Adversarially Diversified Rehearsal Memory (ADRM): Mitigating Memory Overfitting Challenge in Continual Learning

Hikmat Khan
Rowan University
Glassboro, New Jersey, USA
khanhi83@rowan.edu

Ghulam Rasool
Moffitt Cancer Center
Tampa, Florida, USA
ghulam.rasool@moffitt.org

Nidhal Carla Bouaynaya
Rowan University
Glassboro, New Jersey, USA
bouaynaya@rowan.edu

***Abstract*—Continual learning (CL) focuses on learning non-stationary data distribution without forgetting previous knowledge. Rehearsal-based approaches are commonly used to combat catastrophic forgetting. However, these approaches suffer from a problem called "rehearsal memory overfitting", where the model becomes too specialized on limited memory samples and loses its ability to generalize effectively. As a result, the effectiveness of the rehearsal memory progressively decays, ultimately resulting in catastrophically forgetting the learned tasks.**

To address the memory overfitting challenge, we introduce the *Adversarially Diversified Rehearsal Memory*, or ADRM, a novel method designed to enrich memory sample diversity and bolster resistance against natural and adversarial noise disruptions. ADRM employs the Fast Gradient Sign Method (FGSM) to introduce adversarially modified memory samples, achieving two primary objectives: enhancing memory diversity and fostering a robust response to continual feature drifts in memory samples.

We conducted extensive experiments on the CIFAR10 dataset and found that ADRM outperforms several existing CL approaches and performs comparable to state-of-the-art methods. Additionally, we demonstrated ADRM's ability to enhance CL model robustness under natural and adversarial conditions by using CIFAR10-C and adversarially perturbed CIFAR10 datasets.

Our contributions are as follows: Firstly, ADRM addresses overfitting in rehearsal memory by employing FGSM to diversify and increase the complexity of the memory buffer. Secondly, we demonstrate that ADRM mitigates memory overfitting and significantly improves the robustness of CL models, which is crucial for safety-critical applications. Finally, our detailed analysis of features and visualization demonstrates that ADRM mitigates feature drifts in CL memory samples, significantly reducing catastrophic forgetting and resulting in a more resilient CL model. Additionally, our in-depth t-SNE visualizations of feature distribution and the quantification of the feature similarity further enrich our understanding of feature representation in existing CL approaches. Our code is publically available at https://github.com/hikmatkhan/ADRM.

I. Introduction

Continual Learning (CL), also called incremental, lifelong, or sequential learning, equips deep learning models with the ability to accumulate and expand knowledge over time, similar to humans [1]. Despite improvements in CL methodologies, current approaches still suffer from a phenomenon known as *catastrophic forgetting* or *catastrophic interference* [2], [3], where models forget previously acquired knowledge after acquiring new knowledge or skills. Several approaches have been proposed to mitigate catastrophic forgetting in CL, broadly categorized into four types: 1). Regularization-based approaches introduce the penalties on alterations to crucial parameters of the model to prevent previously acquired knowledge [1], [2]. 2). Dynamic architecture-based approaches employ sub-networks or expandable architectures designed to adapt to new tasks while minimally impacting previously acquired knowledge [1], [2]. 3). Rehearsal-based approaches involve storing a subset of previous data in rehearsal memory and periodically revisiting examples from past tasks to assist the model in retaining older knowledge [4], [5]. 4). Hybrid approaches combine and aim to leverage the strengths of different learning strategies to propose the CL model that suffers less from catastrophic forgetting [1], [6], [7]. Rehearsal-based approaches are becoming more popular in CL approaches. These approaches use a rehearsal memory to store and rehearse the limited data samples from previously learned tasks. However, these approaches have some limitations. The efficacy of the stored samples decreases or diminishes over time due to the limited budget of memory and lack of diversity in the stored samples [1], [2], [8], which potentially leads to overfitting of the stored samples [1], [2]. To tackle this challenge, the authors in [9] proposed a continuous and reversible memory transformation function that ensures memory data is resistant to overfitting and injected the uncertainty in the transformation function, thereby enhancing the diversity of the transformed memory buffer and improving generalization. The authors in [10] proposed an approach, called ConPL, which incorporated the previous class's prototype-based memory enhancement, resulting in less memory sample overfit and improved performance. The authors of [11] dynamically reshape old class feature distributions by incorporating previous class prototypes with arriving new class features, preserving prior task decision boundaries. The authors in [12] dynamically evolve the distribution of memory data to improve robustness and prevent memory overfitting. Similarly, the authors in [13] proposed an evolved mixture model that adapts to data distribution shifts of the prior tasks. The authors in [14] adopted a distinct strategy

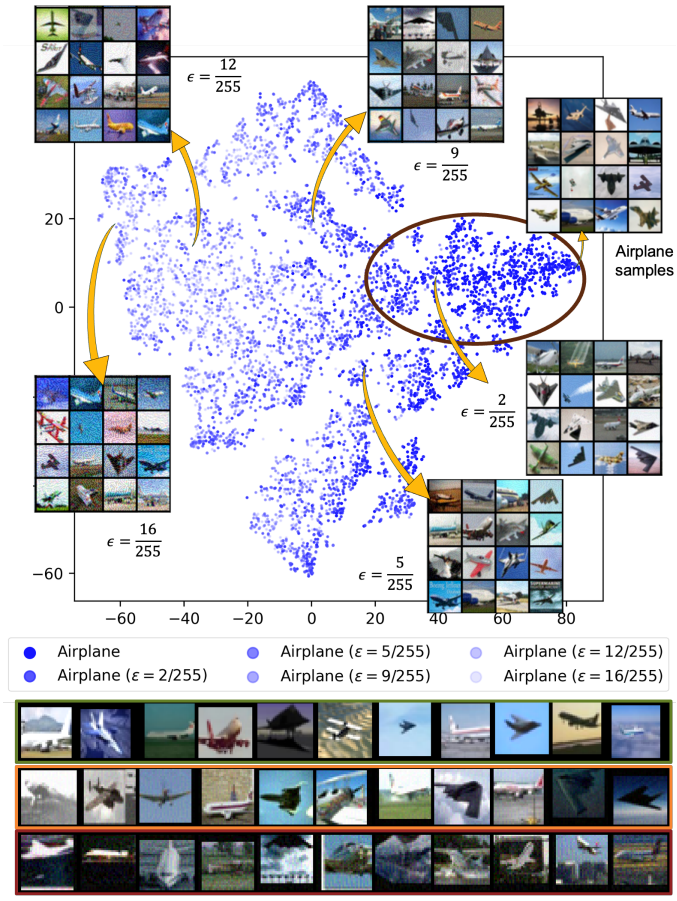

Fig. 1. Idea illustration: t-SNE visualization showing the dispersion of 'airplane' class features under increasing adversarial perturbations (scaled as $\frac{\epsilon}{255}$). Dark blue dots represent the original airplane class samples, while progressively lighter shades of blue indicate the five levels of adversarially diversified samples. The visualization demonstrates how increasing the strength of the perturbation results in a divergence from the original class cluster. These adversarially diversified samples retain inherent airplane characteristics and can considered as potential rehearsal samples to enhance the diversity and complexity of the memory to prevent rehearsal memory overfitting. 1st row, the samples enclosed in a green rectangle represent the original, undiversified baseline memory. 2nd row, the samples enclosed in an orange rectangle are adversarially diversified airplanes correctly classified by the CL model despite diversification. 3rd row, the samples enclosed in a red rectangle are adversarially diversified airplanes incorrectly classified, highlighting potential areas for enhancing model robustness. The diversity in rehearsal samples is crucial to prevent memory overfitting and increase the complexity of memory samples, consequently improving the robustness of the CL model (best viewed in color).

that utilized gradient-based editing of memory samples. This approach aimed to reduce memory overfitting and forgetting while incorporating updated information. The authors in [15] proposed a strategy called Bang's Rainbow Memory (RM) to diversify memory samples and avoid memory overfitting. This was achieved by leveraging per-sample classification uncertainty and data augmentation. Similarly, the author in [16] introduced diversifying exemplars using a learnable feature generator and semantic contrastive learning.

In this paper, we adopt a relatively straightforward approach, termed Adversarially Diversified Rehearsal Memory (ADRM), which aims to diversify and enhance the complexity of the rehearsal memory sample, which is achieved by employing the one-step Fast Gradient Sign Method (FGSM) attack on the memory samples. We then rehearse a mixture of both successful (misclassified) and unsuccessful (correctly classified) adversarially perturbed/diversified memory samples with the current task mini-batch (see Figure 1). Even after adversarial perturbation, correctly classified sample options are crucial in diversifying the memory samples. Meanwhile, the misclassified samples reveal features more closely associated with the feature distributions of other memory classes or current task classes. This provides a unique opportunity for the CL model to understand the decision boundaries among different tasks. Figure 1 illustrates the adversarially diversified memory samples for the airplane class, showing that as the intensity of the FGSM attack increases, the resultant samples progressively deviate from the original airplane feature distribution (or core cluster). The ADRM uses FGSM to generate such diverse memory samples to effectively tackle the feature distributional drift of memory classes, as depicted in Figure 7. We conducted extensive experiments on the CIFAR10 dataset and demonstrated that ADRM's performance outperforms several well-established CL approaches and is comparable to the state-of-the-art CL approaches. Additionally, we conducted comprehensive evaluations of ADRM and state-of-the-art baseline methods to demonstrate its stable feature distribution and robustness in natural and adversarial environments, using the CIFAR10-C [17] and an adversarially perturbed CIFAR10 dataset [4]. Lastly, we analyzed feature distribution and similarity in existing CL approaches in detail using t-SNE visualizations and Central Kernel Alignment (CKA), respectively. This analysis helped us to gain a better understanding of the learned feature representation in CL approaches. Our findings revealed that the learned feature representations in existing CL approaches were more similar to ADRM on the standard CIFAR10 dataset. However, when tested under natural and adversarial conditions, all existing CL approaches demonstrated degraded performance. Interestingly, CL approaches that suffered less forgetting showed higher similarity in learned feature representation to our ADRM, indicating the existence of continually robust features.

Our contributions are fourfold:

- To address overfitting in rehearsal memory, we proposed ADRM, utilizing the FGSM attack to diversify and enhance the complexity of the memory samples, thereby preventing memory overfit. This approach is based on the premise that robust representations aid generalization to out-of-distribution data [18].
- We conducted extensive evaluations of ADRM and state-of-the-art methods under natural and adversarial noise conditions, demonstrating that ADRM prevents memory overfitting and enhances the robustness of the CL model, which is crucial for safety-critical scenarios.
- Through t-SNE visualization [19] of learned feature distributions, we observed that adversarial diversification of

the memory helps mitigate feature distribution drifts in memory samples compared to existing CL approaches, resulting in reduced catastrophic forgetting and a more robust CL model.
- Our findings indicate that existing CL approaches demonstrate higher feature similarity scores (using central kernel alignment (CKA) [20]) in learned features representation to the learned features of ADRM on standard CIFAR10. In contrast, they showed lesser similarity scores to the learned features of ADRM under natural and adversarial noisy conditions. These observations highlight that ADRM learned continually-adversarially robust features to mitigate catastrophic forgetting, unlike existing CL approaches, which suffered from catastrophic forgetting in natural noise and adversarial conditions.

This section covers the motivation behind the adversarial diversification of memory used in ADRM, which is explained in Section I-A. The ADRM is based on the conventional rehearsal, a widely used strategy in Continual Learning, which is elaborated in Section I-B. We then present our approach in detail, which involves two steps: creating the adversarially diversified memory through FGSM perturbations, discussed in Section I-C, and rehearsing the diversified memory samples using a strategy detailed in Sections I-B, I-D and I-E.

A. Motivation

ADRM prevents memory overfitting in the CL model by diversifying and enhancing the complexity of memory samples using an adversarial FGSM approach [21]. It introduces varying strengths of adversarial perturbations to each class memory sample to enrich diversification within the class (see Figure 1) using one-step FGSM adversarial. The resultant diversified memory samples are more complex and challenging to overfit by the CL model. Each step of adversarial diversification results in a unique diversified sample [21]. Additionally, the adversarial diversification of the rehearsal memory reinforces the CL model to minimize feature distribution drift within each memory class, better learn intra-class boundaries, resulting in generalization and robustness, and reduce catastrophic forgetting.

B. Conventional Memory Replay

The standard rehearsal approach, also known as experience replay [22], involves combining a limited number of memory samples from previous tasks with mini-batches of the current task dataset. This approach minimizes data risk for the rehearsal memory ($\mathcal{M}$) and the current task data ($\mathcal{D}_t$). Formally, the optimization process can be expressed as follows:

$$\min_{\theta \in \Theta} \left[\mathbb{E}_{(x_t, y_t) \sim \mathcal{D}_t} \mathcal{L}(\theta, x_t, y_t) + \mathbb{E}_{(x_m, y_m) \sim \mathcal{M}_t} \mathcal{L}(\theta, x_m, y_m) \right], \tag{1}$$

where t represents the t^{th} task or step, θ denotes the parameters of the CL model, and $\mathcal{L}$ represents the loss function, typically cross-entropy loss, commonly used, $\mathcal{D}_t$ represents the current task data, while $\mathcal{M}_t$ represents the rehearsal memory data for previous tasks. The pairs (x_t, y_t) and (x_m, y_m) are samples, drawn from $\mathcal{D}_t$ and $\mathcal{M}_t$, respectively. Given the memory constraints, rehearsal memory can only store a limited subset of old data. This limitation increases the model's tendency to overfit these memory samples [2], [23]. As a result, the effectiveness of these samples in mitigating forgetting is diminished [2], [23]. Furthermore, this memory overfitting impairs the model's generalization capabilities and intensifies the catastrophic forgetting of previously learned tasks.

Our proposed ADRM addresses the rehearsal memory overfitting by enhancing the diversity and complexity of the rehearsal memory samples in an adversarial way, thereby preventing the model from overfitting the memory samples. The details of this approach are explained in further detail in the next sections (I-C, and I-D).

C. Adversarial Diversification of Rehearsal Memory Using FGSM

Formally, the ADRM employs the one-step FGSM on the rehearsal memory ($\mathcal{M}$) for performing memory sample diversification [21], [24], is represented as follows.

$$x_{\text{diversified}} = x_m + \epsilon \cdot \text{sign}(\nabla_{x_m} J(\theta, x_m, y_m)), \tag{2}$$

where $x_{\text{diversified}}$ represents the adversarially diversified samples of the corresponding x_m memory sample, and epsilon (ϵ) is the perturbation strength, the sign function is used to determine the direction of the perturbation, while $\nabla_{x_m} J(\theta, x_m, y_m)$ signifies the gradient of the loss function with respect to the input x_m. Here, θ denotes the parameters of the CL model, and y_m is the true label for x_m. The pairs (x_m, y_m) belongs to the rehearsal memory ($\mathcal{M}$). The $x_{\text{diversified}}$ represents the adversarially diversified samples (with increased complexity) of the x_m original memory samples. These samples are interleaved with mini-batches sampled from the current task data ($\mathcal{D}_t$). Notably, the adversarially diversified memory samples created at each CL step are unique and can be generated in any desired number for each previously learned class (i.e., memory samples) through a single FGSM step [21], [24]

D. Creating Diversified Rehearsal Memory

We randomly select a mini-batch of data from the rehearsal memory ($\mathcal{M}$) and employ the FGSM to generate adversarially perturbed/diversified memory samples. The diversified samples are divided into subsets $\mathcal{M}_d$ and $\mathcal{M}_b$; the former subset contains those diversified memory samples where the adversarial perturbation led to misclassification, while the latter subset contains diversified memory samples that were still classified correctly by the CL model despite the perturbation. Together, these subsets of diversified memory samples (i.e., $\mathcal{M}_d$) and $\mathcal{M}_b$) can be referred to as a diversified version of rehearsal memory (i.e., $\mathcal{M}$). The $\mathcal{M}_d$ subset increases diversity and reduces overfitting, while the $\mathcal{M}_b$ subset helps maintain decision boundaries between classes and ensures robust continual learning with stable feature distributions, resulting in less catastrophic forgetting (see Figure 1).

E. Memory Diversification Ratios in ADRM

In order to study the impact of memory diversification on the CL model, we incorporated adversarially diversified rehearsal memory in five different ratios: 10%, 25%, 50%, 75%, and 100%. For example, an ADRM model that rehearses 10% of both $\mathcal{M}_d$ and $\mathcal{M}_b$ along with the original memory samples, would be referred to as ADRM (0.1). Similarly, ADRM models trained with 25%, 50%, 75%, and 100% are denoted as ADRM (0.25), ADRM (0.5), ADRM (0.75), and ADRM (1), respectively.

II. Experimental setup

A. Datasets

We utilized the widely-used split CIFAR10 benchmark dataset [2], [25]–[28]. CIFAR10 is known for its complexity and consists of ten distinct classes, each containing $6,000$ images. Among these images, $5,000$ were used for training, while the remaining $1,000$ were reserved for the test set [25]. We also evaluated the robustness of CL models against natural noise using the CIFAR10-C dataset [29]. Additionally, we employed the adversarially perturbed CIFAR10 dataset to evaluate the robustness of CL models against adversarial noise [4]. Figure 4 presents sample images from the adversarially perturbed CIFAR10 dataset [4].

B. Protocols

We utilized the challenging class-incremental learning (CIL) protocol, which simulates real-life situations where the model learns sequentially from streaming tasks without prior task identification [1], [2], [30], [31]. According to this protocol, the CIFAR-10 dataset is divided into nine, five, and two tasks, each with a fixed memory size of 1024 examples, following the CIL protocol [30]–[32].

C. Baselines

We conducted a comprehensive comparative analysis of our proposed method with the existing well-established and state-of-the-art CL methods, which included: 1) Experience Replay (ER), also referred to as Standard Rehearsal [39]. ER employs reservoir sampling to store a compact subset of data from previous tasks. During learning of the new tasks, a random subset of examples from the memory buffer is sampled and combined with the incoming mini-batch data to mitigate forgetting; 2). Incremental Classifier and Representation Learning (iCaRL) incrementally train a model on new classes while retaining knowledge of previous ones through representation learning and an exemplar memory mechanism [33]; 3). Bias Correction (BiC) that adjusts a model's bias to balance performance between newly added and original classes [27]; 4). Pooling-based Online Distillation Network (PODNet) addresses catastrophic forgetting by utilizing spatial-based distillation and feature pooling [34]; 5). Weight Alignment (WA) uses knowledge distillation to maintain the discrimination within old classes adjust, followed by correcting the biased weights in the fully connected layer of the model [26]; 6). Dynamically Expandable Representation (DER) learns and dynamically expands the features representation with a new extractor, which is followed by re-training the classifier using currently available data to reduce bias in the classifier weight caused by imbalanced training [35]; 7). SimpleCIL uses pre-trained model embeddings and sets classifier weights directly to prototype features of each class, avoiding the need for additional downstream task training [28]; 8). Feature BoOSTing and CompreEssion for class-incRemental learning (FOSTER) introduces a two-stage method that expands and then compresses a model for class-incremental learning [36], which effectively balances new-category learning with preserving old knowledge; 9). FeTrIL, Similar to FOSTER, FeTrIL introduces a two-stage method for class-incremental learning, encompassing both model expansion and compression stages [37], which effectively balances new category learning with the preservation of prior knowledge; 10). Memory-efficient Expandable MOdel (MEMO) optimizes memory efficiency by expanding specialized layers for new tasks while sharing generalized layers, ensuring an efficient and effective approach to continual learning [38]. We also included a fine-tuned model that learned all tasks in sequence without specific strategies to prevent forgetting. We also used a joint model trained on the entire dataset, which included all available classes. The fine-tuned model established the lower performance, whereas the joint model established the upper bound performance.

TABLE I

Comparative performance of CL methods on the Split-CIFAR10 dataset over 9 steps, 5 steps, and 2 steps. The proposed ADRM method, rehearsing with 10% adversarially diversified memory, demonstrates superior performance over variants incorporating 25%, 50%, 75%, and 100% diversified memory. The ADRM outperforms several established CL methods and achieves comparable results to state-of-the-art CL approaches. Each experiment was repeated five times with different random seeds to ensure the robustness of our findings.

CL Methods	**Split-CIFAR10**		
	9 steps	**5 steps**	**2 steps**
Joint		94.8	
Fine-tune	11.11	18.54	45.31
Experience Replay [5]	67.62	75.97	80.26
iCaRL [33]	69.25	74.85	79.98
BiC [27]	53.11	71.01	86.57
PODNet [34]	72.41	76.96	87.64
WA [26]	75.56	81.67	85.34
DER [35]	74.33	78.14	82.57
SimpleCIL [28]	52.67	54.54	75.01
FOSTER [36]	74.61	80.40	83.89
FETRIL [37]	65.26	67.51	84.52
MEMO [38]	80.60	85.93	87.81
ADRM (0.1)	74.76	80.59	83.95
ADRM (0.25)	72.39	79.15	85.39
ADRM (0.5)	69.20	76.14	81.41
ADRM (0.75)	68.59	76.36	80.17
ADRM (1)	65.39	76.21	82.86

D. Training details

We used the PyTorch framework [40] and PyCIL codebase [41] to implement our experiments. We employed the standard ResNet32 architecture with Xavier initialization [42], [43] and

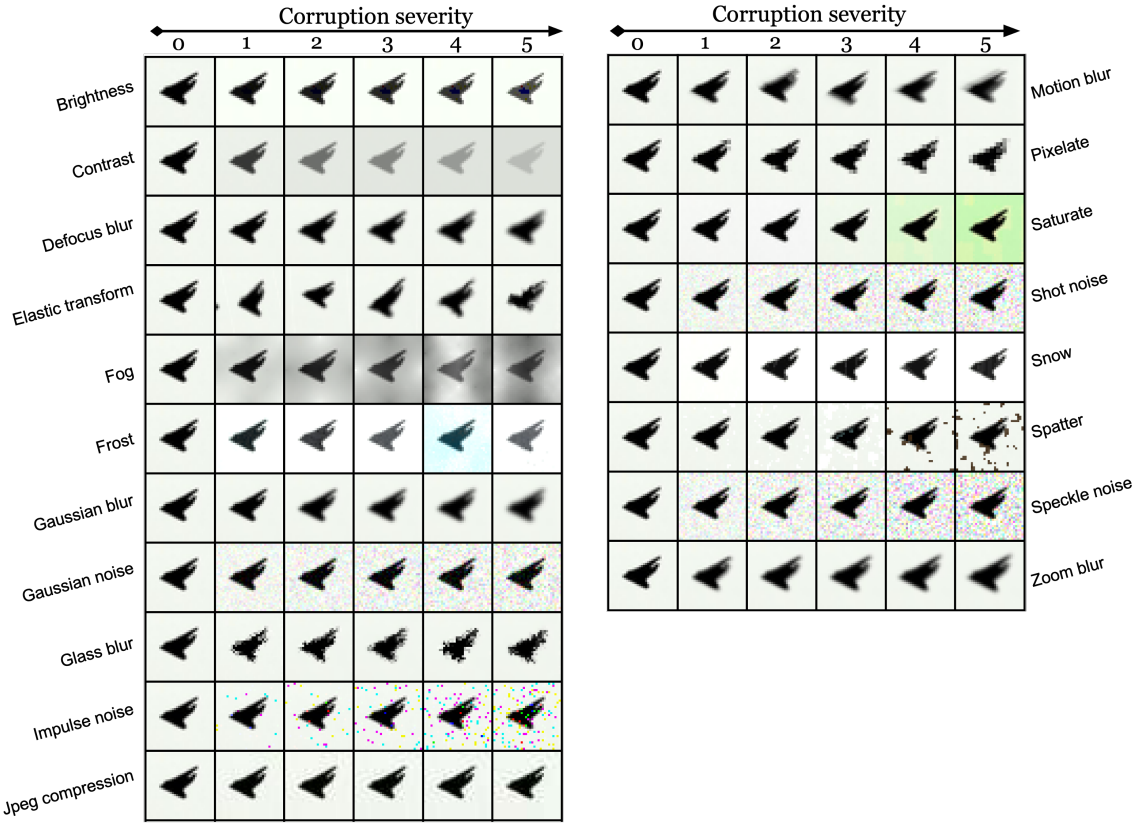

Fig. 2. Examples of the six different severity levels for the 19 different noise types. Severity 0 represents the original clean image whereas severity 5 corresponds to the most severe corruption (Best viewed in color).

a batch size of 256. We used a stochastic gradient descent optimizer to train the models with an initial learning rate of 0.01, a momentum of 0.9, and a learning rate decay of 0.1. We used the standard cross-entropy loss function for classification loss. The training epochs were set to 200 for the initial task and 128 for subsequent tasks. We also used standard data augmentation techniques, such as random flip, cropping, and adjustments in brightness and contrast [44]. We set the hyper-parameters specific to each baseline model to their default configurations as specified in their respective papers [26], [27], [33]–[36] and implemented in PyCIL [41] to ensure optimal performance for each baseline for fair comparison. To diversify memory samples, we utilized a perturbation parameter (epsilon ϵ) in the FGSM attack. We randomly chose epsilon (ϵ) from a uniform distribution between $\frac{1}{255}$ and $\frac{16}{255}$. This range was carefully selected to create adversarial perturbations intense enough to diversify memory samples while preserving the visual characteristics of the original memory class.

E. Evaluation

We evaluate the performance of the CL models using the widely adopted metric of average classification accuracy (ACA) [1], [32], [33], [36], [37], [45], which is calculated by assessing the final trained model on all the learned tasks. Formally, ACA is defined as:

$$ACA = \frac{1}{T}\sum_{i=1}^{T} R_{T,i}, \tag{3}$$

where R represents accuracy, T is the total number of tasks, and i is the task index.

III. Results and Discussion

Table I presents a comparative performance analysis of the ADRM and its variants methods with baseline CL approaches for tasks involving nine, five, and two task configurations on the CIFAR10 dataset. Notably, the ADRM variant with a 10% inclusion ratio of diversified memory samples, referred to as ADRM (0.1), performed the best with the highest average accuracy among all the other variants of ADRM. Moreover, the ADRM (0.1) variant demonstrated superior performance compared to several well-established methods, such as Experience Replay, iCaRL, BiC, PODNet, SimpleCIL, and FetRIL, especially in situations that involve nine tasks. Additionally, ADRM obtained comparable average accuracy to methods like WA, DER, and FOSTER.

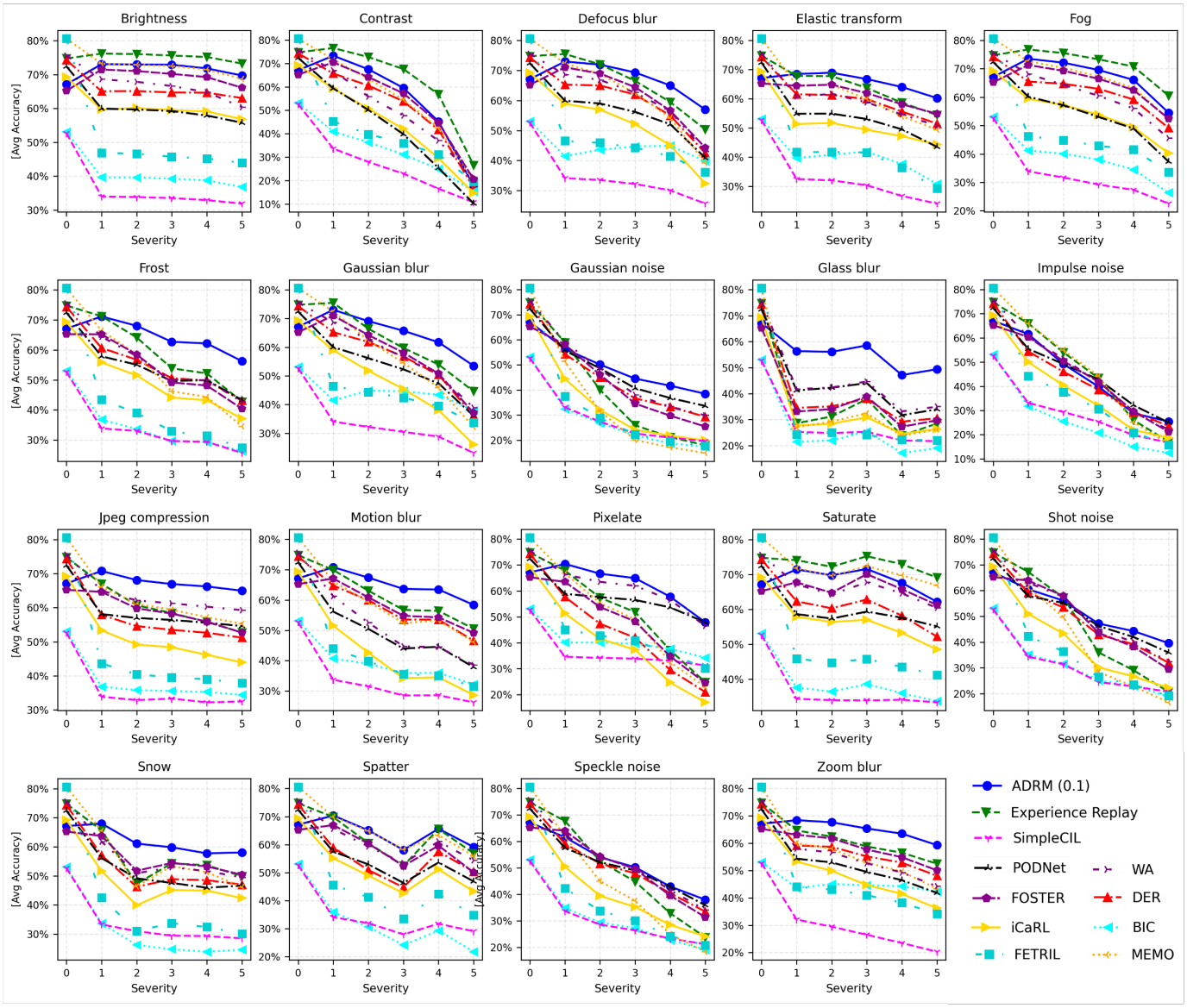

Fig. 3. Robustness comparison of eleven CL approaches against nineteen types of natural image corruptions. Each subplot corresponds to a different noise type, with the x-axis representing the severity level from 0 (no corruption) to 5 (maximum corruption) and the y-axis displaying the average accuracy of each model. The ADRM shows resilience and comparatively suffers less from catastrophic forgetting. It outperformed other models in 15 out of the 19 tested noise conditions, highlighted with a yellow rectangle in the graph (best viewed in color).

A. Balancing Generalization and Robustness: The Impact of Adversarially Diversified Memory Samples in ADRM Model Training

Recall that the different variants of ADRM rehearse various proportions/ratios of adversarially diversified memory samples, such as 10%, 25%, 50%, 75%, and 100% of rehearsal memories (which were represented by $\mathcal{M}_d$ and $\mathcal{M}_b$ in Section I-E). We conducted extensive experiments and found a tradeoff between the generalization and robustness of ADRM models. Specifically, as we increased the proportion of diversified memory samples, the model's robustness improved, but its ability to generalize to new task data was adversely affected. Surprisingly, we observed that the optimal balance was achieved using a 10% inclusion ratio of diversified memory samples. However, when gradually increasing the ratio of diversified memory samples from 10% to 100%, we noticed a decline in performance, especially in the 9-task scenario (see Table I). Furthermore, our findings suggest that rehearsing with a 10% inclusion ratio of diversified memory samples enhances generalization and robustness and prevents memory overfitting, resulting in less catastrophic forgetting in the CL model (discussed in the following Section III-B).

B. Diversifying Rehearsal Memory Leads to Robust Continual Learning Models

The ADRB achieved lower average accuracy in comparison to the highest-performing MEMO approach (refer to Table

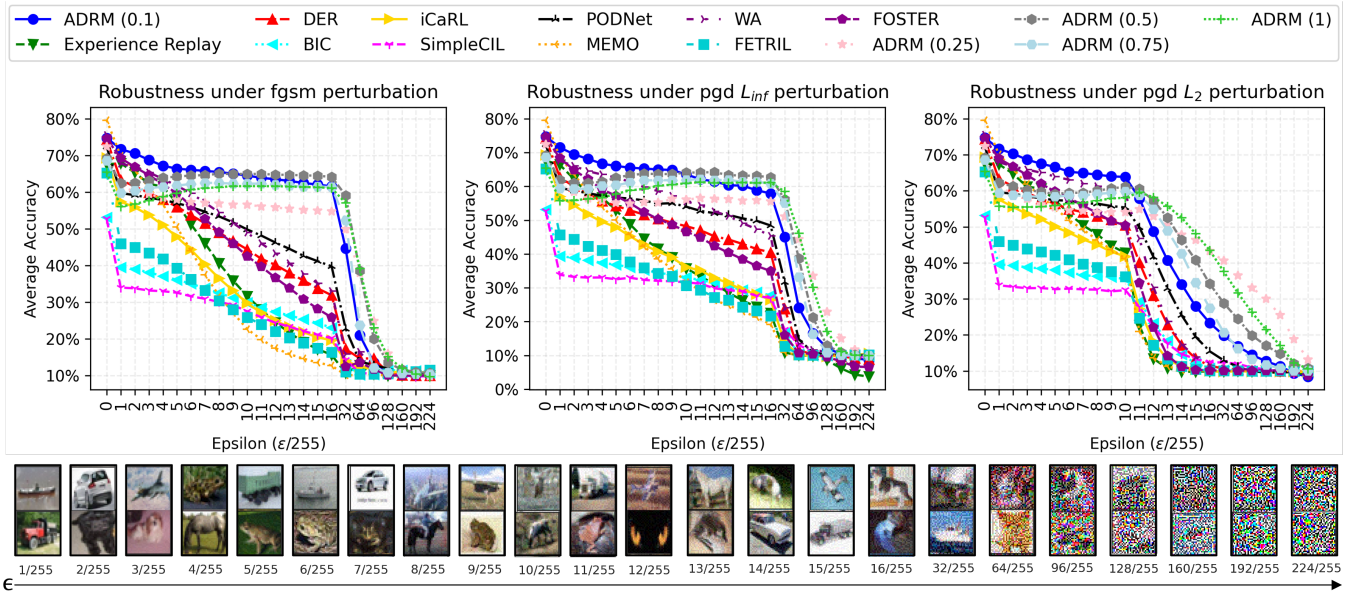

Fig. 4. Robustness comparison of fifteen CL approaches on the adversarially perturbed CIFAR-10 validation set [4]. The x-axis represents the epsilon value ($\epsilon/255$), indicating the strength of the adversarial perturbation, while the y-axis shows the models' average accuracy. The sample images at the bottom are adversarially perturbed using the Fast Gradient Sign Method (FGSM) with increasing perturbation strength (ϵ) from left to right. ADRM and its variants demonstrate enhanced adversarial robustness over a range of attack intensities relative to other CL approaches, indicating a stronger grasp of the fundamental class concepts within the dataset (best viewed in color).

I). However, it demonstrated stability and experienced less catastrophic forgetting under natural and adversarial conditions as compared to the CL baselines.

1) Evaluating robustness against common corruptions: Figure 3 shows the performance of different CL methods against nineteen different types of noise (see Figure 2 for samples images from CIFAR10-C dataset). The ADRM model was the most resilient and least prone to forgetting, maintaining the highest average accuracy in 15 out of 19 noise types. However, it is worth noting that the MEMO model performed the best overall. Nevertheless, it struggled with natural noise and showed a higher tendency to catastrophic forgetting than the ADRM model and other established CL methods.

2) Evaluating robustness against adversarial corruptions: Figure 4 illustrates that ADRM variants consistently outperformed the CL baseline methods in performance across three different versions of the dataset, each altered using different adversarial attack methods: PGD-L_{inf}, PGD-L_2, and FGSM (see Figures 6 and 5 for samples images from adversarially perturbed CIFAR10 datasets). Notably, the highest-performing CL approach, MEMO, exhibited worse performance and the highest forgetting compared to other CL approaches. Interestingly, the ADRM variant with a 10% diversity ratio attained the best results among its peers while maintaining accuracy similar to well-established CL approaches. These observations highlight the effectiveness of adding even a small amount of adversarial diversification (i.e., 10%) in bolstering the resilience of CL models and resulting in less forgetting under adversarial conditions.

C. Diversifying Rehearsal Memory Leads to Stable Feature Distribution

We analyzed the feature distributions learned by various CL approaches to gain insights into the catastrophic forgetting of previously learned classes using the adversarially perturbed CIFAR10 dataset (see Figure 7) [4]. We compared the feature distributions of four CL approaches, ADRM, Experience Replay, PODNet, and MEMO, across nine tasks for 'airplane' and 'automobile' classes using pre-softmax logits of the ResNet32 architecture. Our findings indicate that ADRM models learned stable feature distributions with minimal drifts and class overlap, while the other approaches had noticeable drifts and increasing overlap. Overall, Figure 7 shows that ADRM is more effective in learning core concepts and intra-class boundaries, reducing class forgetting even in challenging adversarial conditions.

D. Analyzing Feature Similarity in Continual Learning Models Using Central Kernel Alignment (CKA)

Figure 8 illustrates the similarity matrices of the learned feature representations for seven CL approaches, computed on the validation sets of the CIFAR10 and adversarially perturbed CIFAR10 datasets [4], computed using CKA [20]. We observed high similarities in the learned feature representations among ADRM and various CL approaches computed on the standard CIFAR10 dataset. In contrast, We observed lower similarities in the learned feature representations among the ADRM and various CL approaches computed on the adversarially perturbed CIFAR10 dataset. In addition, models with higher feature similarity scores to ADRM, such as DER

Fig. 5. Adversarial examples created using PGD-$L_{\inf}$. On the x-axis, we can see the different adversarially perturbed images from the dataset. The different strengths of the level of the attacks (ϵ) are plotted in the y-axis. The level of the attack (ϵ) determines the strength of the adversarial attack; the higher the value, the stronger the adversarial attack and the more perturbed the input image (Best viewed in color).

and PODNet, demonstrated better resistance to adversarial conditions and reduced catastrophic forgetting. Conversely, models with low feature similarity scores to ADRM's features, such as iCaRL and MEMO, showed increased vulnerability to adversarial conditions and greater forgetting of previously learned classes, indicating limitations in their learned representations when faced with adversarial conditions.

E. Learned Feature Disentanglement

We used a feature disentanglement framework from [46] to understand the feature representations learned by CL models. ADRM learned salient features of the automobile, such as tires and the frame, highlighting its focus on the class's key attributes. Other models, such as Experience Replay, PODNet, and MEMO, exhibited a more scattered and less class-specific feature pattern, making them vulnerable to noise and causing a higher forgetting of previously learned classes.

IV. CONCLUSION

In this paper, we introduce the Adversarially Diversified Rehearsal Memory (ADRM) approach to address the challenges of rehearsal memory overfitting and catastrophic forgetting challenges in continual learning. ADRM employs the Fast Gradient Sign Method (FGSM) to introduce memory diversification, aiming to accomplish two primary goals. Firstly, it enhances the rehearsal memory's diversity to mitigate memory overfitting. Our extensive experiments on the CIFAR10 dataset demonstrate that ADRM surpasses several CL approaches and achieves comparable performance to state-of-the-art CL approaches. Secondly, ADRM enhances the robustness of the CL model, leading to reduced forgetting of previously learned classes under both natural and adversarial conditions. Specifically, ADRM maintained the highest average accuracy in 15 out of 19 noise types (e.g., Gaussian noise and impulse noise), as evaluated on the CIFAR10-C dataset. Moreover, ADRM

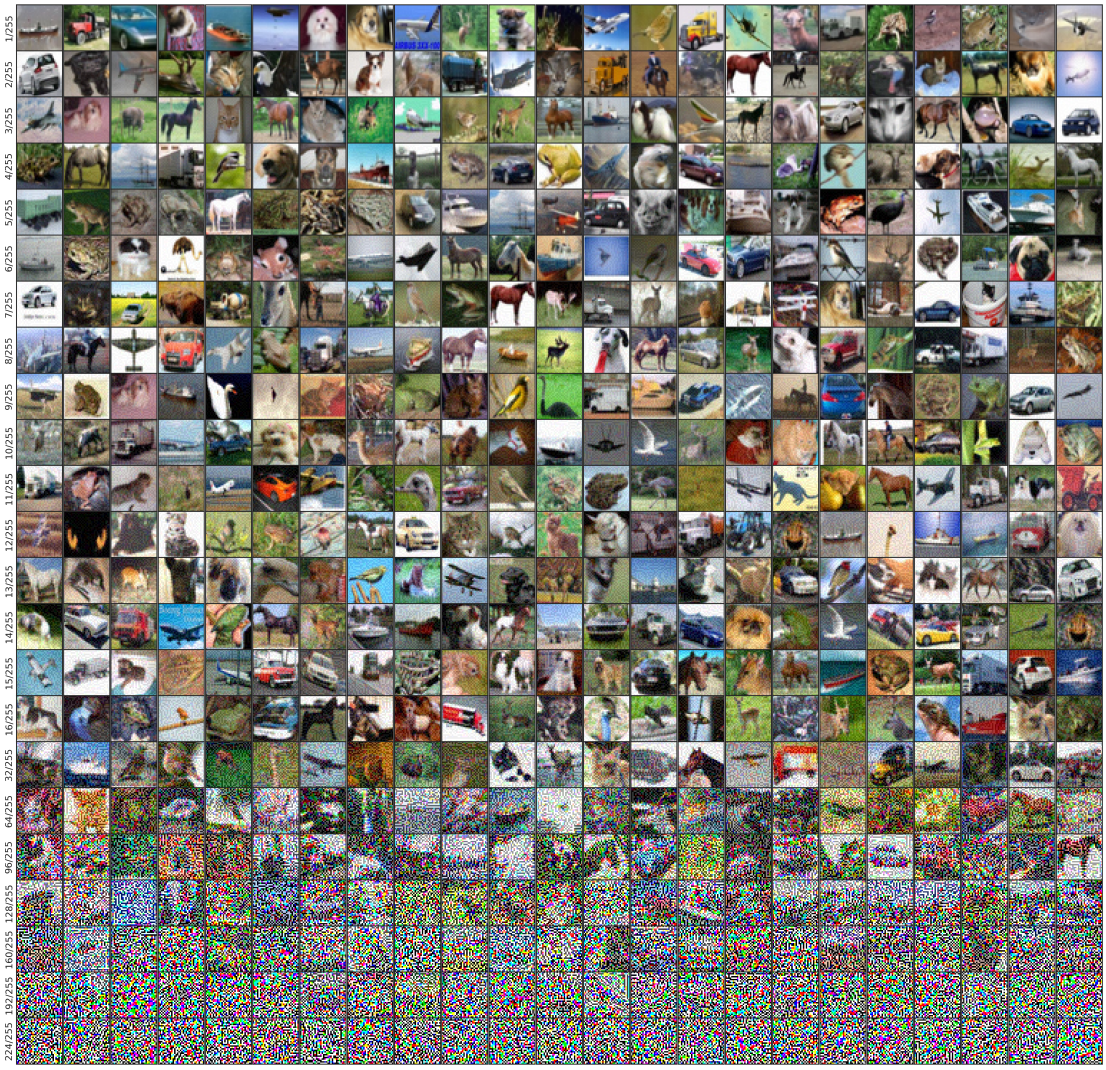

Fig. 6. Adversarial examples created using FSGM attack. On the x-axis, we can see the different adversarially perturbed images from the dataset. The different strengths of the level of the attacks (ϵ) are plotted in the y-axis. The level of the attack (ϵ) determines the strength of the adversarial attack; the higher the value, the stronger the adversarial attack and the more perturbed the input image (Best viewed in color).

demonstrated superior resilience in adversarial conditions compared to other CL models, with less forgetting of previously learned classes. We also performed t-SNE visualizations and employed centered kernel alignment (CKA) similarity scoring to investigate the stability of feature distributions, as well as to understand the learned feature representations and their similarities among various CL approaches. We observed that models with features more aligned with ADRM (i.e., exhibiting a higher similarity score) demonstrated enhanced robustness to both natural and adversarial noises and experienced reduced catastrophic forgetting. These insights deepen our understanding of feature representation learning within the realm of continual learning, underscoring the importance of continually-robust feature learning for CL models to avert catastrophic forgetting.

V. Acknowledgement

This work was supported by National Science Foundation (NSF) Award NSF OAC 2008690

References

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

[2] M. De Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars, "A continual learning survey: Defying forgetting in classification tasks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 7, pp. 3366–3385, 2021.

[3] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," in *Psychology of learning and motivation*. Elsevier, 1989, vol. 24, pp. 109–165.

[4] H. Khan, N. C. Bouaynaya, and G. Rasool, "Adversarially robust continual learning," in *2022 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2022, pp. 1–8.

[5] D. Rolnick, A. Ahuja, J. Schwarz, T. Lillicrap, and G. Wayne, "Experience replay for continual learning," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

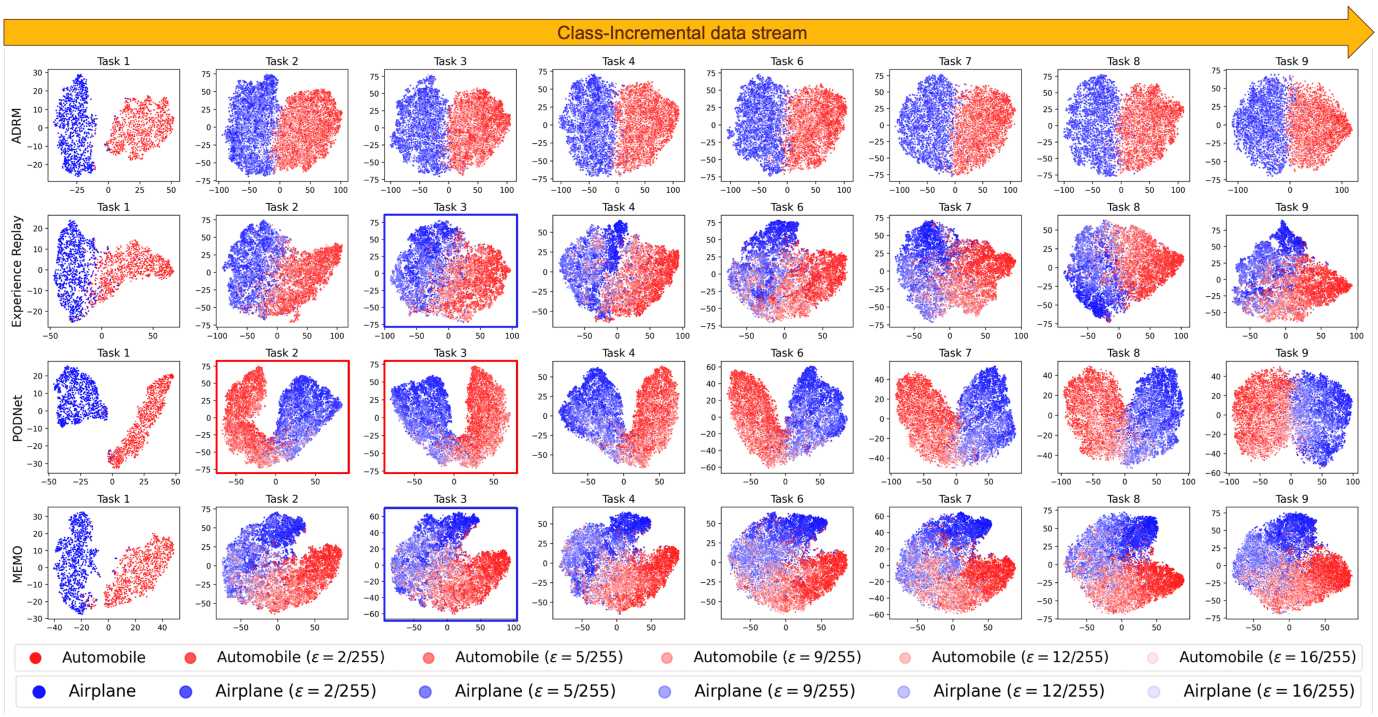

Fig. 7. Presents t-SNE visualizations of feature distributions for 'airplane' and 'automobile' classes in the CIFAR-10 dataset, under adversarial perturbations with ϵ values $\frac{2}{255}$, $\frac{5}{255}$, $\frac{3}{255}$, $\frac{9}{255}$, $\frac{12}{255}$ and $\frac{16}{255}$ across nine learning tasks. The original 'airplane' and 'automobile' class samples are denoted by dark blue and red dots, respectively, with lighter shades representing increased levels of adversarial perturbation. The sub-figures a), b), c), and d) display t-SNE visualizations for ADRB, ER, PODNet, and MEMO models, respectively, using features extracted from the last layer of ResNet32. The first column in each sub-figure presents the feature distribution for the initial two-class (airplane and automobile) learning task, highlighting the distinct separation achieved by all models. The subsequent columns illustrate each model's feature distribution evolution after learning subsequent tasks. In sub-figure a), Throughout the nine task learning, the ADRM model shows stable feature distributions for the 'airplane' and 'automobile' classes, with no shifts and minimum overlap. The other models exhibit an increasing trend of high overlap between the feature distributions (highlighted in a blue rectangle) and also suffer from feature distribution shifts (highlighted in a red rectangle). PODNet in sub-figure c) is the second-best performer, but it suffers from feature distribution shifts. On the other hand, MEMO in sub-figure d) suffers from the highest overlapping of feature distributions, which indicates the highest forgetting and lower performance in adversarial conditions (best viewed in color).

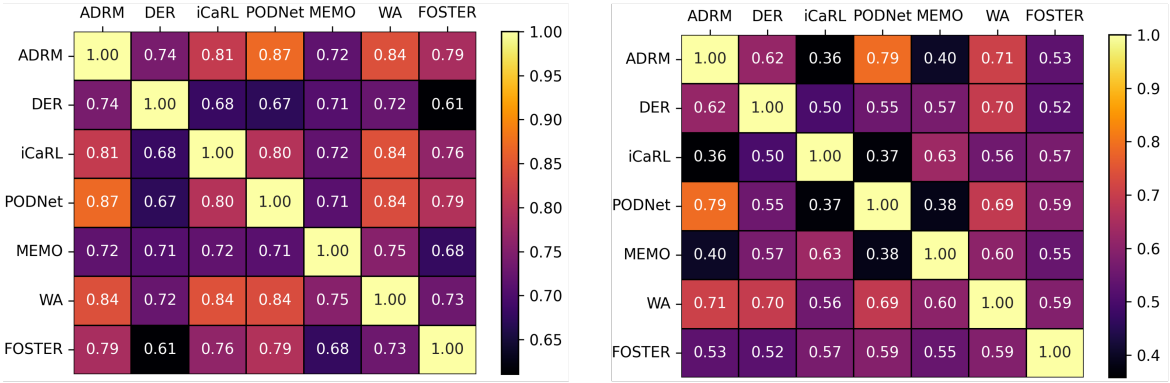

Fig. 8. Features similarity matrices calculated using centered kernel alignment (CKA). On the standard CIFAR-10, the ADRM and the other CL approaches exhibit high feature similarity in learned representation. In contrast, low feature similarities are observed on the adversarially perturbed CIFAR-10 dataset. PODNet, DER, and WA, with high similarity scores to ADRM, also exhibit enhanced performance in adversarial conditions and suffer less catastrophic forgetting. In contrast, models with lower feature similarity to ADRM suffer more in adversarial conditions, highlighting the crucial trade-off between robustness, generalization, and catastrophic forgetting within the CL domain. The matrices point towards the continually adversarially robust features that enhance the overall robustness of CL models and prevent catastrophic forgetting(best viewed in color).

Fig. 9. Visualizations of disentangled features for the 'automobile' (or 'car') class using the framework presented in [46] in the input space. 1st row presents the ADRM learned features that concentrate on salient class features, such as tires and body frames, resulting in less catastrophic forgetting in adversarial conditions. On the other hand, 2nd, 3rd, and 4th rows present the features for Experience Replay, MEMO models, comprising incoherent activations and non-class-specific artifacts, aligning with their lower performances in adversarial scenarios (best viewed in color).

[6] H. Khan, N. C. Bouaynaya, and G. Rasool, "Brain-inspired continual learning: Robust feature distillation and re-consolidation for class incremental learning," *IEEE Access*, vol. 12, pp. 34 054–34 073, 2024.

[7] ——, "The importance of robust features in mitigating catastrophic forgetting," in *2023 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, 2023, pp. 752–757.

[8] H. Khan, P. M. Shah, S. F. A. Zaidi *et al.*, "Susceptibility of continual learning against adversarial attacks," *arXiv preprint arXiv:2207.05225*, 2022.

[9] Z. Wang, L. Shen, Q. Suo, T. Duan, Y. Zhu, T. Liu, and M. Gao, "Make memory buffer stronger in continual learning: A continuous neural transformation approach," 2022.

[10] X. Chen, H. Wu, and X. Shi, "Consistent prototype learning for few-shot continual relation extraction," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2023, pp. 7409–7422.

[11] W. Shi and M. Ye, "Prototype reminiscence and augmented asymmetric knowledge aggregation for non-exemplar class-incremental learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 1772–1781.

[12] Z. Wang, L. Shen, L. Fang, Q. Suo, T. Duan, and M. Gao, "Improving task-free continual learning by distributionally robust memory evolution," in *International Conference on Machine Learning*. PMLR, 2022, pp. 22 985–22 998.

[13] F. Ye and A. G. Bors, "Learning an evolved mixture model for task-free continual learning," in *2022 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2022, pp. 1936–1940.

[14] X. Jin, J. Du, and X. Ren, "Gradient based memory editing for task-free continual learning," in *4th Lifelong Machine Learning Workshop at ICML 2020*, 2020.

[15] J. Bang, H. Kim, Y. Yoo, J.-W. Ha, and J. Choi, "Rainbow memory: Continual learning with a memory of diverse samples," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 8218–8227.

[16] Y.-M. Tang, Y.-X. Peng, and W.-S. Zheng, "Learning to imagine: Diversify memory for incremental learning using unlabeled data," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 9549–9558.

[17] D. Hendrycks and T. Dietterich, "Benchmarking neural network robustness to common corruptions and perturbations," in *International Conference on Learning Representations (ICLR)*, 2018.

[18] L. Engstrom, A. Ilyas, S. Santurkar, D. Tsipras, B. Tran, and A. Madry, "Adversarial robustness as a prior for learned representations," *arXiv preprint arXiv:1906.00945*, 2019.

[19] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.

[20] S. Kornblith, M. Norouzi, H. Lee, and G. Hinton, "Similarity of neural network representations revisited," in *International conference on machine learning*. PMLR, 2019, pp. 3519–3529.

[21] E. Wong, L. Rice, and J. Z. Kolter, "Fast is better than free: Revisiting adversarial training," in *International Conference on Learning Representations*, 2019.

[22] A. Chaudhry, M. Rohrbach, M. Elhoseiny, T. Ajanthan, P. Dokania, P. Torr, and M. Ranzato, "Continual learning with tiny episodic memories," in *Workshop on Multi-Task and Lifelong Reinforcement Learning*, 2019.

[23] X. Jin, A. Sadhu, J. Du, and X. Ren, "Gradient-based editing of memory examples for online task-free continual learning," *Advances in Neural Information Processing Systems*, vol. 34, pp. 29 193–29 205, 2021.

[24] A. Shafahi, M. Najibi, M. A. Ghiasi, Z. Xu, J. Dickerson, C. Studer, L. S. Davis, G. Taylor, and T. Goldstein, "Adversarial training for free!" *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[25] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[26] B. Zhao, X. Xiao, G. Gan, B. Zhang, and S.-T. Xia, "Maintaining discrimination and fairness in class incremental learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 13 208–13 217.

[27] Y. Wu, Y. Chen, L. Wang, Y. Ye, Z. Liu, Y. Guo, and Y. Fu, "Large scale incremental learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 374–382.

[28] D.-W. Zhou, H.-J. Ye, D.-C. Zhan, and Z. Liu, "Revisiting class-incremental learning with pre-trained models: Generalizability and adaptivity are all you need," *arXiv preprint arXiv:2303.07338*, 2023.

[29] D. Hendrycks and T. Dietterich, "Benchmarking neural network robustness to common corruptions and perturbations," in *International Conference on Learning Representations*, 2018.

[30] M. Masana, X. Liu, B. Twardowski, M. Menta, A. D. Bagdanov, and J. Van De Weijer, "Class-incremental learning: survey and performance evaluation on image classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 5, pp. 5513–5533, 2022.

[31] G. M. Van de Ven and A. S. Tolias, "Three scenarios for continual learning," *Advances in Neural Information Processing Systems (NIPS)*, 2019.

[32] D. Lopez-Paz and M. Ranzato, "Gradient episodic memory for continual learning," *Advances in Neural Information Processing Systems (NIPS)*, vol. 30, 2017.

[33] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "icarl: Incremental classifier and representation learning," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017, pp. 2001–2010.

[34] A. Douillard, M. Cord, C. Ollion, T. Robert, and E. Valle, "Podnet: Pooled outputs distillation for small-tasks incremental learning," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16*. Springer, 2020, pp. 86–102.

[35] S. Yan, J. Xie, and X. He, "Der: Dynamically expandable representation for class incremental learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3014–3023.

[36] F.-Y. Wang, D.-W. Zhou, H.-J. Ye, and D.-C. Zhan, "Foster: Feature boosting and compression for class-incremental learning," in *European conference on computer vision*. Springer, 2022, pp. 398–414.

[37] G. Petit, A. Popescu, H. Schindler, D. Picard, and B. Delezoide, "Fetril: Feature translation for exemplar-free class-incremental learning," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 3911–3920.

[38] D.-W. Zhou, Q.-W. Wang, H.-J. Ye, and D.-C. Zhan, "A model or 603 exemplars: Towards memory-efficient class-incremental learning," *arXiv preprint arXiv:2205.13218*, 2022.

[39] A. Chaudhry, M. Rohrbach, M. Elhoseiny, T. Ajanthan, P. K. Dokania, P. H. Torr, and M. Ranzato, "On tiny episodic memories in continual learning," *arXiv preprint arXiv:1902.10486*, 2019.

[40] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.

[41] D.-W. Zhou, F.-Y. Wang, H.-J. Ye, and D.-C. Zhan, "Pycil: A python toolbox for class-incremental learning," 2023.

[42] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[43] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2010, pp. 249–256.

[44] K. Alomar, H. I. Aysel, and X. Cai, "Data augmentation in classification and segmentation: A survey and new strategies," *Journal of Imaging*, vol. 9, no. 2, p. 46, 2023.
[45] A. Douillard, A. Ramé, G. Couairon, and M. Cord, "Dytox: Transformers for continual learning with dynamic token expansion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 9285–9295.
[46] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry, "Adversarial examples are not bugs, they are features," *Advances in Neural Information Processing Systems (NIPS)*, vol. 32, 2019.