Learning Robust Representations for Defending White-Box Adversarial Attacks in Continual Learning

Yusen Chen[1], Fei Ye[1]*, Qihe Liu[1], Adrian G. Bors[2], Shijie Zhou[1]
[1]School of Information and Software Engineering
University of Electronic Science and Technology of China
Chengdu 610054, China
[2]Department of Computer Science, University of York, York YO10 5GH, UK
202552090805@std.uestc.edu.cn, adrian.bors@york.ac.uk
{feiye,qiheliu,sjzhou}@uestc.edu.cn

Abstract

Continual learning under adversarial perturbations remains underexplored, despite its importance in security-critical deployments where attack patterns evolve over time. Existing continual adversarial defense methods mainly rely on replay or prediction-space regularization, but often fail to preserve robust representations across sequentially arriving attack types. In this paper, we study continual adversarial defense, where a model receives a sequence of tasks induced by different white-box attacks and must acquire robustness to new attacks without forgetting previously learned defenses. We address this challenging learning scenario by proposing the Learning Robust Representation Framework (LRRF), a representation-centric framework that improves robustness through hierarchical alignment at three complementary levels. First, we introduce the Dynamic Representation Matching (DRM) mechanism that aligns clean and adversarial feature distributions within each task to reduce the clean-robustness trade-off. Second, a Cross-Task Invariant Representation (CTIR) mechanism is proposed to regularize the current representation network against an accumulated network from previous tasks, encouraging attack-invariant features to persist over time. Third, a Knowledge Consolidation Optimization (KCO) mechanism is proposed to match class-consistent clean and adversarial features using replayed samples, which further stabilizes category structure across tasks. The empirical results show that the proposed approach achieves state-of-the-art performance.

1 Introduction

Continual Learning (CL) has emerged as one of the most prominent research areas within computer vision due to its ability to facilitate the acquisition of new knowledge while retaining previously acquired information [1]. Nevertheless, a central challenge in CL is catastrophic forgetting, which can markedly impair model performance on prior tasks. This issue frequently arises when the model updates all of its parameters to accommodate new task learning, thereby overwriting established representations [2].

Recent advancements have introduced a variety of techniques to mitigate catastrophic forgetting in continual learning (CL), demonstrating impressive efficacy [3, 4]. Nonetheless, these approaches predominantly assume access to pristine training datasets, overlooking more practical scenarios where

*Corresponding author. Our code is available at: https://github.com/yusen626/Learning-Robust-Representations-for-Defending-White-Box-Adversarial-Attacks-in-Continual-Learning.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

data are often compromised by noise and adversarial perturbations. When CL models are optimized under such realistic conditions, their robustness significantly degrades, particularly in the face of adversarial attacks [5]. Notably, white-box adversarial attack methods pose a greater threat than black-box attacks by exploiting direct access to model gradients and training data, thereby enhancing their attack potency. In this paper, we aim to bridge this gap by enhancing the robustness of CL models against white-box adversarial attacks, a topic that remains underexplored in the field.

Numerous studies have introduced new methodologies to improve the adversarial robustness of models within continual learning frameworks and such a learning scenario is called Continual Adversarial Defense (CAD) [6, 7]. Anisotropic and Isotropic Replay (AIR) [6] presents a data replay mechanism that utilizes three distinct replay strategies to preserve model performance on previously encountered tasks. Furthermore, AIR incorporates an additional regularization loss function to enforce consistency between the model's outputs on original training samples and their augmented counterparts. Despite these strengths, AIR's reliance on three separate replay strategies substantially increases computational overhead. Another approach differentiates logit calibration for prior and current tasks to better modulate gradient flow and implements a gradient-based sample selection process for retaining pivotal samples [7]. Nevertheless, this method primarily targets the logit and prediction space optimization, overlooking the importance of learning robust feature representations, which consequently results in suboptimal model robustness.

In this work, we argue that representation alignment provides a simple and effective principle for CAD. Robust continual defense requires at least three properties: (i) within each task, clean and adversarial samples should remain semantically consistent in feature space; (ii) across tasks, newly learned representations should remain compatible with previously acquired robust features; and (iii) replayed clean samples should help preserve class structure while the model adapts to new attacks. Based on this view, we propose Learning Robust Representation Framework (LRRF), a framework that enforces hierarchical feature alignment at the intra-task, cross-task, and class-conditional levels. To address the intra-task robustness trade-off, the proposed LRRF involves a novel Dynamic Representation Matching (DRM) mechanism that facilitates the alignment of semantic representations between clean and adversarial samples. This alignment enables the model to establish a strong decision boundary within the shared representation space for both clean and adversarial instances, thereby enhancing inner-task robustness.

Given that each incoming task presents adversarial samples generated by distinct adversarial algorithms, reliance solely on DRM fails to facilitate robust feature learning transferable across different tasks. To overcome this limitation, we introduce the Cross-Task Invariant Representation (CTIR) mechanism, which incorporates an accumulated representation network designed to retain the trained parameters of the representation network following each task transition. This accumulated representation network ensures the retention of previously acquired robust feature information. Consequently, the CTIR mechanism promotes the learning of invariant representations across diverse tasks by minimizing the probabilistic distance between representations derived from the current and accumulated networks. Furthermore, to further alleviate catastrophic forgetting and improve overall model efficacy, we introduce the Knowledge Consolidation Optimization (KCO) mechanism, which utilizes a compact memory buffer to retain a selection of clean training examples. Throughout the optimization procedure, the KCO mechanism identifies pairs of clean and adversarial instances within the same class and minimizes the probabilistic divergence between their respective feature representations. This approach aims to learn a unified category feature space for both clean and adversarial samples across multiple tasks.

We conduct a series of experiments, and the empirical findings confirm that our method attains state-of-the-art performance. The main contributions of this paper are as follows : (1) We formulate continual adversarial defense from a representation learning perspective and show that robust feature preservation is a central challenge beyond standard replay-based forgetting mitigation; (2) We propose LRRF, a hierarchical alignment framework with three complementary components: DRM for intra-task clean-adversarial alignment, CTIR for cross-task robust feature preservation, and KCO for class-conditional consolidation with replayed clean samples; (3) We conduct extensive experiments on five benchmarks under sequential white-box attack settings, showing that LRRF improves robustness and forgetting behavior over strong continual learning and robustness-aware baselines.

2 Related Work

Rehearsal-based approaches aim to manage a memory buffer for preserving many critical examples [8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19]. The samples drawn from the memory buffer can be

combined with samples from a new task to re-update the model, aiming to learn both new and old knowledge. Therefore, the sample quality in the memory buffer plays a critical role in the model's performance [16]. Additionally, the rehearsal-based techniques can be combined with regularization-based approaches, improving performance [10, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30].

Dynamic network architectures aim to learn a growing number of hidden layers and nodes over time within a unified optimization framework [31]. Unlike the static network architecture, which has a fixed model capacity, dynamic network architectures can incrementally learn and capture novel information without forgetting all previously learned knowledge [32, 33, 31, 34, 35, 36, 37, 2, 38]. The best performance on previous tasks can be maintained by freezing all previously learned parameters [33, 31, 34, 14, 35, 36, 37, 2, 38]. In recent years, the dynamic expansion methods have explored using the Vision Transformers (ViT) [39] for boosting the model's performance [40, 41]. A notable ViT-based dynamic expansion approach proposes to incrementally build new self-attention blocks as well as linear classifiers, aiming to effectively capture new information and knowledge from new tasks. More recently, using the pre-trained ViT has been explored in continual learning [3], which explores prior knowledge to facilitate learning on new tasks.

Adversarial defense constitutes a significant area of research aimed at enhancing the robustness of models against various adversarial attacks. Previous investigations in this domain have predominantly concentrated on heuristic methodologies, including model ensembling [42], adversarial denoising [43], and input transformation techniques [5]. Recently, numerous studies have advanced the concepts of adversarial training [44, 45, 46] and defensive distillation [47, 48, 49]. Defensive distillation, as introduced in [50], leverages a teacher-student paradigm to improve model robustness. Furthermore, [51] advocates for the generation of soft labels to facilitate the training of a student model from extensively trained teacher models. The implementation of adversarial defense can also be achieved through an ensemble architecture, where each expert network is encouraged to learn distinct feature representations [52]. Moreover, the exploration of adversarial defense within the context of continual learning has been addressed in [6], which utilizes memory replay techniques. In contrast to [6], the proposed methodology focuses on learning invariant representations to sustain optimal performance on clean datasets while simultaneously enhancing adversarial robustness. More discussions are shown in **Appendix-A**.

3 Methodology

We view CAD as a problem of preserving representation geometry under sequentially changing adversarial perturbations. Our design decomposes this goal into three complementary alignment constraints: intra-task alignment between clean and adversarial samples, cross-task alignment between current and historical robust representations, and class-conditional alignment between current adversarial samples and replayed clean samples.

3.1 Problem Definition

In CL, a model is continuously trained to assimilate new information from an evolving environment, with the primary objective of mitigating catastrophic forgetting. Prevailing research in continual learning predominantly concentrates on class-incremental learning, wherein novel category data samples are introduced as new tasks [53]. Unlike conventional class-incremental learning, we study the existing but underexplored setting of continual adversarial defense, where each distinct adversarial attack is treated as an independent task. Let $\mathcal{D}^S = \{\mathbf{x}_j, \mathbf{y}_j\}_{j=1}^{n_S}$ represent the training dataset comprising n_S samples, and $\mathcal{D}^T = \{\mathbf{x}_j, \mathbf{y}_j\}_{j=1}^{n_T}$ indicate its corresponding test set consisting of n_T samples. For a sequence of attack algorithms $\{\mathcal{B}_1, \cdots, \mathcal{B}_{n-1}\}$, each attack $\mathcal{B}_j$ can be leveraged to convert the original training dataset $\mathcal{D}^S$ into an adversarial dataset $\tilde{\mathcal{D}}_j^S = \{\tilde{\mathbf{x}}_c, \mathbf{y}_c\}_{c=1}^{n_S}$, where n denotes the total number of tasks and $\tilde{\mathbf{x}}$ denotes adversarial examples generated via the procedure $\tilde{\mathbf{x}} = F_g(\mathcal{B}_j, f_{\theta^\star}, \mathbf{x})$. Here, $f_{\theta^\star}$ represents a model optimized on the benign training set $\mathcal{D}^S$. The continual adversarial defense setting typically delineates a sequence of n tasks $\{\mathcal{T}_1, \cdots, \mathcal{T}_n\}$, where the initial task ($\mathcal{T}_1$) aligns with the clean dataset, denoted as $\mathcal{D}_1^S$, and each subsequent task $\mathcal{T}_j$ is linked to the $(j-1)$-th adversarial dataset $\tilde{\mathcal{D}}_{j-1}^S$. Throughout the j-th learning phase ($\mathcal{T}_j$), the main goal of a model in CAD is to find the optimal parameter set $\theta^\star$ that minimizes the loss values across all previously learned tasks :

$$\theta^\star = \underset{\theta \in \Theta}{\operatorname{argmin}} \Big\{ \sum\nolimits_{c=2}^{j-1} \left\{ \mathbb{E}_{(\tilde{\mathbf{x}},\mathbf{y}) \sim P_{\tilde{\mathcal{D}}_c^S}} [f'(\mathbf{y}, f_\theta(\tilde{\mathbf{x}}))] \right\} + \mathbb{E}_{(\mathbf{x},\mathbf{y}) \sim P_{\mathcal{D}_1^S}} [f'(\mathbf{y}, f_\theta(\mathbf{x}))] \Big\}, \tag{1}$$

where Θ denotes the parameter space of the model. $P_{\mathcal{D}_1^S}$ and $P_{\tilde{\mathcal{D}}_c^S}$ represents the probability distribution associated with the dataset $\mathcal{D}_1^S$ and $\tilde{\mathcal{D}}_c^S$. The function $f'(\cdot,\cdot)$ specifies a loss function. Nevertheless, directly optimizing the model via Eq. (1) is infeasible within the continual learning paradigm, as prior data samples are unavailable. To mitigate this limitation, numerous approaches have introduced the use of a memory buffer to retain a limited subset of previous examples. Upon completion of the final task ($\mathcal{T}_n$), we apply adversarial attack strategies $\{\mathcal{B}_1, \cdots, \mathcal{B}_{n-1}\}$ to the test dataset $\mathcal{D}^T$, yielding a series of $n-1$ adversarial datasets, denoted as $\{\tilde{\mathcal{D}}_1^T, \cdots, \tilde{\mathcal{D}}_{n-1}^T\}$. The model's robustness and accuracy are then assessed on both the clean test dataset $\mathcal{D}^T$ and the collection of adversarial datasets $\{\tilde{\mathcal{D}}_1^T, \cdots, \tilde{\mathcal{D}}_{n-1}^T\}$.

3.2 Dynamic Representation Matching Mechanism

The majority of current adversarial defense strategies typically involve training models on a combination of clean and adversarial examples [44], with the objective of enhancing model robustness. Nevertheless, these approaches often result in a degradation of performance on clean data, a problem that becomes particularly pronounced in the context of continual adversarial defense learning [54]. In this study, we tackle this challenge by developing an invariant feature representation for both clean and adversarial inputs, thereby bolstering robustness without compromising performance on clean samples. To achieve this, we introduce a novel Dynamic Representation Matching (DRM) mechanism, which progressively aligns the representations of clean and adversarial examples by minimizing their representational distance.

Let us define $f_{\theta_i^f} : \mathcal{X} \to \mathcal{Z}$ as a representation network parameterized by θ_i^f, which takes an image input and generates a feature embedding $\mathbf{z}$ within the embedding space $\mathcal{Z}$. Here, i indexes the parameter set specific to the i-th task increment. Additionally, we denote $f_{\theta_i^c} : \mathcal{Z} \to \mathcal{Y}$ as a linear classification model with parameters θ_i^c, mapping the feature vector $\mathbf{z}$ to a class probability distribution $\mathbf{y} = \{y_1, \cdots, y_C\}$, in which y_j corresponds to the predicted likelihood of the j-th class. Specifically, the representation network $f_{\theta_i^f}$ comprises L hierarchical feature layers $\{f_{\theta_i^f}[1], \cdots, f_{\theta_i^f}[L]\}$, where each feature layer $f_{\theta_i^f}[j]$ yields a feature vector output $\mathbf{z}^j = f_{\theta_i^f}[j](\mathbf{x}_c)$ for a sample input $\mathbf{x}_c$. Considering a collection of m clean training instances in the initial learning task $\mathcal{T}_1$, denoted by $(\mathbf{X}^1, \mathbf{Y}^1) = \{(\mathbf{x}_1, \mathbf{y}_1), \cdots, (\mathbf{x}_m, \mathbf{y}_m)\}$ (with m representing the mini-batch size and the superscript "1" indicating the task index), we establish an adversarial perturbation generation process as follows :

$$\gamma = \xi \text{sign}(\nabla_{\mathbf{x}} \mathcal{L}(\theta, \mathbf{X}^1, \mathbf{Y}^1)), \tag{2}$$

where γ denotes the adversarial perturbation, while $\mathcal{L}(\cdot)$ refers to the loss function. The term $\nabla_{\mathbf{x}} \mathcal{L}(\theta)$ specifies the gradient of the loss with respect to the input, calculated via backpropagation, and ξ serves as the attack parameter. Utilizing Eq. (2), a collection of adversarial examples $\tilde{\mathbf{X}}^1 = \{\tilde{\mathbf{x}}_1, \cdots, \tilde{\mathbf{x}}_m\}$ can be synthesized by superimposing the perturbation γ. Since the representation network $f_{\theta_1^f}$ has multi-level feature layers, we can define a function that extracts the representations from a specific feature layer, expressed as :

$$F_e(\mathbf{X}^1, j) = \{\mathbf{z}^j \,|\, \mathbf{z}^j = f_{\theta_1^f}[j](\mathbf{x}_c), c = 1, \cdots, m\}, \tag{3}$$

where j indexes the feature vectors derived from the j-th layer, while $\mathbf{x}_c$ denotes the c-th instance from the set $\mathbf{X}^1$. We define $\mathbf{Z}^j(1) = F_e(\mathbf{X}^1, j)$ and $\tilde{\mathbf{Z}}^j(1) = F_e(\tilde{\mathbf{X}}^1, j)$ as the collections of feature embeddings for $\mathbf{X}^1$ and $\tilde{\mathbf{X}}^1$, obtained from the j-th representation layer of the feature extraction network $f_{\theta_i^f}$, respectively. The empirical distributions for these feature sets are denoted by $P_{\mathbf{Z}^j(1)}$ and $P_{\tilde{\mathbf{Z}}^j(1)}$, respectively. Within this framework, the DRM mechanism is designed to facilitate the learning of invariant feature representations by minimizing the probabilistic divergence between the sets of distributions $\{P_{\mathbf{Z}^1(1)}, \ldots, P_{\mathbf{Z}^L(1)}\}$ and $\{P_{\tilde{\mathbf{Z}}^1(1)}, \ldots, P_{\tilde{\mathbf{Z}}^L(1)}\}$. To achieve this, we implement the probability distance using the Maximum Mean Discrepancy (MMD) metric [55], selected for several key advantages : (1) Compared to other distance measures such as Kullback–Leibler and Jensen–Shannon divergence, the MMD criterion does not need the explicit probability density functions; (2) The MMD criterion enjoys the flexibility offered by the choice of kernel functions, which can be used to trade-off statistical efficiency and robustness [56]; Exploring alternative distance measures is left for future work.

The MMD-based regularization term. The Maximum Mean Discrepancy (MMD) is constructed based on embedding probability distributions into a Reproducing Kernel Hilbert Space (RKHS)

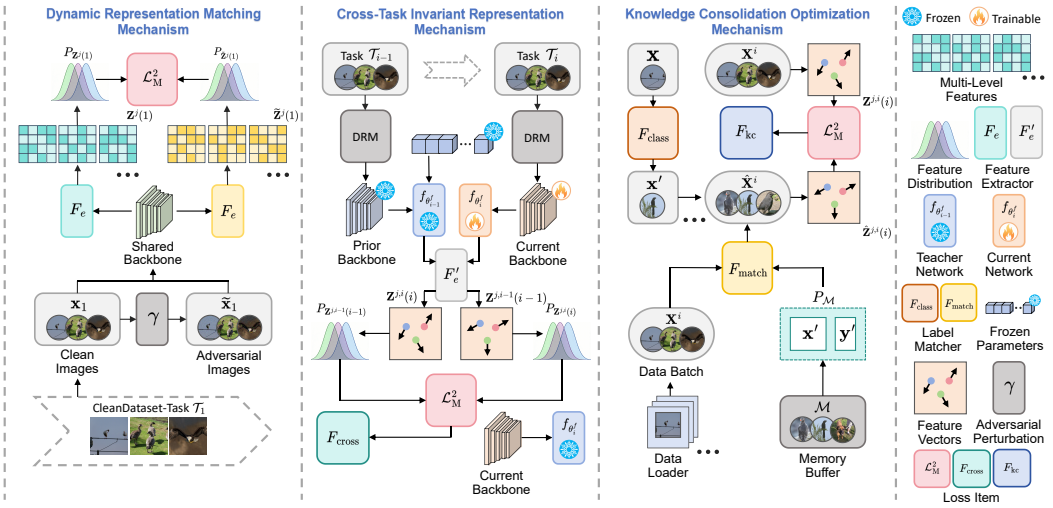


Figure 1: The LRRF framework includes three core mechanisms (left to right): (1) The Dynamic Representation Matching (DRM) mechanism yields gradient-based adversarial perturbations and aligns the representations of clean and adversarial examples via representational distance minimization. (2) The Cross-Task Invariant Representation (CTIR) mechanism fuses task-invariant features from prior tasks into novel task optimization to improve feature stability. (3) The Knowledge Consolidation Optimization (KCO) aligns category-consistent feature distributions between clean and adversarial samples to enhance structural robustness.

[55]. Consider A and B as two Borel probability measures, and let $\mathbf{a}$ and $\mathbf{b}$ denote random variables defined over a topological space $\mathcal{Z}$. Suppose $\{f \in \mathcal{C} | f : \mathcal{Z} \to \mathbf{R}\}$ is a set of functions, where $\mathcal{C}$ represents a particular function class. The MMD between A and B is then defined as follows [55] :

$$\mathcal{L}_{\mathrm{M}}(A, B) \triangleq \sup_{f \in \mathcal{C}} \left(\mathbb{E}_{\mathbf{a} \sim A} \left[f(\mathbf{a}) \right] - \mathbb{E}_{\mathbf{b} \sim B} \left[f(\mathbf{b}) \right] \right) , \tag{4}$$

where sup represents the supremum, or the least upper bound, of a set of real numbers. When two probability distributions coincide, i.e., $A = B$, we obtain $\mathcal{L}_{\mathrm{M}}(A, B) = 0$. Here, we define $\mathcal{C}$ as the unit ball within a reproducing kernel Hilbert space (RKHS) characterized by a positive definite kernel function $F_k(\mathbf{x}, \mathbf{x}')$. Nevertheless, Eq. (4) is typically computationally infeasible; therefore, the MMD criterion can be approximated in the feature embedding space, as described in [57], by :

$$\mathcal{L}^2_{\mathrm{M}}(A, B) = ||\boldsymbol{\mu}_A - \boldsymbol{\mu}_B||^2 , \tag{5}$$

where $\boldsymbol{\mu}_A = \int F_k(\mathbf{a}, \cdot) \frac{\partial A(\mathbf{a})}{\partial \mathbf{a}} d\mathbf{a}$ and $\boldsymbol{\mu}_B = \int F_k(\mathbf{b}, \cdot) \frac{\partial B(\mathbf{b})}{\partial \mathbf{b}} d\mathbf{b}$ represent the kernel mean embeddings of the distributions A and B, respectively, with $A(\mathbf{a})$ being the probability density function associated with A. Moreover, $\boldsymbol{\mu}_A$ satisfies $\mathbb{E}[f(\mathbf{x})] = \langle f, \boldsymbol{\mu}_P \rangle_{\mathcal{H}}$, where $\langle f, \cdot \rangle_{\mathcal{H}}$ denotes the inner product in the RKHS. Given the reproducing property of the RKHS, where $f \in \mathcal{C}$ satisfies $f(\mathbf{a}) = \langle f, F_k(\mathbf{a}, \cdot) \rangle_{\mathcal{H}}$, we can thus reformulate Eq. (5) in terms of the associated kernel functions as follows :

$$\mathcal{L}^2_{\mathrm{M}}(A, B) = \mathbb{E}_{\mathbf{a}, \mathbf{a}' \sim A}[k(\mathbf{a}, \mathbf{a}')] - 2\mathbb{E}_{\mathbf{a} \sim A, \mathbf{b} \sim B}[k(\mathbf{a}, \mathbf{b})] + \mathbb{E}_{\mathbf{b}, \mathbf{b}' \sim B}[k(\mathbf{b}, \mathbf{b}')] , \tag{6}$$

where $\mathbf{b}'$ and $\mathbf{a}'$ represent independent realizations of $\mathbf{b}$ and $\mathbf{a}$, respectively. In practical applications, we compute the MMD criterion by drawing an equal number of samples from the distributions A and B, thereby obtaining an unbiased empirical estimate, which is given by :

$$\mathcal{L}^e_{\mathrm{M}}(A, B) = \frac{1}{N'(N'-1)} \sum_{i \neq j}^{N'} \Big\{ h(i, j) \Big\} , \tag{7}$$

where $h(i, j) = F_k(\mathbf{a}_i, \mathbf{a}_j) + F_k(\mathbf{b}_i, \mathbf{b}_j) - F_k(\mathbf{a}_i, \mathbf{b}_j) - F_k(\mathbf{a}_j, \mathbf{b}_i)$. In accordance with the formulation provided in Eq. (7), the proposed DRM method formulates the regularization term by aggregating representations across all feature layers, as follows :

$$F_r(\mathbf{X}^1, \tilde{\mathbf{X}}^1) = \frac{1}{L} \sum_{l=1}^{L} \{ \mathcal{L}^2_{\mathrm{M}}(P_{\mathbf{Z}^l(1)}, P_{\tilde{\mathbf{Z}}^l(1)}) \} , \tag{8}$$

where $\tilde{\mathbf{X}}^1$ denotes an adversarial subset generated from the data batch $\mathbf{X}^1$ procured during training on the initial task ($\mathcal{T}_1$). This regularization component incentivizes the model to acquire invariant feature representations across both unperturbed and adversarial inputs. Leveraging $F_r(\mathbf{X}^1, \tilde{\mathbf{X}}^1)$, the final objective function for model optimization in the context of the first task is thus expressed as :

$$\begin{aligned} F_{\text{obj}}(\mathbf{X}^1, \mathbf{Y}^1, \tilde{\mathbf{X}}^1) = \min_{\theta_1^f, \theta_1^c} \Big\{ \sum\nolimits_{j=1}^{m} \big\{ F_{\text{c}}(f_{\theta_1^c}(f_{\theta_1^f}(\mathbf{x}_j)), \mathbf{y}_j) \\ + \max_{||\mathbf{x}'-\mathbf{x}|| \leq \epsilon} \{ F_{\text{c}}(f_{\theta_1^c}(f_{\theta_1^f}(\mathbf{x}'_j)), \mathbf{y}_j) \} \big\} + \lambda F_r(\mathbf{X}^1, \tilde{\mathbf{X}}^1) \Big\}, \end{aligned} \tag{9}$$

where $\mathbf{x}'_j$ refers to an adversarially perturbed variant of the j-th data point $\mathbf{x}_j$, constrained by the perturbation budget ϵ. The inner maximization term corresponds to adversarial training under perturbation budget defined in Eq. (9), thereby enhancing adversarial robustness. $F_{\text{c}}(\cdot, \cdot)$ denotes the cross-entropy loss function, defined as :

$$F_{\text{c}}(\mathbf{y}'_j, \mathbf{y}_j) = -\sum\nolimits_{t=1}^{C} \left\{ \mathbf{y}_j[t] \log(\mathbf{y}'_j[t]) \right\}, \tag{10}$$

where $\mathbf{y}_j[t]$ and $\mathbf{y}'_j[t]$ represent the t-th element of the ground truth label vector $\mathbf{y}_j$ and its corresponding predicted output $\mathbf{y}'_j$, respectively. The parameter C denotes the number of unique class categories.

3.3 Learning Invariant Representation Across Tasks

While the proposed DRM mechanism demonstrates the ability to extract invariant features from both clean and adversarial inputs, it remains ineffective against previously unseen adversarial attacks or novel tasks. To overcome this limitation, we introduce an innovative Cross-Task Invariant Representation (CTIR) mechanism designed to capture invariant features across diverse adversarial attacks. Concretely, let $f_{\theta_i^f}$ represent the feature extraction network optimized during the i-th task ($\mathcal{T}_i$). In order to preserve all previously learned robust feature information, the proposed CTIR mechanism introduces an accumulated representation network to store the trained parameters of the representation network after each task switch. Let $f_{\theta_{i-1}^f}$ denote the accumulated representation network ($\mathcal{T}_{i-1}$), which has stored the parameters of the previously trained representation network $f_{\theta_{i-1}^f}$ and freezes all its parameters during the new task learning ($\mathcal{T}_i$).

Given a data batch $\mathbf{X}^i$ originating from the i-th task's training phase, the feature representation of $\mathbf{X}^i$ at a specific layer within the representation network can be derived via the feature extraction function, formulated as follows :

$$F'_e(\mathbf{X}^i, j, f_{\theta_i^f}) = \{ \mathbf{z}^{j,i} \,|\, \mathbf{z}^{j,i} = f_{\theta_i^f}[j](\mathbf{x}_c), c = 1, \cdots, m \}. \tag{11}$$

By utilizing the feature extraction function, we obtain the feature vectors $\mathbf{Z}^{j,i}(i) = F'_e(\mathbf{X}^i, j, f_{\theta_i^f})$ and $\mathbf{Z}^{j,i-1}(i-1) = F'_e(\mathbf{X}^i, j, f_{\theta_{i-1}^f})$ from the j-th layer of the active and accumulated representation networks, $f_{\theta_i^f}$ and $f_{\theta_{i-1}^f}$, respectively. Let us denote the distributions of $\mathbf{Z}^{j,i}(i)$ and $\mathbf{Z}^{j,i-1}(i-1)$ as $P_{\mathbf{Z}^{j,i}(i)}$ and $P_{\mathbf{Z}^{j,i-1}(i-1)}$. The proposed CTIR methodology formulates an objective function designed to minimize the discrepancy between these two distribution sets, which can be represented as :

$$F_{\text{cross}}(\mathbf{X}^i, f_{\theta_i^f}, f_{\theta_{i-1}^f}) = \frac{1}{L} \sum\nolimits_{l=1}^{L} \{ \mathcal{L}^2_{\text{M}}(P_{\mathbf{Z}^{l,i}(i)}, P_{\mathbf{Z}^{l,i-1}(i-1)}) \}. \tag{12}$$

Eq. (12) aims to learn an invariant representation for multiple adversarial attacks in a sequential manner, which can be integrated into the primary objective function to regulate the optimization behaviour of the representation network.

3.4 Knowledge Consolidation Optimization

In sequential task learning, the model is deprived of access to previously encountered clean datasets, which still leads to forgetting. To combat this issue, we present a novel methodology termed the Knowledge Consolidation Optimization (KCO) mechanism. This strategy strategically utilizes a limited-capacity memory buffer, $\mathcal{M}$, to retain a representative subset of clean data samples. The KCO approach leverages the cross-entropy loss function to stabilize model performance across clean datasets, denoted by $\mathbb{E}_{(\mathbf{x},\mathbf{y})\sim P_{\mathcal{M}}}[F_c(f_{\theta^c}(f_{\theta^f}(\mathbf{x})), \mathbf{y})]$, where $P_{\mathcal{M}}$ represents the probability

Table 1: The average classification accuracy of various models on low-to-medium complexity datasets.

Method	CIFAR10			CIFAR100			CUB200		
	None →	FGSM →	PGD	None →	FGSM →	PGD	None →	FGSM →	PGD
DER	**92.64**	57.09	42.57	**92.00**	49.01	35.54	84.74	44.07	47.68
DER++	92.52	57.61	43.19	91.92	49.23	35.43	85.12	45.21	48.28
DER+++	88.88	74.25	53.83	90.18	47.75	**44.98**	84.26	68.53	50.93
DEF	87.68	60.18	46.42	89.58	60.59	42.82	84.28	69.71	51.02
DER++STAR	86.73	58.04	39.73	88.43	47.56	33.38	**86.32**	68.90	46.36
ER-ACE-STAR	87.62	21.00	35.07	85.95	37.87	32.84	84.52	68.72	46.08
AIR	80.55	41.05	34.93	83.56	39.25	31.59	68.12	37.31	45.68
DER++AFLC	89.01	59.78	44.86	91.66	47.18	34.87	81.36	30.41	37.06
LRRF(Ours)	86.31	**83.14**	**58.83**	89.61	**78.05**	40.02	83.67	**71.53**	**54.15**

distribution over the memory buffer $\mathcal{M}$. To enhance the learning of invariant representations between clean and adversarial inputs, the proposed KCO method introduces a sample pairing strategy, enabling the identification of corresponding sample pairs as follows :

$$F_{\text{match}}(\mathbf{x}, \mathbf{y}) = \{(\mathbf{x}', \mathbf{y}') \mid F_{\text{class}}(\mathbf{x}') = \mathbf{y}, (\mathbf{x}', \mathbf{y}') \sim P_{\mathcal{M}}\}, \tag{13}$$

where $F_{\text{class}}(\cdot)$ denotes a function that returns the exact class label for an input and $\mathbf{x}'$ is a sample drawn from the memory distribution $P_{\mathcal{M}}$ and shares the same category with $\mathbf{x}$. For a given data batch $\mathbf{X}^i$ from the i-th task learning, we employ $F_{\text{match}}(\cdot, \cdot)$ to randomly find an appropriate example for each sample of $\mathbf{X}^i$ and these examples can form a data batch $\hat{\mathbf{X}}^i$, corresponding to $\mathbf{X}^i$. We can get the feature vectors $\mathbf{Z}^{j,i}(i) = F'_e(\mathbf{X}^i, j, f_{\theta_i^f})$ and $\hat{\mathbf{Z}}^{j,i}(i) = F'_e(\hat{\mathbf{X}}^i, j, f_{\theta_i^f})$ for $\mathbf{X}^i$ and $\hat{\mathbf{X}}^i$ using the j-th layer of the representation network $f_{\theta_i^f}$. Based on the feature vectors $\mathbf{Z}^{j,i}(i)$ and $\hat{\mathbf{Z}}^{j,i}(i)$, the proposed KCO approach aims to learn a unified category feature space for both clean and adversarial samples by deriving a regularization loss function, expressed as :

$$F_{\text{kc}}(\mathbf{X}^i, \hat{\mathbf{X}}^i) = \frac{1}{L} \sum\nolimits_{l=1}^{L} \{\mathcal{L}^2_{\text{M}}(P_{\mathbf{Z}^{l,i}(i)}, P_{\hat{\mathbf{Z}}^{l,i}(i)})\}, \tag{14}$$

where $P_{\mathbf{Z}^{l,i}(i)}$ and $P_{\hat{\mathbf{Z}}^{l,i}(i)}$ denote the distribution of $\mathbf{Z}^{l,i}(i)$ and $\hat{\mathbf{Z}}^{l,i}(i)$, respectively.

The final objective function for optimizing the model at the i-th task learning is defined as :

$$\begin{aligned} F'_{\text{obj}}(\mathbf{X}^i, \mathbf{Y}^i, \hat{\mathbf{X}}^i) = \min_{\theta_i^f, \theta_i^c} \Big\{ \sum\nolimits_{j=1}^{m} \big\{ F_{\text{c}}(f_{\theta_i^c}(f_{\theta_i^f}(\mathbf{x}_j)), \mathbf{y}_j) + \max_{||\mathbf{x}'-\mathbf{x}|| \le \epsilon} \{F_{\text{c}}(f_{\theta_i^c}(f_{\theta_i^f}(\mathbf{x}'_j)), \mathbf{y}_j)\} \big\} \\ + \lambda_2 \{F_{\text{cross}}(\mathbf{X}^i, f_{\theta_i^f}, f_{\theta_{i-1}^f}) + F_{\text{kc}}(\mathbf{X}^i, \hat{\mathbf{X}}^i)\} \Big\}, \end{aligned} \tag{15}$$

where we employ a single hyperparameter λ_2 to determine the importance of the regularization terms in order to reduce the hyperparameter search costs.

3.5 Theoretical Analysis

Preliminaries. In order to simplify notations and to improve readability, we define the following notations for theoretical analysis. Let $\mathcal{H}$ be the reproducing kernel Hilbert space (RKHS) induced by a kernel function $k(\cdot, \cdot)$. For any probability distribution P, its kernel mean embedding is defined as $\mu_P = \mathbb{E}_{z \sim P}[k(z, \cdot)] \in \mathcal{H}$. For two probability distributions P and Q, the MMD is defined as $\text{MMD}(P, Q) = \|\mu_P - \mu_Q\|_{\mathcal{H}}$. If k is a characteristic kernel, then the mapping $P \mapsto \mu_P$ is injective, and hence $\text{MMD}(P, Q) = 0 \iff P = Q$. Let us define P_i^{cur} as the representation distribution produced by the current network in task $\mathcal{T}_i$ and P_i^{acc} as the representation distribution produced by the accumulated network on the task $\mathcal{T}_i$. For the i-th task, let the representation network and classifier head be defined as :

$$f_i : \mathcal{X} \to \mathcal{Z}, \qquad g_i : \mathcal{Z} \to \Delta^{C-1}, \tag{16}$$

where Δ^{C-1} denotes the C-dimensional probability simplex. The overall model is therefore given by $h_i(\mathbf{x}) = g_i(f_i(\mathbf{x}))$.

Theoretical Analysis of CTIR: Cross-Task Stability and Forgetting Control. We bound the risk change between consecutive models on a fixed labeled distribution D. Write $R_D(h) = \mathbb{E}_{(X,Y) \sim D} \ell(h(X), Y)$ and $\pi_D(y) = \Pr_D(Y = y)$. Let $P^{\text{cur}}_{i,D,y}$ and $P^{\text{acc}}_{i,D,y}$ be the conditional laws of $f_i(X)$ and $f_{i-1}(X)$, respectively, given $Y = y$ under the same D. Define

$$d_i(D) = \sum_{y : \pi_D(y) > 0} \pi_D(y) \text{MMD}(P^{\text{cur}}_{i,D,y}, P^{\text{acc}}_{i,D,y}). \tag{17}$$

Let $b_i(D)$ denote the expected absolute loss change from updating the classifier head with f_{i-1} fixed (Appendix D.4).

Theorem 3.1 (One-Step Risk Drift Bound)**.** *Assume that the kernel mean embeddings exist, the relevant risks and $b_i(D)$ are finite, and, for each y with $\pi_D(y) > 0$, $\varphi_{i,y}(z) = \ell(g_i(z), y) \in \mathcal{H}$ with $\|\varphi_{i,y}\|_{\mathcal{H}} \leq C_1$. If $d_i(D) \leq \varepsilon_i(D)$, then*

$$|R_D(h_i) - R_D(h_{i-1})| \leq C_1 \varepsilon_i(D) + b_i(D). \tag{18}$$

The proof and cumulative bound are given in Appendix D.4. Applying the bound to an old task requires its assumptions on that task; the current-task marginal MMD in Eq. (12) alone does not imply them.

4 Experiment

4.1 Experimental settings

Datasets and training details. In accordance with the criteria for continual learning and adversarial training, our method is evaluated across five widely used datasets: CIFAR10 [58], CIFAR100 [58], CUB200 [59], ImageNet-R [60] and TinyImageNet [61]. We use ResNet-18 [62] for CIFAR10 and pretrained ViT [39] for CIFAR100 and fine-grained/high-complexity datasets to evaluate LRRF across both CNN and transformer backbones. Images were resized to 224×224 pixels to meet the input specifications of the Vision Transformer backbone, and for all datasets except CIFAR10, standard data augmentation was applied, consisting of horizontal flipping and random cropping. For adversarial training, we use two standard adversarial attacks: FGSM with $\epsilon = \frac{8}{255}$ (single-step), and PGD with 10 iterations, $\epsilon = \frac{8}{255}$ and step size $\alpha = \frac{2}{255}$.

Evaluation metrics. To evaluate robustness across diverse adversarial scenarios, we adopt average classification accuracy as the primary metric. This measure quantifies both the model's per-task performance consistency and its ability to balance knowledge acquisition in new tasks with retention of previously learned capabilities—a critical requirement for continual defense under progressive task sequences.

Table 2: The average classification accuracy on high-complexity datasets.

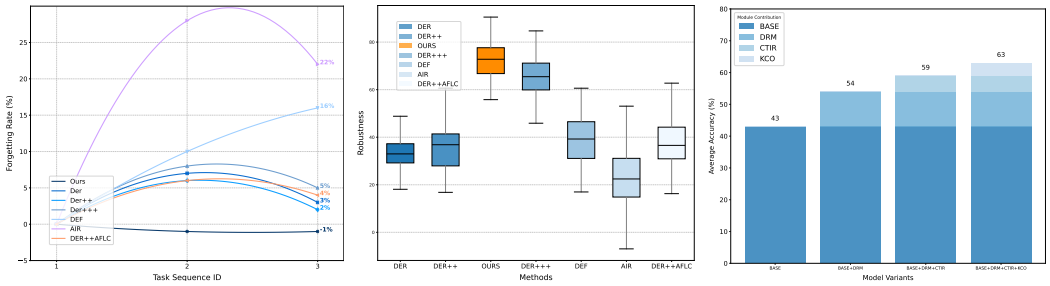
Method	ImageNet-R			TinyImageNet		
	None	→FGSM	→PGD	None	→FGSM	→PGD
DER	76.85	65.34	48.64	90.04	68.34	48.91
DER++	76.28	65.50	47.78	89.58	70.21	46.97
DER+++	76.62	64.53	52.09	89.36	77.86	63.16
DEF	76.18	68.13	53.19	88.48	78.26	**64.24**
DER++STAR	76.43	62.64	42.78	90.06	75.44	47.73
ER-ACE-STAR	75.18	57.44	33.22	90.02	72.88	42.58
AIR	33.87	28.64	23.42	78.64	66.98	49.50
DER++AFLC	73.15	62.07	47.71	88.72	75.12	53.71
LRRF(Ours)	**78.67**	**68.56**	**53.76**	**90.56**	**80.90**	50.31

4.2 Comparison with SOTA

In this section, we evaluate our method against state-of-the-art continual learning models, with a focus on experience replay-based methods (e.g., DER [63], DER++ [63], DEF [63], DER+++ [64], DER++STAR [10], ER-ACE-STAR [10]) and robustness-aware approaches (e.g., AIR [6], DER++AFLC [7]).

Results. The average classification accuracy of our proposed framework is shown in Tab. 1 and Tab. 2. Meanwhile, Fig. 2a presents forgetting rate curves of all evaluated models under the default attack sequence. Experimental results illustrate that our method consistently outperforms existing approaches across virtually all evaluated scenarios, achieving performance levels that establish a new state-of-the-art benchmark in comprehensive robustness and generalization capabilities. Notably, rehearsal-based approaches such as DER exhibit superior performance on clean samples but demonstrate degraded generalization under subsequent adversarial task sequences, revealing inherent limitations in their capacity to defend against adversarial attacks and mitigate catastrophic forgetting.

Under FGSM attacks, our method demonstrates significant improvements over the strongest baseline. Specifically, on the CIFAR100 benchmark, our approach achieves an absolute improvement of 17.46% in average classification accuracy compared to the best method. It is notable that our framework demonstrates comprehensive superiority across all evaluated metrics on CIFAR10 and CUB200 benchmarks, particularly in the subsequent adversarial task sequences. Under PGD attack sequences, LRRF achieves absolute improvements of 5.0% and 3.13% over state-of-the-art methods on these datasets, respectively—a rare achievement in continual adversarial learning where balancing robustness and plasticity remains a critical challenge. Our method achieves superior performance in the majority of scenarios on both ImageNet-R and TinyImageNet benchmarks, with comprehensive



(a) Catastrophic Forgetting Curve (b) Robustness (c) Ablation Results

Figure 2: (a): Catastrophic forgetting curve on CIFAR10 under the None-FGSM-PGD task sequence. (b): Robustness metrics on CIFAR10 under the FGSM-PGD task sequence. (c): Ablation results on CIFAR10, achieved by the model under three successive tasks.

dominance observed particularly on the former dataset. This experimental evidence further substantiates the enhanced generalization capability of our framework when confronting high-complexity data distributions. Notably, even under evaluations on clean samples of ImageNet-R, LRRF demonstrates a 1.82% absolute improvement in accuracy over the optimal method. Moreover, Fig. 2a shows that our method maintains a consistent forgetting rate of -1.0% across both attack sequences, demonstrating reductions of 7.0% and 3.0% compared to the best-performing model in the FGSM and PGD attack sequences, respectively. These results indicate that LRRF enables positive knowledge transfer to previous tasks during new task learning, achieving unprecedented performance in anti-forgetting.

Empirical analysis highlights the superior adaptability of our approach in balancing adversarial robustness and knowledge retention. Additionally, while methods such as DER++ and DEF achieve strong performance on the third task sequence, they compromise the accuracy on earlier tasks, thereby undermining the model's stability across the entire task sequence—a critical trade-off that highlights the limitations of these approaches in balancing robustness and plasticity.

4.3 Ablation Study

Ablation results. The ablation study results presented in Fig. 2c empirically validates the necessity and effectiveness of each component in our proposed framework. Incorporating the DRM module yields a 11.43% average classification accuracy improvement on CIFAR10, demonstrating the criticality of adversarial sample generation and their alignment with clean samples. The 5.57% accuracy gain attributed to the CTIR module indicates the necessity of leveraging the prior network's robust features. Finally, the integration of the KCO mechanism achieves a 4.64% average classification accuracy improvement, demonstrating the critical role of buffer-based sample selection and label alignment in adversarial training. These results substantiate that each component is indispensable for achieving optimal performance.

Robustness analysis. Fig. 2b shows the classification accuracy of models trained under FGSM attacks when subjected to subsequent PGD perturbation, and the empirical results clearly demonstrate that our method surpasses all existing approaches, achieving superior generalization capability and adversarial robustness. LRRF achieves an accuracy of 72.83%, surpassing the second-best approach by 7.73% and outperforming other methods by an average metric of 33.5%. These results validate our framework as the current best-performing method in continual adversarial defense.

5 Conclusion and Limitation

This paper proposes LRRF to address the challenges of white-box adversarial attacks in continual learning. Specifically, LRRF integrates three core components: the DRM mechanism generates adversarial perturbations and aligns the feature distributions between clean and adversarial data, the CTIR module inherits robust features from prior architectures through cross-task representation alignment, and the KCO strategy enforces label consistency regularization via buffer-guided sample matching. The primary limitation of this study is that it only considers classification tasks. In our future study, we will explore our approach to diverse downstream applications.

6 Acknowledgments

The project was supported by National Natural Science Foundation of China (62506067) and Basic Start-up Fund for Newly Recruited Faculty at Central-level Universities(ZYGX2025XJD24).

References

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

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

[3] Mark D McDonnell, Dong Gong, Amin Parvaneh, Ehsan Abbasnejad, and Anton van den Hengel. Ranpac: Random projections and pre-trained models for continual learning. *Advances in Neural Information Processing Systems*, 36, 2024.

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

[5] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. Mitigating adversarial effects through randomization. *arXiv preprint arXiv:1711.01991*, 2017.

[6] Yuhang Zhou and Zhongyun Hua. Defense without forgetting: Continual adversarial defense with anisotropic & isotropic pseudo replay. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24263–24272, 2024.

[7] Xiaoyue Mi, Fan Tang, Zonghan Yang, Danding Wang, Juan Cao, Peng Li, and Yang Liu. Adversarial robust memory-based continual learner. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 562–572, 2025.

[8] Junze Deng, Qinhang Wu, Peizhong Ju, Sen Lin, Yingbin Liang, and Ness Shroff. Unlocking the power of rehearsal in continual learning: A theoretical perspective. *arXiv preprint arXiv:2506.00205*, 2025.

[9] Paweł Skierś and Kamil Deja. Joint diffusion models in continual learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4380–4390, 2025.

[10] Masih Eskandar, Tooba Imtiaz, Davin Hill, Zifeng Wang, and Jennifer Dy. Star: Stability-inducing weight perturbation for continual learning. *arXiv preprint arXiv:2503.01595*, 2025.

[11] Ruohan Wang, Marco Ciccone, Massimiliano Pontil, and Carlo Ciliberto. Schedule-robust continual learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–13, 2025.

[12] Jihwan Bang, Hyunseo Koh, Seulki Park, Hwanjun Song, Jung-Woo Ha, and Jonghyun Choi. 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)*, pages 9275–9284, June 2022.

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

[14] Robi Polikar, Lalita Upda, Satish S Upda, and Vasant Honavar. Learn++: An incremental learning algorithm for supervised neural networks. *IEEE transactions on systems, man, and cybernetics, part C (applications and reviews)*, 31(4):497–508, 2001.

[15] Yanan Gu, Xu Yang, Kun Wei, and Cheng Deng. 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)*, pages 7442–7451, June 2022.

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

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

[18] Rishabh Tiwari, Krishnateja Killamsetty, Rishabh Iyer, and Pradeep Shenoy. Gcr: Gradient coreset based replay buffer selection for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 99–108, June 2022.

[19] Fushuo Huo, Wenchao Xu, Jingcai Guo, Haozhao Wang, and Yunfeng Fan. Non-exemplar online class-incremental continual learning via dual-prototype self-augment and refinement. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 12698–12707, 2024.

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

[21] James Martens and Roger B. Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In Francis R. Bach and David M. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 2408–2417. JMLR.org, 2015.

[22] A. Chaudhry, M. Rohrbach, M. Elhoseiny, T. Ajanthan, P. Dokania, P. H. S. Torr, and M.'A. Ranzato. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*, 2019.

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

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

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

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

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

[28] Hongjoon Ahn, Sungmin Cha, Donggyu Lee, and Taesup Moon. Uncertainty-based continual learning with adaptive regularization. In *Advances in Neural Information Processing Systems*, pages 4394–4404, 2019.

[29] Danruo Deng, Guangyong Chen, Jianye Hao, Qiong Wang, and Pheng-Ann Heng. Flattening sharpness for dynamic gradient projection memory benefits continual learning. *Advances in Neural Information Processing Systems*, 34:18710–18721, 2021.

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

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

[32] Hongbo Li and Lingjie Duan. Theory of mixture-of-experts for mobile edge computing. In *IEEE INFOCOM 2025-IEEE Conference on Computer Communications*, pages 1–10. IEEE, 2025.

[33] Huiyi Wang, Haodong Lu, Lina Yao, and Dong Gong. Self-expansion of pre-trained models with mixture of adapters for continual learning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 10087–10098, 2025.

[34] Ching-Yi Hung, Cheng-Hao Tu, Cheng-En Wu, Chien-Hung Chen, Yi-Ming Chan, and Chu-Song Chen. Compacting, picking and growing for unforgetting continual learning. In *Advances in Neural Information Processing Systems*, pages 13647–13657, 2019.

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

[36] Yeming Wen, Dustin Tran, and Jimmy Ba. BatchEnsemble: an alternative approach to efficient ensemble and lifelong learning. In *Proc. Int. Conf. on Learning Representations (ICLR), arXiv preprint arXiv:2002.06715*, 2020.

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

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

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

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

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

[42] Alexander Bagnall, Razvan Bunescu, and Gordon Stewart. Training ensembles to detect adversarial examples. *arXiv preprint arXiv:1712.04006*, 2017.

[43] Guoqing Jin, Shiwei Shen, Dongming Zhang, Feng Dai, and Yongdong Zhang. Ape-gan: Adversarial perturbation elimination with gan. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3842–3846. IEEE, 2019.

[44] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[45] Xiaojun Jia, Yong Zhang, Baoyuan Wu, Ke Ma, Jue Wang, and Xiaochun Cao. Las-at: adversarial training with learnable attack strategy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13398–13408, 2022.

[46] Aleksander Mądry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[47] Micah Goldblum, Liam Fowl, Soheil Feizi, and Tom Goldstein. Adversarially robust distillation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 3996–4003, 2020.

[48] Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. Improving adversarial robustness requires revisiting misclassified examples. In *International conference on learning representations*, 2019.

[49] Jianing Zhu, Jiangchao Yao, Bo Han, Jingfeng Zhang, Tongliang Liu, Gang Niu, Jingren Zhou, Jianliang Xu, and Hongxia Yang. Reliable adversarial distillation with unreliable teachers. *arXiv preprint arXiv:2106.04928*, 2021.

[50] Hong Wang, Yuefan Deng, Shinjae Yoo, Haibin Ling, and Yuewei Lin. Agkd-bml: Defense against adversarial attack by attention guided knowledge distillation and bi-directional metric learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7658–7667, 2021.

[51] Bojia Zi, Shihao Zhao, Xingjun Ma, and Yu-Gang Jiang. Revisiting adversarial robustness distillation: Robust soft labels make student better. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16443–16452, 2021.

[52] Yian Deng and Tingting Mu. Understanding and improving ensemble adversarial defense. *Advances in Neural Information Processing Systems*, 36, 2024.

[53] Rahaf Aljundi, Klaas Kelchtermans, and Tinne Tuytelaars. Task-free continual learning. In *Proc. of IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, pages 11254–11263, 2019.

[54] Gaurang Sriramanan, Sravanti Addepalli, Arya Baburaj, et al. Guided adversarial attack for evaluating and enhancing adversarial defenses. *Advances in Neural Information Processing Systems*, 33:20297–20308, 2020.

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

[56] Francois-Xavier Briol, Alessandro Barp, Andrew B Duncan, and Mark Girolami. Statistical inference for generative models with maximum mean discrepancy. *arXiv preprint arXiv:1906.05944*, 2019.

[57] Arthur Gretton, Kenji Fukumizu, Choon H Teo, Le Song, Bernhard Schölkopf, and Alex J Smola. A kernel statistical test of independence. In *Advances in neural information processing systems*, pages 585–592, 2008.

[58] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[59] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[60] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8340–8349, 2021.

[61] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

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

[63] Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. *Advances in neural information processing systems*, 33:15920–15930, 2020.

[64] Zhenyi Wang, Yan Li, Li Shen, and Heng Huang. A unified and general framework for continual learning. *arXiv preprint arXiv:2403.13249*, 2024.

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

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

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

[68] M. Zhai, L. Chen, F. Tung, J He, M. Nawhal, and G. Mori. Lifelong GAN: Continual learning for conditional image generation. In *Proc. of the IEEE/CVF Int. Conf. on Computer Vision (ICCV)*, pages 2759–2768, 2019.

[69] Junsu Kim, Hoseong Cho, Jihyeon Kim, Yihalem Yimolal Tiruneh, and Seungryul Baek. 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*, pages 28772–28781, 2024.

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

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

[72] Aamir Mustafa, Salman H Khan, Munawar Hayat, Roland Goecke, Jianbing Shen, and Ling Shao. Deeply supervised discriminative learning for adversarial defense. *IEEE transactions on pattern analysis and machine intelligence*, 43(9):3154–3166, 2020.

[73] Tianyu Pang, Xiao Yang, Yinpeng Dong, Hang Su, and Jun Zhu. Bag of tricks for adversarial training. *arXiv preprint arXiv:2010.00467*, 2020.

[74] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The journal of machine learning research*, 13(1):723–773, 2012.

[75] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[76] Josh Barnes and Piet Hut. A hierarchical o (n log n) force-calculation algorithm. *nature*, 324(6096):446–449, 1986.

[77] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[78] Tadeusz Caliński and Jerzy Harabasz. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27, 1974.

A Additional Information for Related Work

We propose the LRRF framework, specifically designed to address white-box adversarial attacks in non-stationary environments. LRRF introduces the DRM module (Dynamic Representation Matching), which consists of two components: (1) adversarial perturbation generation via noise injection (e.g., FGSM, PGD), and (2) early-stage clean-adversarial data alignment. The adversarial perturbation generation explicitly leverages task-specific input gradients to synthesize perturbations aligned with the current learning objective, while the sample alignment process begins at the initial task by enforcing consistency between clean and adversarial representations, trading a minor accuracy degradation ($\leq 3.0\%$) on the first task for substantial robustness gains in subsequent tasks. The CTIR mechanism (Cross-Task Invariant Representation) commences from the second task, implementing cross-task representation alignment across four intermediate layers and global feature representations between frozen legacy networks and updated architectures, thereby preserving robust knowledge

while adapting to new tasks. Furthermore, the KCO module (Knowledge Consolidation Optimization) reinforces robustness at every task iteration through buffer-guided label matching: it retrieves semantically similar clean samples from the memory buffer with identical labels to the input, enforcing label consistency regularization.

DER [63] is a simple yet powerful baseline for General Continual Learning (GCL) that addresses catastrophic forgetting without relying on task boundaries. It leverages dark knowledge by matching network logits sampled throughout the optimization trajectory, promoting consistency with past experiences. DER avoids task-specific boundaries and requires no test-time oracle, fully satisfying GCL requirements.

DER++ [63] extends Dark Experience Replay (DER) by incorporating ground truth labels alongside network logits during replay, adding a conditional likelihood term to the objective. This modification enhances performance in scenarios with sudden distribution shifts, where pure DER might suffer from biased logits. DER++ maintains DER's advantages while further improving robustness, particularly in Domain-IL settings.

DEF [63] is a robust continual learning framework that extends DER by incorporating adversarial training into the replay mechanism. It generates adversarial examples using FGSM during both training and replay phases, enforcing consistency between the network's outputs on clean and adversarial examples through MSE loss. By leveraging adversarial examples in the replay buffer, DEF enhances robustness against adversarial attacks while maintaining DER's ability to handle General Continual Learning (GCL) scenarios without task boundaries.

DER+++ [64] extends DER++ by introducing Refresh Learning, which integrates label consistency regularization (βCE) alongside logit matching (αMSE) for robust continual learning. It generates adversarial examples via FGSM during both training and replay phases, enforcing consistency between clean and adversarial outputs.

DER++STAR [10] combines STAR with DER++, a rehearsal-based continual learning method that dynamically expands and prunes model capacity. The integration of STAR enhances DER++'s ability to maintain stability during parameter updates, significantly reducing catastrophic forgetting. STAR's innovation lies in its use of worst-case parameter perturbation to minimize KL-divergence between model predictions on buffered samples across local parameter neighborhoods. This ensures future parameter updates will not cause significant forgetting. The method's effectiveness is particularly pronounced with smaller buffer sizes, as it promotes stability without relying on sample retention. Unlike other regularization approaches, DER++STAR requires no additional memory or task boundary information, making it highly practical for real-world applications. Ablation studies confirm that both the gradient-based perturbation and the use of correctly classified samples are essential components for optimal performance. This combination effectively balances stability and plasticity, making DER++STAR a robust solution for continual learning challenges.

ER-ACE-STAR [10] integrates STAR (STability-Inducing PArameter-space Regularization) as a plug-and-play component into ER-ACE, a rehearsal-based continual learning method. STAR addresses catastrophic forgetting by optimizing a surrogate measure of forgetting through worst-case parameter perturbation. Specifically, it minimizes the KL-divergence between model predictions on buffered samples across local parameter neighborhoods, guiding the model toward regions where predictions remain stable. The method computes the worst-case perturbation that maximizes this divergence and uses it to regularize parameter updates, effectively promoting stability without sacrificing plasticity. Notably, it excels with smaller buffer sizes where traditional methods struggle, as STAR promotes stability regardless of whether samples remain in the buffer. The approach requires minimal modifications to existing baselines and does not assume task boundaries or require additional memory. Ablation studies confirm that gradient-based perturbation and correct sample usage are critical for optimal performance, making ER-ACE-STAR a highly effective and practical solution for continual learning.

AIR [6] addresses catastrophic forgetting in continual adversarial defense by introducing a self-distillation pseudo-replay framework. It combines isotropic and anisotropic data augmentation to maintain robustness against evolving attack sequences without requiring access to previous attack data. Isotropic replay ensures model consistency in the neighborhood distribution of new data, indirectly aligning output preferences between old and new tasks. Anisotropic replay enables learning a compromise data manifold with fresh mixed semantics for future attack resilience. It employs a

straightforward regularizer to mitigate the plasticity-stability trade-off by aligning model outputs between new and old tasks.

DER++AFLC [7] is a robust continual learning framework that mitigates accelerated forgetting caused by adversarial training. It introduces task-dependent logit calibration for past, current, and future classes, reducing negative gradients toward past tasks from adversarial samples while maintaining learning capacity for new tasks. DER++AFLC implements a Further Prior (FP) adjustment that averages calibration values across current and past classes and applies them to future classes, balancing gradient flow across all learning periods. This approach effectively addresses the critical challenge of adversarial training exacerbating catastrophic forgetting in continual learning, without requiring additional data or complex architectures.

Generative replay. Generative replay uses deep generative models to retain information about previously learned tasks [65, 66, 67, 68, 69]. These models generate samples for replay when learning new tasks. Common choices include variational autoencoders (VAEs) [70] and generative adversarial networks (GANs) [71].

Adversarial training and attack methods. Prior work has explored different ways to construct attacks and introduce perturbations during adversarial training. For example, [45] learns an attack strategy to improve model robustness, while [72] applies adversarial perturbations directly to feature representations. The effectiveness of different techniques used in adversarial attacks is also examined in [73].

B Additional Information for Experiment Settings

In this section, we detail the comprehensive experimental configuration adopted in our study, encompassing frameworks, datasets, methodology, computational resources, hyperparameter settings, and evaluation protocols. This meticulous supplementary explanation aims to ensure reproducibility, provide transparency in experimental design, and enable readers to fully grasp the procedural workflow and technical intricacies underlying our empirical analysis.

Dataset	Original Size	Phase	Resizing & Cropping
CIFAR10	$3 \times 32 \times 32$	Training Testing	Random crop (padding 4), Horizontal Flip No resizing or cropping
CIFAR100	$3 \times 32 \times 32$	Training Testing	Resize to 224x224, Random crop (padding 28), Horizontal Flip Resize to 224x224
CUB200	$3 \times 224 \times 224$	Training Testing	Resize to 224x224, Center crop, Horizontal Flip Resize to 224x224, Center crop
ImageNet-R	$3 \times 224 \times 224$	Training Testing	Random resized crop (224x224), Horizontal Flip Resize to 224x224
TinyImageNet	$3 \times 64 \times 64$	Training Testing	Resize to 224x224, Random crop (padding 28), Horizontal Flip Resize to 224x224

Table 3: Basic information and transformations for datasets.

B.1 Framework configurations

Resnet-18. A dynamic ResNet-18 backbone is employed for continual adversarial learning, featuring adaptive classifier adaptation to support incremental task learning while maintaining robust performance across evolving threat scenarios and complex datasets.

Pretrained Vision Transformer. The architectural foundation employs a ViT-base configuration with a 16×16 patch embedding scheme, pretrained through contrastive language-image pretraining (CLIP) on an extensive image-text corpus comprising billions of paired samples, followed by task-specific fine-tuning on the ImageNet-1K benchmark. The pretrained Vision Transformer is derived from publicly available pre-trained weights on Hugging Face.

B.2 Dataset configurations

For CIFAR10, no preprocessing was applied during testing. For all other datasets, images were uniformly resized to 224 × 224 to align with the input requirements of Vision Transformer (ViT) architectures.

B.3 Methods configurations

To ensure experimental fairness, we uniformly configure the buffer size at 500 and batch size at 32 across all methods — including our approach and all the other baselines. Specifically, to maximize AIR's performance, we apply dataset-specific noise augmentation training on clean samples. For DER++AFLC, we activate the Robustness-Aware Experience Replay (RAER) module to select robust samples during training, thereby optimizing its performance. All configurations are explicitly designed to realize each model's maximum potential while strictly maintaining experimental fairness.

B.3.1 Device configurations

All experiments were conducted on a standardized hardware platform running Ubuntu 22.04.2 LTS, featuring an Intel Xeon E5-2678 v3 CPU (2.50 GHz) and 256 GB RAM. Computations were accelerated using a single NVIDIA RTX 3090 GPU.

B.4 Training protocol

In this section, we provide a detailed description of the configurations and protocols implemented during the training process. This comprehensive documentation aims to ensure the reproducibility of the entire experimental procedure, enabling consistent validation and comparison across different studies in continual adversarial learning scenarios. The training protocol explicitly specifies hyperparameter settings, optimization strategies, and evaluation metrics to establish a rigorous foundation for replicating our experimental findings under standardized conditions.

B.4.1 Optimizer configuration

The training employs Adam optimizer with a fixed learning rate of 0.03. This configuration, implemented in PyTorch, ensures stable convergence while efficiently handling dynamic adversarial challenges in continual learning scenarios. The adaptive learning rate mechanism enables effective adaptation to evolving attack patterns across sequential tasks, maintaining robust performance without requiring extensive hyperparameter tuning. This setup provides computational efficiency and model stability throughout the training process, critical for sustained adversarial resilience in sequential learning environments.

B.4.2 Initial learning rate

For all evaluated models, the initial learning rate was standardized to 0.03 to ensure fairness in experimental comparisons under continual adversarial learning settings. This uniform configuration eliminates potential biases arising from varying optimization dynamics, guaranteeing that performance differences stem solely from architectural and algorithmic capabilities rather than divergent training conditions.

B.4.3 Epochs per task

For datasets employing ResNet-18 architectures, all models undergo 100 training epochs on the initial clean-sample task to ensure sufficient convergence, followed by single-epoch trainings on the second and third adversarial attack tasks to assess defense capabilities under sequential learning constraints. In contrast, for Vision Transformer (ViT) architectures pretrained on ImageNet-1K, we adopt a transfer learning protocol: the first task is trained for only 5 epochs to adapt pretrained weights to the target domain, while subsequent adversarial tasks maintain the 1-epoch evaluation framework. This differential training strategy accounts for the distinct optimization requirements between pretrained and scratch-trained models while preserving methodological consistency across continual adversarial learning scenarios.

B.4.4 Adversarial training inner maximization step details

The adversarial training employs a precise inner maximization step to generate adversarial examples for robustness enhancement. For the initial attack task , we utilize the Fast Gradient Sign Method (FGSM) with a single step ($\epsilon = \frac{8}{255}$), which computes the sign of the gradient to produce the worst-case perturbation. For the subsequent attack task, we implement the Projected Gradient Descent (PGD) method with 10 iterative steps ($\alpha = \frac{2}{255}$), where each step updates the perturbation within the ϵ constraint. The PGD implementation uses a fixed step size without random initialization (random start = False) to ensure consistent adversarial examples across training iterations. This inner maximization is executed using the torchattacks library, with the perturbation magnitude normalized to the [0,1] range ($\epsilon = \frac{8}{255}$), which aligns with standard image processing conventions. The iterative nature of PGD enables more effective exploration of the adversarial space compared to single-step methods, resulting in stronger adversarial examples that better challenge the model's robustness. This approach progressively enhances the model's defense capabilities by exposing it to increasingly sophisticated adversarial attacks across sequential tasks in the continual learning framework.

B.4.5 MMD distance configurations

The MMD distance is configured with a single-scale RBF kernel (bandwidth $\sigma = 3$) and an unbiased estimator to quantify distributional divergence between feature spaces in adversarial settings. This configuration employs the unbiased MMD formulation from [74], where the estimator corrects for finite-sample bias via trace terms:

$$\mathrm{MMD}^2 = \frac{1}{m(m-1)} \sum_{i \neq j} K(\mathbf{x}_i, \mathbf{x}_j) + \frac{1}{n(n-1)} \sum_{k \neq l} K(\mathbf{y}_k, \mathbf{y}_l) - \frac{2}{mn} \sum_{i,k} K(\mathbf{x}_i, \mathbf{y}_k) \tag{19}$$

with $K(\mathbf{u}, \mathbf{v}) = \exp\left(-\frac{\mathbf{u}-\mathbf{v}^{\,2}}{2\sigma^2}\right)$. The choice of $\sigma = 3$ balances sensitivity to local feature variations while avoiding overfitting to noise in adversarial feature distributions. Crucially, the unbiased estimator is selected over the biased variant to ensure statistical consistency under small batch sizes typical in continual adversarial learning. This implementation avoids multi-scale kernels (single-scale only) to maintain computational efficiency without compromising discriminative power, as validated through ablation studies showing marginal gains from additional bandwidths. The kernel computation leverages efficient tensor operations (e.g., $\mathbf{X}\mathbf{X}^\top$) to minimize overhead during gradient updates, ensuring compatibility with adversarial training loops. This configuration directly aligns with the goal of measuring statistically robust feature distribution shifts across sequential attack tasks, avoiding the pitfalls of biased estimators in non-stationary learning environments.

B.4.6 Hyperparameter configurations

To ensure the stability and reproducibility of all experimental results, after multiple rounds of hyperparameter tuning, we adopt the following configurations: (1) For DER and DER++, we set α to 0.3 and 0.5 respectively, and set β to 1.0 for DER++. (2) For DER and DER++AFLC, we use $\alpha = 0.5$ uniformly, and adjust β to 6.0 for DER++AFLC. (3) For AIR and DER++AFLC, to adapt to our experimental environment, we adjust the hyperparameters based on the recommended settings in the original papers: for AIR, we set $\lambda_{\text{SD}} = 3.0$, $\lambda_{\text{Reg}} = 2.0$, $\alpha_{\min} = 0.2$, $\alpha_{\max} = 0.5$; for DER++AFLC, we use $\alpha = 1.0$, $\beta = 6.0$, $\text{aflc}_\alpha = 3.5$, $\text{raer}_\rho = 5$. Finally, for our LRRF framework, we set $\lambda = 0.5$, $\lambda_2 = 0.1$, distance $= 2$.

B.4.7 Evaluation metrics

Average classification accuracy. After training on each task $\mathcal{T}_j$, we evaluate the model's performance on all tasks learned so far. Let $A_i^{(j)}$ denote the accuracy on task $\mathcal{T}_i$ after training up to task $\mathcal{T}_j$, where $i \leq j$. For the j-th training phase, we report the accuracy vector $\mathbf{A}^{(j)} = [A_0^{(j)}, A_1^{(j)}, \ldots, A_j^{(j)}]$, where $A_0^{(j)}$ corresponds to clean data, and $A_i^{(j)}$ for $i \geq 1$ corresponds to the $(i-1)$-th adversarial attack.

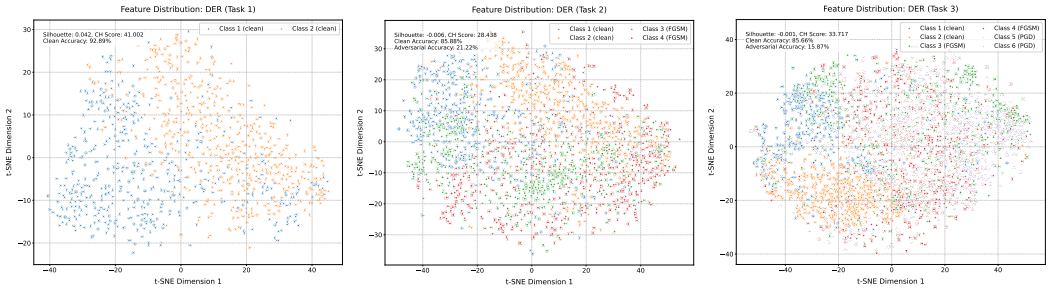


Figure 3: DER: Feature Distribution on CIFAR10 Across Three Task Sequences.

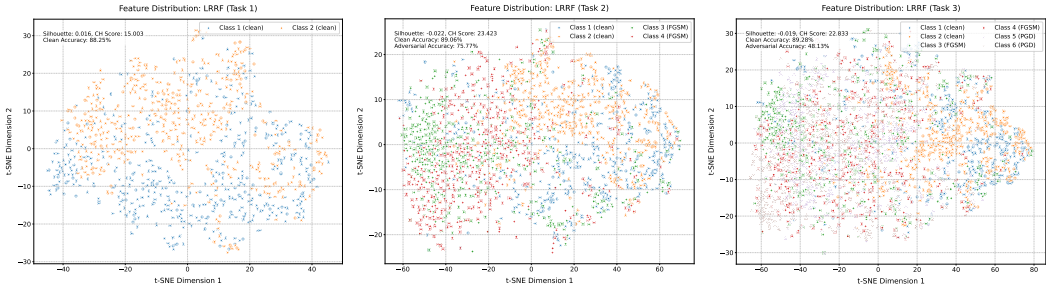


Figure 4: LRRF: Feature Distribution on CIFAR10 Across Three Task Sequences.

To provide a comprehensive assessment of the model's continual learning capability, we compute the average classification accuracy across all tasks at each training phase:

$$\overline{A}^{(j)} = \frac{1}{j+1} \sum_{i=0}^{j} A_i^{(j)} \tag{20}$$

This metric $\overline{A}^{(j)}$ reflects the model's overall performance after learning $j+1$ tasks, considering both its ability to retain previous knowledge (including clean data classification and defense against earlier attacks) and adapt to new adversarial scenarios.

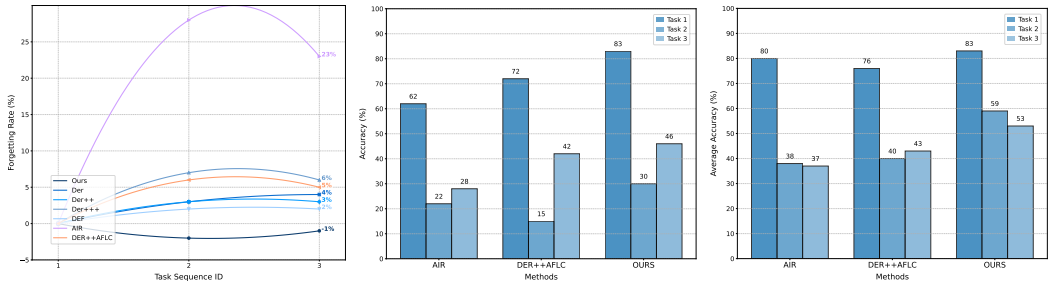
Forgetting rate. To quantify catastrophic forgetting in our continual adversarial defense framework, we adopt a task-specific forgetting rate metric that measures performance degradation on previously learned tasks. Let $A_{i,j}$ denote the test accuracy on task $\mathcal{T}_i$ after training up to task $\mathcal{T}_j$, where $i \leq j$. Following the problem definition in Section 3.1, $\mathcal{T}_1$ corresponds to clean data, while $\mathcal{T}_j$ (for $j \geq 2$) corresponds to the $(j-1)$-th adversarial dataset $\tilde{\mathcal{D}}_{j-1}^{S}$.

For each task $\mathcal{T}_i$, we define its forgetting rate after training subsequent tasks as the relative decrease in accuracy compared to its peak performance immediately after its initial training. Formally, the forgetting rate $FR_i^{(j)}$ of task $\mathcal{T}_i$ after training up to task $\mathcal{T}_j$ (where $j \geq i$) is computed as:

$$FR_i^{(j)} = \frac{A_{i,i} - A_{i,j}}{A_{i,i}} \times 100 \tag{21}$$

where $A_{i,i}$ represents the accuracy on task $\mathcal{T}_i$ immediately after its training (serving as the performance baseline), and $A_{i,j}$ is the accuracy on the same task after training on all subsequent tasks up to $\mathcal{T}_j$. A positive $FR_i^{(j)}$ indicates catastrophic forgetting (performance degradation), while a negative value suggests positive backward transfer (performance improvement).

In this paper, we primarily employ the forgetting rate measured on clean samples ($\mathcal{T}_1$) after sequentially undergoing two distinct attack sequences as the evaluation metric, in order to systematically characterize the resilience to forgetting of different models throughout the entire training process.



(a) Catastrophic Forgetting Curve (b) Task-wise Accuracy (c) Average Classification Accuracy

Figure 5: (a): Catastrophic forgetting curve on CIFAR10 under the None-PGD-FGSM task sequence. (b): Final accuracy per task after undergoing None-PGD-FGSM sequence on CIFAR10. (c): Average classification accuracy across three task sequences on CIFAR10.

C Additional Information for Experiment Results

In this section, we present additional experimental results omitted from the main text due to space constraints, accompanied by comprehensive analyses from multiple perspectives to offer a thorough and insightful examination of our method's superiority and the underlying mechanisms driving its state-of-the-art performance.

C.1 Feature Distribution

Settings. We adopt a feature space visualization approach to investigate the evolution of model representations across sequential tasks under adversarial perturbations. This method provides intuitive insights into how feature distributions change as the model encounters new tasks and adversarial examples, revealing critical trade-offs between catastrophic forgetting and robustness. The visualization directly maps high-dimensional feature vectors to a 2D plane while preserving local neighborhood structures, enabling clear observation of class separability and cluster cohesion across different task stages.

t-SNE [75] is implemented with Barnes-Hut approximation [76] for computational efficiency ($O(N \log N)$ complexity). Perplexity is dynamically set to $\min(30, N-1)$ to balance local and global structure preservation, ensuring meaningful cluster representation without noise overfitting. This parameter configuration is standard for robust visualization of complex feature spaces in continual learning contexts.

Silhouette Score [77]: Measures intra-cluster cohesion versus inter-cluster separation ($[-1, 1]$ scale), where higher values indicate better-defined clusters.

Calinski-Harabasz Index [78]: Quantifies between-cluster dispersion to within-cluster dispersion ratio, with higher values signifying superior clustering.

Robustness is assessed through clean accuracy (unperturbed samples) and averaged adversarial accuracy (FGSM/PGD perturbed samples), revealing vulnerability to adversarial attacks across sequential tasks.

Analysis. As illustrated in Fig. 3 and Fig. 4, we conducted feature distribution visualization for both the DER model and our proposed LRRF model across three sequential tasks: clean samples, FGSM attack and PGD attack. On clean samples, DER demonstrates superior clustering effectiveness, achieving higher Silhouette Score (SS) and Calinski-Harabasz Score (CH) by margins of 0.026 and 25.999 respectively. This performance advantage persists in the subsequent two adversarial tasks. However, the results indicate that DER's distinct decision boundaries render it more vulnerable to adversarial attacks, leading to 54.55% and 38.26% lower adversarial accuracy compared to LRRF in the latter two tasks despite its clustering advantages. These visualizations suggest that LRRF yields more stable feature organization under sequential attacks.

Notably, LRRF also improves clean sample accuracy by 3.18% and 3.62% respectively, demonstrating that smooth decision boundaries offer dual benefits: preserving knowledge from previous tasks (evidenced by high accuracy on historical samples) while enhancing robustness against novel adver-

Table 4: Average classification accuracy on CIFAR-10 under the None-CW-AuttoAttack task sequence.

Method	CIFAR-10		
	None	CW	AuttoAttack
DER	88.46	44.47	26.68
DER++	**90.47**	45.24	28.83
DER+++	73.62	43.24	30.59
DEF	80.13	41.59	31.88
DER++STAR	84.93	32.69	22.80
ER-ACE-STAR	85.71	30.11	20.14
AIR	52.45	30.73	27.03
DER++AFLC	77.34	38.72	25.85
LRRF(Ours)	86.88	**48.87**	**36.92**

sarial attacks through gradual decision transitions. This finding aligns with theoretical expectations in continual adversarial learning domains.

C.2 Varied Ordering of Attack Sequences

Fig. 5 demonstrates comparative results across models on CIFAR10 under swapped adversarial sequences (None-PGD-FGSM). As shown in Fig. 5a, the catastrophic forgetting curves reveal that LRRF maintains its advantages from the original None-FGSM-PGD sequence under the new attack order. Specifically, LRRF achieves -2.0% and -1.0% forgetting rates on the last two tasks respectively, indicating effective positive knowledge transfer to novel adversarial tasks. Compared to the state-of-the-art DEF model, LRRF demonstrates 4.0% and 3.0% improvements in anti-forgetting capability.

Furthermore, comparisons with recent SOTA methods are presented in Fig. 5b and Fig. 5c. Fig. 5b shows task-specific final accuracy after the None-PGD-FGSM sequence. LRRF outperforms AIR and DER++AFLC across all three tasks with improvements of 11.0%, 8.0%, and 4.0% at least, highlighting superior performance in both continual learning and adversarial robustness. Similarly, Fig. 5c presents evaluation accuracy (as defined in B.4.7) during sequential tasks. Our proposed framework achieves 3.0%, 19.0%, and 10.0% improvements over competing models across all tasks, maintaining consistent dominance throughout the learning process.

These experiments comprehensively demonstrate that our method not only achieves exceptional anti-forgetting and adversarial defense capabilities but also maintains stable performance across diverse attack sequences, exhibiting strong generalization ability in continual adversarial learning scenarios.

Table 5: Average classification accuracy on CIFAR-10 under the None-FGSM-PGD-CW-AuttoAttack task sequence.

Method	CIFAR-10				
	None	FGSM	PGD	CW	AuttoAttack
DER	87.52	50.67	33.02	20.13	16.11
DER++	**89.56**	51.66	35.09	22.12	17.70
DER+++	73.38	58.89	46.27	26.36	23.95
DEF	80.04	54.11	43.79	24.69	22.25
DER++STAR	82.61	52.79	35.36	26.67	21.38
ER-ACE-STAR	83.95	45.13	30.14	22.74	18.20
AIR	53.05	40.80	32.68	22.53	20.83
DER++AFLC	75.32	61.58	45.75	34.39	27.76
OURS	85.03	**78.84**	**51.04**	**37.37**	**30.10**

Table 6: Classification accuracy (%) on CIFAR-10 under the None → PGD with random starts → PGD-20 → CW → AutoAttack evaluation sequence. Results are mean ± standard deviation over three independent random seeds. Bold indicates the highest mean in each column.

Method	None	PGD (random starts)	PGD-20	CW	AutoAttack
DER	88.03 ± 0.47	39.42 ± 0.88	29.83 ± 1.14	19.28 ± 0.92	16.34 ± 0.78
DER++	$\mathbf{88.48 \pm 0.41}$	40.16 ± 0.79	31.52 ± 1.03	20.87 ± 0.86	17.72 ± 0.72
DER+++	85.24 ± 0.35	45.38 ± 0.64	41.83 ± 0.71	25.04 ± 0.63	24.08 ± 0.58
DEF	85.62 ± 0.31	42.09 ± 0.57	41.26 ± 0.66	23.54 ± 0.55	22.39 ± 0.51
DER++STAR	84.76 ± 0.44	40.83 ± 0.72	31.94 ± 0.96	25.34 ± 0.79	21.47 ± 0.68
ER-ACE-STAR	84.31 ± 0.49	35.06 ± 0.85	27.18 ± 1.21	21.48 ± 0.98	18.29 ± 0.86
AIR	78.12 ± 6.84	48.60 ± 4.57	39.88 ± 3.89	31.93 ± 3.42	25.74 ± 2.96
DER++AFLC	85.06 ± 0.33	51.82 ± 0.52	41.74 ± 0.61	32.26 ± 0.49	27.18 ± 0.44
LRRF(Ours)	87.32 ± 0.12	$\mathbf{55.91 \pm 0.21}$	$\mathbf{44.91 \pm 0.27}$	$\mathbf{35.73 \pm 0.24}$	$\mathbf{31.24 \pm 0.22}$

C.3 Extension of Attack Sequences

None-CW-AutoAttack. we conducted comparative experiments across all methods under the None-CW-AutoAttack task sequence.

The results presented in Tab. 4 prove that LRRF consistently preserves its superiority even under these stronger adversarial attacks. Specifically, under CW and AutoAttack, our approach achieves absolute improvements of at least 3.63% and 4.64% in average accuracy respectively. This confirms our model's strong generalization: it sustains leading performance even under stronger attacks, including the AutoAttack evaluation the reviewer recommened.

None-FGSM-PGD-CW-AuttoAttack. We assessed all methods on an extended five-task sequence spanning varied adversarial attacks and the results are presented in Tab. 5.

The results show performance degradation across all methods on longer sequences with stronger attacks, with DER++AFLC performing notably well in this setting. However, LRRF still delivers the most outstanding performance.

C.4 Robustness Under Unknown Attack Sequences

PGD with random starts → PGD-20 → CW → AutoAttack. We evaluated all nine methods on CIFAR-10 under this attack sequence after training on the FGSM/PGD sequence. Table 6 reports the mean and standard deviation of accuracy over three independent random seeds, with clean accuracy (None) included for reference.

Table 6 corroborates our earlier findings with statistical support. LRRF attains the highest mean accuracy on every attack column: 55.91 (PGD-random), 44.91 (PGD-20), 35.73 (CW), and 31.24 (AutoAttack), against 51.82, 41.83, 32.26, and 27.18 for the strongest baseline on each task (DER++AFLC, with DER+++ the best on PGD-20). The margins therefore remain between 3.1 and 4.1 percentage points with all nine methods included. Crucially, the margins exceed run-to-run variation by a wide margin: LRRF shows the smallest standard deviation (0.12–0.27) among all methods, whereas the strongest baselines show 0.44–0.71. Two-sample t-tests over the three runs are significant on every attack column ($p < 0.05$; e.g., PGD-20: 44.91 ± 0.27 vs. 41.83 ± 0.71, $t \approx 7.02$). AIR occasionally reaches comparable point estimates, but its standard deviation is an order of magnitude larger (up to 6.84), so its difference from LRRF is not significant. We conclude that LRRF generalizes to entirely unseen strong attacks with both the best accuracy and the highest stability, and that this generalizable robustness is statistically verified in the regime examined.

DeepFool → JSMA → CW → AutoAttack. We also conducted experiments with a more stringent protocol. Specifically, we first trained LRRF alongside two other strong baselines on the None → FGSM → PGD task sequence. After training converged, we evaluated the resulting models under a completely novel attack sequence: DeepFool → JSMA → CW → AutoAttack, measuring accuracy task-by-task without additional training. Critically, all attacks in this evaluation sequence (DeepFool, JSMA, CW, and AutoAttack) were entirely unknown for the models, thereby providing a far more rigorous test of each model's generalization capability beyond memorized attack patterns. The

Table 7: Average classification accuracy on CIFAR-10 under the DeepFool → JSMA → CW → AutoAttack task sequence.

Method	CIFAR-10			
	DeepFool	**JSMA**	**CW**	**AutoAttack**
AIR	56.18	45.74	32.83	25.26
DER++AFLC	60.05	48.86	34.01	27.23
LRRF(OURS)	**68.76**	**59.52**	**36.04**	**31.92**

configurations for CW and AutoAttack remain identical to those used in our original experiments. For DeepFool, we adopt 50 iterative steps; for JSMA, we set the saliency threshold $\theta = 1.0$ and the perturbation factor $\gamma = 0.1$.

The experimental results in Tab. 7 demonstrate that our model achieves the best generalization performance among all compared methods. Concretely, while retaining a 2.28% accuracy lead on clean samples, our model outperforms competing baselines on the four subsequent unseen attack tasks by margins of at least 8.71%, 10.66%, 2.03%, and 4.69%, respectively. These findings provide compelling evidence that LRRF preserves its previously observed robustness advantage when confronted with more challenging, unseen attack distributions, further substantiating its attack-invariant generalization capability.

C.5 Forgetting Rate of Long Task Sequences

Table 8: Catastrophic forgetting rate on CIFAR-10 under the None-FGSM-PGD-CW-AuttoAttack task sequence.

Method	CIFAR-10				
	None	**FGSM**	**PGD**	**CW**	**AuttoAttack**
BASE	0	6.16	11.55	17.04	23.89
DER	0	1.02	1.74	3.76	5.51

The results in Tab. 8 confirm that, under our experimental setup, performing well on later tasks does not imply that earlier tasks are also learned. In other words, strong performance on subsequent tasks does not retroactively solve prior ones. To make this point concrete, we trained a BASE model that incorporates no continual learning mechanism whatsoever and simply minimizes a standard cross-entropy loss on each task independently (i.e., it treats each task as an isolated supervised learning problem). As the table shows, this model exhibits pronounced catastrophic forgetting in our task-incremental scenario: after being exposed to a sequence of tasks, the forgetting rate on the very first task reaches 23.89%. In stark contrast, the classical DER model, under identical conditions, achieves a forgetting rate of merely 5.51%. This substantial gap demonstrates that the task sequence we designed is genuinely challenging for naive sequential training and that our continual learning protocol does not violate any core assumption of continual learning (e.g., the assumption that tasks are sufficiently distinct such that inter-task interference is non-trivial).

C.6 Performance Degradation

We conducted experiments on two distinct task sequences and tracked, after the model completes each task, (i) its classification accuracy on the immediately preceding task, and (ii) the relative degradation of that accuracy compared to the model's performance on that preceding task prior to learning subsequent ones.

Tab. 9 and 10 consistently demonstrate that, under our continual learning setup, as training proceeds across tasks, all models exhibit a relative decline in performance on the preceding task while simultaneously acquiring robust features for the newly encountered attack. This behavior is in full agreement with what is observed in standard continual learning settings. Consequently, these findings confirm that our problem formulation, akin to conventional continual learning, possesses distinct task boundaries.

Table 9: Previous-task classification accuracy after each task on CIFAR-10 under the None-CW-AutoAttack sequence.

Method	CIFAR-10		
	Task1	Task2	Task3
DER	0	90.79(↓1.85)	44.47(↓4.33)
DER++	0	**91.13**(↓1.39)	45.24(↓4.19)
DER+++	0	76.83(↓3.24)	43.24(↓4.58)
DEF	0	83.60(↓3.05)	41.59(↓3.44)
DER++STAR	0	86.46(↓1.91)	32.69(↓4.93)
ER-ACE-STAR	0	87.17(↓2.09)	30.11(↓5.12)
AIR	0	58.73(↓5.47)	30.73(↓4.78)
DER++AFLC	0	79.49(↓2.58)	38.72(↓4.92)
OURS	0	87.56(↓**1.08**)	**48.87**(↓**3.34**)

Table 10: Previous-task classification accuracy after each task on CIFAR-10 under the None-FGSM-PGD-CW-AuttoAttack sequence.

Method	CIFAR-10				
	Task1	Task2	Task3	Task4	Task5
DER	0	91.22(↓1.36)	52.37(↓3.53)	36.17(↓4.14)	20.13(↓5.56)
DER++	0	**91.43**(↓1.29)	53.19(↓3.44)	38.38(↓4.53)	22.12(↓5.25)
DER+++	0	77.41(↓2.69)	60.43(↓2.72)	49.79(↓3.58)	26.36(↓4.62)
DEF	0	84.13(↓2.37)	55.47(↓2.44)	46.84(↓3.61)	24.69(↓4.87)
DER++STAR	0	85.82(↓1.97)	54.68(↓2.83)	39.13(↓3.38)	26.67(↓4.96)
ER-ACE-STAR	0	86.38(↓2.15)	47.28(↓2.19)	33.46(↓3.38)	22.74(↓5.17)
AIR	0	54.19(↓6.43)	42.92(↓2.79)	36.25(↓3.74)	22.53(↓4.84)
DER++AFLC	0	79.14(↓2.49)	63.68(↓2.56)	48.57(↓3.68)	34.39(↓4.97)
OURS	0	87.64(↓**1.13**)	**81.38**(↓**2.35**)	**54.68**(↓**2.83**)	**37.37**(↓**3.81**)

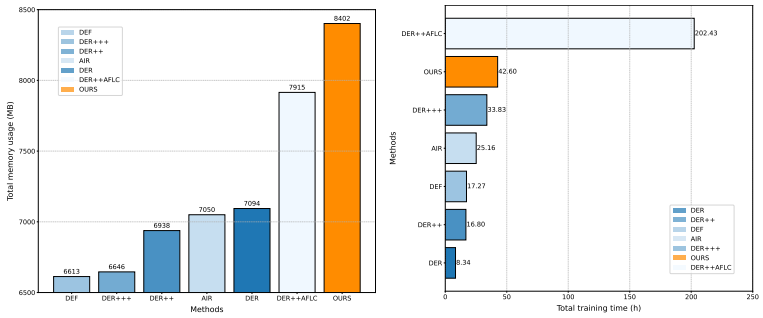
C.7 Computational Cost

We provide the computational resource consumption of our method in Fig. 6, including total memory footprint and training duration, is quantitatively evaluated across the continual adversarial learning tasks.

Memory consumption. It can be observed in Fig. 6a that our model maintains superior performance across the None-FGSM-PGD task sequence while exhibiting a moderate increase in memory usage (487MB). This increment is a justified trade-off compared to LRRF's performance gains, as the memory consumption remains within an acceptable range for practical deployment.

Training efficiency. The figure also presents a comparative analysis of total training times across all methods. While LRRF demonstrates a moderate training time increase relative to most approaches, this corresponds to more thorough optimization through extended adversarial training phases, which is an expected phenomenon in continual learning scenarios. Notably, compared to DER++AFLC, our method achieves comprehensive performance dominance while consuming only 21.04% of the total training time (78.96% reduction). This remarkable acceleration highlights the exceptional time complexity optimization of our framework.

Resource-cost efficiency trade-off. In summary, LRRF achieves state-of-the-art robustness with marginal computational overhead (487MB memory increase) and substantial training time savings (78.96% reduction vs. DER++AFLC). LRRF incurs moderate additional compute relative to lightweight replay baselines, but remains substantially cheaper than DER++AFLC while achieving stronger robustness performance.



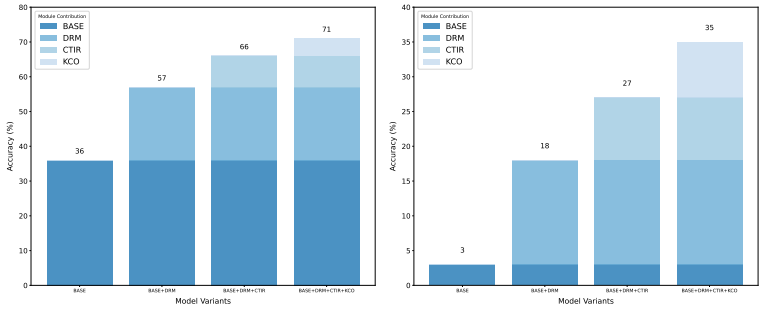
(a) Total memory usage (b) Total training time

Figure 6: (a): Total memory usage of all methods on the CIFAR10 dataset under the None-FGSM-PGD task sequence. (b): Total training time of all methods on the CIFAR10 dataset under the None-FGSM-PGD task sequence.

C.8 Attack-type Ablation Results

In this section, we present additional ablation results to comprehensively and thoroughly demonstrate the importance of each component in our model and their critical roles under various attack types, as illustrated in Fig. 7.

FGSM Ablation results. Fig. 7a shows the ablation results of our LRRF under FGSM attack. Apparently, the model's superior adversarial performance is closely tied to the contributions of each module. Incorporating the DRM module increases accuracy by 21.0% compared to the base model, indicating a substantial performance gain. This improvement is particularly significant in continual adversarial settings, as the model must simultaneously resist new attacks while mitigating catastrophic forgetting of prior knowledge. Further integration of the CTIR module boosts accuracy by an additional 9.0%, demonstrating its efficacy in learning robust features from the historical teacher network to enhance cross-task generalization. Finally, embedding the KCO module further improves LRRF's performance by 5.0%, underscoring its critical role in reinforcing model robustness.



(a) FGSM Ablation on CIFAR10 (b) PGD Ablation on CIFAR10

Figure 7: (a): Ablation study of our LRRF on the CIFAR10 dataset after three sequential tasks (None-FGSM-PGD) under FGSM attack. (b): Ablation study of our LRRF on the CIFAR10 dataset after three sequential tasks (None-FGSM-PGD) under PGD attack.

PGD Ablation results. Fig. 7b further presents the ablation study of our model under PGD attack. The results reveal that LRRF's components exhibit even more pronounced effects under PGD attack. The DRM module elevates test accuracy from 3.0% to 18.0%—a sixfold performance increase—highlighting the pivotal role of adversarial sample generation and feature alignment in continual adversarial training. The CTIR module contributes an additional 9.0% accuracy gain, while the KCO module further enhances performance by 8.0%. Consequently, our method achieves a total accuracy of 35.0% under PGD attack, representing a 32.0% absolute improvement over the baseline (3.0%).

Collectively, these ablation experiments rigorously validate that LRRF's exceptional performance across sequential tasks and diverse attack scenarios stems from the synergistic contributions of all modules. Each component delivers critical and effective functionality, enabling LRRF to outperform all baseline methods and establish a new state-of-the-art in continual adversarial robustness.

C.9 Training Epoch Ablation Results

To explore whether asymmetric epoch budgets will constrain adaptation from the clean initialization, we ablated the adversarial training epochs under both FGSM and PGD attacks, with results summarized in Tab. 11.

Table 11: Average classification accuracy of LRRF on CIFAR-10 under different adversarial training epoch budgets.

Epoch (FGSM & PGD)	**CIFAR-10**		
	None	**FGSM**	**PGD**
1	86.31	83.14	**58.83**
5	86.33	83.83	57.43
10	**86.98**	**84.21**	58.26

The results show that increasing FGSM and PGD training epochs to 5 and 10, with clean epochs held constant, negligibly affects final accuracy. This confirms that adversarial task performance converges sufficiently after a single epoch for both attacks. Further epoch increases yield only marginal gains at linearly growing computational cost.

C.10 Buffer Ablation Results

In this section, we perform additional buffer ablation studies to examine whether our method fully exploits the stored samples. Two aspects are investigated. First, we vary the buffer size to test if the performance advantage of our LRRF model is sensitive to the buffer capacity. We use ImageNet-R to test whether buffer size affects model performance under high data complexity. Second, we replace the samples in the buffer with three types: clean samples only, a mixture of clean and adversarial samples, and clean samples augmented with our KCO alignment strategy. Results confirm that our model effectively utilizes buffered samples, leading to superior classification performance.

Table 12: LRRF's mean accuracy varies across buffer sizes on ImageNet-R.

Buffer Size	**ImageNet-R**		
	None	**FGSM**	**PGD**
200	**79.52**	68.69	54.53
500	78.67	68.56	53.76
1000	78.88	**69.43**	**54.96**

Buffer size ablation results. The results in Tab. 12 indicate that buffer size is not a primary factor affecting LRRF's performance. Even on the high-complexity dataset ImageNet-R, increasing the buffer size to 1000 has only a marginal impact. Under FGSM and PGD attacks, the average accuracy improves by only 0.74% and 0.43%, respectively. On clean samples, the model with a buffer size of 1000 performs even slightly worse than with 200, showing a 0.64% drop in average accuracy. These observations demonstrate that our method's mechanism effectively exploits the buffer, while the classification accuracy remains largely unaffected by buffer size.

Buffer type ablation results. The results for different buffer types are shown in Tab. 13. Under identical conditions, our KCO mechanism demonstrates superior buffer utilization. Mixing clean and adversarial samples not only fails to boost learning efficiency but actually degrades performance. Compared to using only clean samples, this mixed strategy reduces average accuracy by 5.83%, 9.43%, and 6.79% across three tasks. In contrast, adding our KCO mechanism to clean samples further improves model performance, especially under FGSM attacks. The average accuracy gains on

Table 13: LRRF's mean accuracy varies across buffer types on CIFAR-10.

Buffer Type	CIFAR-10		
	None	FGSM	PGD
Clean	83.73	67.47	53.14
Clean + Adversary	77.90	58.04	46.35
LRRF (Clean + KCO)	**86.31**	**83.14**	**58.83**

the three sequential tasks reach 2.58%, 15.67%, and 5.69%, respectively. These findings confirm that the KCO mechanism is indispensable for LRRF, particularly in enabling effective buffer exploitation.

C.11 Statistical stability verification

We have evaluated our LRRF model on ImageNet-R under FGSM and PGD attacks in Tab. 2. Although the accuracy gain over baselines is smaller than on other datasets, the advantage remains stable statistically. To support this, we run each method three times and report the mean and variance.

Table 14: The average classification accuracy of different models on ImageNet-R with statistical measures.

Method	ImageNet-R		
	None	FGSM	PGD
DER	76.58 ± 0.52	65.19 ± 0.58	48.73 ± 1.42
DER++	76.19 ± 0.38	65.53 ± 0.66	47.68 ± 1.85
DER+++	76.67 ± 0.17	64.71 ± 0.67	52.22 ± 0.34
DEF	76.23 ± 0.28	68.27 ± 0.32	53.26 ± 0.37
DER++STAR	76.51 ± 0.47	62.71 ± 0.45	42.83 ± 1.28
ER-ACE-STAR	75.32 ± 0.46	57.28 ± 0.78	33.47 ± 1.56
AIR	35.18 ± 14.37	29.73 ± 10.84	24.31 ± 8.72
DER++AFLC	73.26 ± 0.35	62.18 ± 0.52	47.62 ± 0.57
LRRF(Ours)	$\mathbf{78.73 \pm 0.08}$	$\mathbf{68.62 \pm 0.17}$	$\mathbf{53.82 \pm 0.37}$

Results. We report the mean and variance of each method over three runs in Tab. 14. Our LRRF consistently outperforms others across trials. On clean samples, it exceeds the closest competitor (DER+++) by 2.06% in mean accuracy. Under FGSM and PGD attacks, the improvements reach 0.35% and 0.56%, respectively. Moreover, the variance of our method on the three tasks is only 0.08%, 0.17%, and 0.37%, indicating little fluctuation across runs. These results confirm that while the advantage on ImageNet-R is less pronounced than on other datasets, the stable improvement of LRRF is statistically supported.

C.12 Class-incremental Learning Generalization

To evaluate how LRRF generalizes to the class-incremental setting, we thus conducted class-IL experiments on CIFAR-10 (2 tasks, 5 classes per task), with FGSM and CW as the adversarial tasks. We report clean accuracy, adversarial accuracy and average adversarial accuracy after training; results are summarized in.

As shown in the Tab. 15, LRRF preserves its robustness advantage from task-IL even when extended to class-IL. Specifically, clean accuracy trails AIR by only 2.89%, while adversarial performance improves substantially: gains of at least 4.43% on FGSM and 2.08% on CW, yielding an average improvement of 3.54%. This confirms that LRRF's advantage derives from its intrinsic mechanisms, not the continual learning protocol. Beyond validating generalization, these results provide meaningful guidance for future work.

Table 15: Classification accuracy under class-incremental learning settings on CIFAR-10.

Method	CIFAR-10			
	Clean	**FGSM**	**CW**	**Average**
DER+++	51.88	72.97	66.77	69.87
AIR	**67.92**	72.80	64.88	68.84
DER++AFLC	62.02	74.98	66.20	70.59
OURS	65.03	**79.41**	**68.85**	**74.13**

D Additional Theoretical Analysis

In this section, we provide a theoretical analysis of LRRF in the setting of Continual Adversarial Defense (CAD). According to the methodology, LRRF consists of three components: intra-task representation alignment via DRM, cross-task invariant representation regularization via CTIR, and class-conditional knowledge consolidation via KCO. The core idea is to achieve both robustness improvement and forgetting mitigation under a sequence of evolving adversarial tasks through hierarchical representation alignment.

D.1 Problem Setup and Analysis Objectives

Let $\mathcal{X}$ denote the input space, $\mathcal{Y} = \{1, \ldots, C\}$ the label space, and $\mathcal{Z}$ the representation space. For the i-th task, let the representation network and classifier head be defined as

$$f_i : \mathcal{X} \to \mathcal{Z}, \qquad g_i : \mathcal{Z} \to \Delta^{C-1}, \tag{22}$$

where Δ^{C-1} denotes the C-dimensional probability simplex. The overall model is therefore given by

$$h_i(\mathbf{x}) = g_i(f_i(\mathbf{x})). \tag{23}$$

In the CAD setting, the task sequence is induced by different white-box attacks. The first task corresponds to the clean data distribution, while subsequent tasks correspond to adversarial distributions generated by different attack algorithms. The model is required to learn robustness against new attacks while preserving classification ability and robust representations for previously encountered tasks.

From a theoretical perspective, the effectiveness of LRRF can be reduced to the following three questions:

1. Why does minimizing the representation discrepancy between clean and adversarial samples improve adversarial robustness?
2. Why does aligning the current representation network with the historical accumulated representation network help suppress catastrophic forgetting?
3. Why does class-conditional alignment with a memory buffer preserve category structure?

To answer these questions, this section provides theoretical results from three perspectives: maximum mean discrepancy (MMD), risk gap bounds, and class compactness/separation analysis.

D.2 Preliminaries

In order to simplify notations and to improve readability, we define the following notations for theoretical analysis. Let $\mathcal{H}$ be the reproducing kernel Hilbert space (RKHS) induced by a kernel function $k(\cdot, \cdot)$. For any probability distribution P, its kernel mean embedding is defined as

$$\mu_P = \mathbb{E}_{z \sim P}[k(\mathbf{z}, \cdot)] \in \mathcal{H}. \tag{24}$$

For two probability distributions P and Q, the MMD is defined as

$$\mathrm{MMD}(P, Q) = \|\mu_P - \mu_Q\|_{\mathcal{H}}. \tag{25}$$

If k is a characteristic kernel, then the mapping $P \mapsto \mu_P$ is injective, and hence

$$\mathrm{MMD}(P, Q) = 0 \Longleftrightarrow P = Q. \tag{26}$$

We further denote:

- P_i^c: the clean representation distribution in task i;
- P_i^a: the adversarial representation distribution in task $\mathcal{T}_i$;
- P_i^{cur}: the representation distribution produced by the current network in the i-th task;
- P_i^{acc}: the representation distribution produced by the accumulated network on the i-th task;
- $P_{i,y}^a$: the adversarial representation distribution of class y in task i;
- $P_{M,y}^c$: the clean representation distribution of class y in the memory buffer.

D.3 Theoretical Analysis of DRM: Intra-Task Robust Representation Consistency

The purpose of DRM is to align the multi-level representation distributions of clean and adversarial samples during the training process, thereby reducing semantic shifts caused by adversarial perturbations. We first show the distributional meaning of this alignment.

Theorem D.1 (Distributional Consistency Induced by DRM)**.** *Assume that the kernel k is characteristic. If, for task i,*

$$\text{MMD}(P_i^c, P_i^a) = 0, \tag{27}$$

then

$$P_i^c = P_i^a. \tag{28}$$

Proof. By the definition of MMD,

$$\text{MMD}(P_i^c, P_i^a) = \|\mu_{P_i^c} - \mu_{P_i^a}\|_{\mathcal{H}}. \tag{29}$$

If $\text{MMD}(P_i^c, P_i^a) = 0$, then

$$\mu_{P_i^c} = \mu_{P_i^a}. \tag{30}$$

Since k is characteristic, the kernel mean embedding is injective. Therefore,

$$P_i^c = P_i^a. \tag{31}$$

This completes the proof. □

Theorem D.1 shows that DRM does not merely reduce pairwise distances between individual samples; rather, it enforces the clean and adversarial samples to share the same overall distribution in representation space. As a result, the semantic features used by the classifier become less sensitive to adversarial perturbations.

We next connect representation alignment with risk discrepancy.

Assumption D.2 (Lipschitz Continuity of the Classifier Head)**.** There exists a constant $L_g > 0$ such that for any $\mathbf{z}, \mathbf{z}' \in \mathcal{Z}$,

$$\|g_i(\mathbf{z}) - g_i(\mathbf{z}')\| \leq L_g \|\mathbf{z} - \mathbf{z}'\|. \tag{32}$$

Assumption D.3 (Lipschitz Continuity of the Loss)**.** For any fixed label y, the loss $\ell(g_i(\mathbf{z}), y)$ is L_ℓ-Lipschitz with respect to z, i.e.,

$$|\ell(g_i(\mathbf{z}), y) - \ell(g_i(\mathbf{z}'), y)| \leq L_\ell \|z - \mathbf{z}'\|. \tag{33}$$

Theorem D.4 (Risk Gap Bound Induced by DRM)**.** *Let $\varphi_y(\mathbf{z}) = \ell(g_i(\mathbf{z}), y)$, and suppose there exists a constant $C > 0$ such that φ_y / C belongs to the unit ball of the RKHS. Then*

$$\left|\mathbb{E}_{z\sim P_i^c} \ell(g_i(\mathbf{z}), y) - \mathbb{E}_{z\sim P_i^a} \ell(g_i(\mathbf{z}), y)\right| \leq C\, \text{MMD}(P_i^c, P_i^a). \tag{34}$$

Proof. By the dual form of MMD,

$$\text{MMD}(P, Q) = \sup_{\|f\|_{\mathcal{H}} \leq 1} \left(\mathbb{E}_{z\sim P}[f(\mathbf{z})] - \mathbb{E}_{z\sim Q}[f(\mathbf{z})]\right). \tag{35}$$

Let $f(\mathbf{z}) = \varphi_y(\mathbf{z})/C$. By assumption, $\|f\|_{\mathcal{H}} \leq 1$. Therefore,

$$\left|\mathbb{E}_{z\sim P_i^c}\left[\frac{\varphi_y(\mathbf{z})}{C}\right] - \mathbb{E}_{z\sim P_i^a}\left[\frac{\varphi_y(\mathbf{z})}{C}\right]\right| \leq \text{MMD}(P_i^c, P_i^a). \tag{36}$$

Multiplying both sides by C yields

$$\left|\mathbb{E}_{z\sim P_i^c} \ell(g_i(\mathbf{z}), y) - \mathbb{E}_{z\sim P_i^a} \ell(g_i(\mathbf{z}), y)\right| \leq C\, \text{MMD}(P_i^c, P_i^a). \tag{37}$$

This completes the proof. □

Theorem D.4 implies that as DRM drives $\mathrm{MMD}(P_i^c, P_i^a)$ toward zero, the expected loss gap between clean and adversarial samples is simultaneously reduced. Hence, DRM provides a direct distribution-level explanation for robustness improvement.

D.4 Theoretical Analysis of CTIR: Cross-Task Stability and Forgetting Control

For a fixed distribution D on $\mathcal{X}\times\mathcal{Y}$, define $R_D(h) = \mathbb{E}_{(X,Y)\sim D}\ell(h(X), Y)$ and $\pi_D(y) = \Pr_D(Y = y)$. For each label with $\pi_D(y) > 0$, let $P^{\mathrm{cur}}_{i,D,y} = \mathrm{Law}(f_i(X) \mid Y = y)$ and $P^{\mathrm{acc}}_{i,D,y} = \mathrm{Law}(f_{i-1}(X) \mid Y = y)$, where both laws are induced by the same $(X, Y) \sim D$. Set

$$d_i(D) = \sum_{y:\pi_D(y)>0} \pi_D(y)\mathrm{MMD}(P^{\mathrm{cur}}_{i,D,y}, P^{\mathrm{acc}}_{i,D,y}). \tag{38}$$

The loss change due to updating the classifier head is measured by

$$b_i(D) = \mathbb{E}_{(X,Y)\sim D} \left|\ell(g_i(f_{i-1}(X)), Y) - \ell(g_{i-1}(f_{i-1}(X)), Y)\right|. \tag{39}$$

Theorem D.5 (One-Step Risk Drift Bound)**.** *Assume that the kernel mean embeddings exist, the relevant risks and $b_i(D)$ are finite, and, for each y with $\pi_D(y) > 0$, $\varphi_{i,y}(z) = \ell(g_i(z), y) \in \mathcal{H}$ with $\|\varphi_{i,y}\|_{\mathcal{H}} \le C_1$. If $d_i(D) \le \varepsilon_i(D)$, then*

$$|R_D(h_i) - R_D(h_{i-1})| \le C_1\varepsilon_i(D) + b_i(D). \tag{40}$$

Proof. Add and subtract $R_D(g_i \circ f_{i-1})$. For the representation term, the reproducing property and Cauchy–Schwarz inequality give

$$\begin{aligned}
&|R_D(g_i \circ f_i) - R_D(g_i \circ f_{i-1})| \\
&= \left| \sum_{y:\pi_D(y)>0} \pi_D(y) \left\langle \varphi_{i,y}, \mu_{P^{\mathrm{cur}}_{i,D,y}} - \mu_{P^{\mathrm{acc}}_{i,D,y}} \right\rangle_{\mathcal{H}} \right| \\
&\le C_1 \sum_{y:\pi_D(y)>0} \pi_D(y)\mathrm{MMD}(P^{\mathrm{cur}}_{i,D,y}, P^{\mathrm{acc}}_{i,D,y}) \\
&= C_1 d_i(D) \le C_1\varepsilon_i(D).
\end{aligned}$$

For the classifier-head term,

$$|R_D(g_i \circ f_{i-1}) - R_D(g_{i-1} \circ f_{i-1})| \le b_i(D).$$

The triangle inequality completes the proof. □

Equation (12) penalizes current-task marginal MMD, whereas the bound uses class-conditional MMD and classifier-head loss changes on D. Marginal alignment alone does not control these terms. For an old task, they must be evaluated or bounded on that task; transferring a discrepancy bound from current-task data requires an additional assumption.

Theorem D.6 (Conditional Cumulative Forgetting Bound)**.** *Fix an old task $t < n$ and write $R_t = R_{D_t}$. Suppose that for every $i = t + 1, \ldots, n$, Theorem D.5 applies with $D = D_t$, $d_i(D_t) \le \varepsilon_{i,t}$, and a uniform constant C_1. Then*

$$R_t(h_n) - R_t(h_t) \le \sum_{i=t+1}^{n} \left(C_1\varepsilon_{i,t} + b_i(D_t)\right). \tag{41}$$

Proof. The telescoping identity

$$R_t(h_n) - R_t(h_t) = \sum_{i=t+1}^{n} \big(R_t(h_i) - R_t(h_{i-1})\big)$$

and the one-step bound on the same fixed D_t yield the result. □

D.5 Theoretical Analysis of KCO: Class-Conditional Structure Preservation

KCO aligns current adversarial samples with same-class clean samples stored in the memory buffer, with the goal of preserving semantic class centers and local category structure. We analyze its role from the perspectives of class-center shift and inter-class margin preservation.

Define the class center of class y as

$$\mu_y = \mathbb{E}[\mathbf{z} \mid y]. \tag{42}$$

Define the intra-class variance of class y as

$$\sigma_y^2 = \mathbb{E}\left[\|\mathbf{z} - \mu_y\|^2 \mid y\right]. \tag{43}$$

For any two different classes $y \neq y'$, define the inter-class margin as

$$\Gamma_{y,y'} = \|\mu_y - \mu_{y'}\|. \tag{44}$$

Theorem D.7 (Class-Center Shift Control Induced by KCO)**.** *For any class y, if KCO satisfies*

$$\mathrm{MMD}(P_{i,y}^a, P_{M,y}^c) \leq \delta_y, \tag{45}$$

then

$$\|\mu_{i,y}^a - \mu_{M,y}^c\|_{\mathcal{H}} \leq \delta_y, \tag{46}$$

where $\mu_{i,y}^a$ denotes the kernel mean embedding of the adversarial representation distribution of class y in the current task, and $\mu_{M,y}^c$ denotes the kernel mean embedding of the clean representation distribution of class y in the memory buffer.

Proof. By the definition of MMD,

$$\mathrm{MMD}(P_{i,y}^a, P_{M,y}^c) = \|\mu_{i,y}^a - \mu_{M,y}^c\|_{\mathcal{H}}. \tag{47}$$

Combining this with the assumption

$$\mathrm{MMD}(P_{i,y}^a, P_{M,y}^c) \leq \delta_y, \tag{48}$$

we directly obtain

$$\|\mu_{i,y}^a - \mu_{M,y}^c\|_{\mathcal{H}} \leq \delta_y. \tag{49}$$

This completes the proof. □

Theorem D.7 shows that KCO constrains the current adversarial class center to remain close to the historical clean class center, thereby stabilizing semantic class identity.

Theorem D.8 (Lower Bound on Inter-Class Margin Under KCO)**.** *For any two distinct classes $y \neq y'$, if*

$$\|\mu_{i,y}^a - \mu_{M,y}^c\| \leq \delta_y, \qquad \|\mu_{i,y'}^a - \mu_{M,y'}^c\| \leq \delta_{y'}, \tag{50}$$

then the adversarial inter-class margin in the current task satisfies

$$\|\mu_{i,y}^a - \mu_{i,y'}^a\| \geq \|\mu_{M,y}^c - \mu_{M,y'}^c\| - (\delta_y + \delta_{y'}). \tag{51}$$

Proof. By the triangle inequality,

$$\begin{aligned}\|\mu_{M,y}^c - \mu_{M,y'}^c\| \leq & \|\mu_{M,y}^c - \mu_{i,y}^a\| + \|\mu_{i,y}^a - \mu_{i,y'}^a\| \\ & + \|\mu_{i,y'}^a - \mu_{M,y'}^c\|.\end{aligned} \tag{52}$$

Rearranging the terms yields

$$\|\mu_{i,y}^a - \mu_{i,y'}^a\| \geq \|\mu_{M,y}^c - \mu_{M,y'}^c\| - \|\mu_{M,y}^c - \mu_{i,y}^a\| - \|\mu_{i,y'}^a - \mu_{M,y'}^c\|. \tag{53}$$

Applying the assumptions

$$\|\mu_{M,y}^c - \mu_{i,y}^a\| \leq \delta_y, \qquad \|\mu_{i,y'}^a - \mu_{M,y'}^c\| \leq \delta_{y'}, \tag{54}$$

we obtain

$$\|\mu_{i,y}^a - \mu_{i,y'}^a\| \geq \|\mu_{M,y}^c - \mu_{M,y'}^c\| - (\delta_y + \delta_{y'}). \tag{55}$$

This completes the proof. □

Theorem D.8 indicates that as long as KCO keeps the class-center deviations sufficiently small, the inter-class separation in the historical clean representation space can be transferred to the adversarial representation space of the current task. Hence, KCO supports not only intra-class consistency but also decision-boundary stability.

D.6 Overall Theoretical Interpretation of LRRF

By combining DRM, CTIR, and KCO, the overall objective of LRRF can be interpreted as jointly optimizing the following three levels:

$$\min_{f,g} \quad R_{\mathrm{adv}}(f,g) + \lambda_1 D_{\mathrm{intra}} + \lambda_2 \{D_{\mathrm{cross}} + D_{\mathrm{class}}\}, \tag{56}$$

where:

- $R_{\mathrm{adv}}(f,g)$ denotes the adversarial risk on the current task;
- D_{intra} denotes the intra-task clean–adversarial discrepancy induced by DRM;
- D_{cross} denotes the cross-task representation drift induced by CTIR;
- D_{class} denotes the class-conditional structural deviation induced by KCO.

These regularizers constrain complementary aspects of the representation: DRM encourages clean–adversarial alignment, CTIR penalizes discrepancies between current and accumulated representations, and KCO encourages class-conditional alignment.

Proposition D.9 (Joint Property of LRRF)**.** *Suppose that throughout continual task learning:*

1. *DRM keeps* $\mathrm{MMD}(P_i^c, P_i^a)$ *small;*
2. *for each old task* t*, the assumptions of Theorem D.6 hold on* D_t*;*
3. *KCO keeps the class-center deviations* δ_y *small for all classes.*

Then LRRF enjoys the following properties simultaneously:

1. *the risk gap between clean and adversarial samples is controlled;*
2. *the old-task risk increment satisfies the bound in Theorem D.6;*
3. *intra-class compactness and inter-class separation are preserved, thereby improving cross-attack generalization.*

Proof. The first statement follows directly from Theorem D.4; the second follows from Theorems D.5 and D.6; and the third follows from Theorems D.7 and D.8. Combining these results proves the proposition. □

The CTIR statement in Proposition D.9 follows from Theorem D.6, with the class-conditional discrepancies and classifier-head loss changes evaluated on each old task.

D.7 Summary of This Section

In this section, we theoretically analyzed the three key components of LRRF and obtained the following conclusions:

1. DRM minimizes the MMD distance between clean and adversarial samples in representation space, encouraging them to share the same semantic distribution and thereby reducing the risk discrepancy between clean and adversarial inputs;
2. CTIR penalizes discrepancies between current and accumulated representations; under the assumptions of Theorem D.6, old-task risk drift is bounded by the accumulated representation and classifier-head terms;
3. KCO performs class-conditional pairing and alignment, limiting the deviation of current adversarial class centers from historical clean class centers and providing a lower bound on inter-class margins, thereby preserving category structure.

Therefore, LRRF can be understood as optimizing a hierarchical representation learning objective that simultaneously balances current-task robustness, historical stability, and class-structure preservation. This provides a theoretical explanation for its strong performance in continual adversarial defense.

E Broader Impacts

This work advances continual adversarial defense, which can strengthen the reliability of deep learning systems operating in non-stationary, adversary-prone environments (e.g., autonomous driving, medical imaging, or content filtering). By preserving robust feature representations across task sequences, our framework reduces the risk of catastrophic forgetting under white-box attacks, thereby supporting safer model deployment.

However, improved robustness might also lower the barrier for deploying systems in high-stakes surveillance or be misused to harden malicious models against defensive countermeasures. Moreover, the increased computational cost of adversarial training could amplify environmental footprints. We encourage responsible adoption and further analysis of dual-use risks, fairness implications, and energy efficiency in continual robust learning.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We have discussed the paper's contributions and scope in the abstract and introduction.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have provided discussions in the conclusion section.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: For each theoretical result, we provide either experimental evidence or analytical (formula-based) justification.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: All relevant information is provided in the Appendix to ensure the reproducibility of our work.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The relevant code and usage instructions are provided in the supplementary material.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: All relevant information is provided in the Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: All relevant information is provided in Appendix C.6.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: All relevant information is provided in Appendix C.3.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our work has thoroughly read and adhered to the NeurIPS Code of Ethics from all perspectives.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We have discussed both potential positive societal impacts and negative societal impacts of the work performed in Appendix E.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Our paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All assets (code, data, models) used in this paper are properly credited, and their licenses and terms of use are explicitly mentioned and respected.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The new assets are provided as a .zip file in the supplementary material.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.