.* 2024. “Task-similarity is a crucial factor for few-shot meta-learning of structure-activity relationships”. *ChemBioChem* 25(19):e202400095.

Le, C. P., J. Dong, M. Soltani, and V. Tarokh. 2022. “Task affinity with maximum bipartite matching in few-shot learning”. In *International Conference on Learning Representations (ICLR)*. Virtual Conference.

Le, C. P., M. Soltani, J. Dong, and V. Tarokh. 2022. “Fisher task distance and its application in neural architecture search”. *IEEE Access* 10:47235–47249.

Lee, S., S. Goldt, and A. Saxe. 2021. “Continual learning in the teacher-student setup: Impact of task similarity”. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, Volume 139 of *Proceedings of Machine Learning Research*, 6109–6119. Virtual Conference: PMLR.

Parisi, G. I., R. Kemker, J. L. Part, C. Kanan, and S. Wermter. 2021. “A survey of continual learning with deep networks: Theory, method, and application”. *arXiv preprint arXiv:1802.07569*.

Shin, H., J. K. Lee, J. Kim, and J. Kim. 2017. “Continual learning with deep generative replay”. In *Advances in Neural Information Processing Systems 30 (NeurIPS)*, 2990–2999. Long Beach, California, USA: Curran Associates, Inc.

Shui, C., M. Abbasi, L.- Robitaille, B. Wang, and C. Gagné. 2019. “A principled approach for learning task similarity in multitask learning”. *arXiv preprint arXiv:1903.09109*.

Upadhyay, R., R. Phlypo, and M. Liwicki. 2024. “Sharing to learn and learning to share: A meta-review on meta, multi-task, and transfer learning”. *IEEE Access* 12:13456–13478.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, *et al.* 2017. “Attention is all you need”. In *Advances in Neural Information Processing Systems 30 (NeurIPS)*, 5998–6008. Long Beach, California, USA: Curran Associates, Inc.

Wang, E. 2024. “Continual learning with applications in different scenarios”. *University of Texas at Dallas Repository*.

Yoon, J., E. Yang, J. Lee, and S. J. Hwang. 2018. “Lifelong learning with dynamically expandable networks”. In *International Conference on Learning Representations (ICLR)*. Vancouver, BC, Canada.

Zhang, Y., and Q. Yang. 2022. “A survey on multi-task learning”. *IEEE Transactions on Knowledge and Data Engineering* 34(12):5586–5609.

## AUTHOR BIOGRAPHIES

**JIANYU WANG** is a Ph.D. student at the Department of Computational and Data Sciences at the George Mason University. He earned his Master’s degree in Information System from the Johns Hopkins University. His research interests include machine learning and artificial intelligence. His email address is [jwang72@gmu.edu](mailto:jwang72@gmu.edu).

**JACOB NEAN-HUA SHEIKH** is pursuing a dual B.S./M.S. degree in the Department of Computer Science at George Mason University, with a focus on computational neuroscience. His research interests include biologically inspired machine learning, continual learning, and applied medical informatics. His email is [jsheikh2@gmu.edu](mailto:jsheikh2@gmu.edu).

**CAT P. LE** received a B.S. degree in electrical and computer engineering from Rutgers University, an M.S. degree in electrical engineering from the California Institute of Technology (Caltech), and a Ph.D. degree in electrical and computer engineering from Duke University. His research interests include image processing, computer vision, and machine learning, with a focus on transfer learning, and continual learning. His email address is [cat.le@duke.edu](mailto:cat.le@duke.edu). His website is <https://www.catphuocle.com/>

**HODA BIDKHORI** is an assistant professor at the Department of Computational and Data Sciences at the George Mason University. She earned her Ph.D. in Applied Mathematics from the Massachusetts Institute of Technology (MIT), where she subsequently spent several years as a postdoctoral researcher and lecturer in Operations Research and Statistics. Her current research focuses on the theory and applications of data science. Her e-mail address is [hbidkhori@gmu.edu](mailto:hbidkhori@gmu.edu). Her website is <https://sites.google.com/view/hoda-bidkhori/home>.