TOWARDS CONTINUAL ROBUSTNESS: ADAPTIVE EXPERT COMPOSITION FOR EVOLVING ADVERSARIAL ATTACKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Continual adversarial defense aims to incrementally adapt a model to newly emerging adversarial attacks while preserving robustness against previously encountered ones. Existing methods primarily focus on mitigating catastrophic forgetting but largely overlook two fundamental challenges: effectively reusing transferable knowledge across evolving attacks and improving generalization to unseen adversarial distributions. To address these limitations, we propose a parameter-efficient continual adversarial adaptation framework based on Low-Rank Expert Expansion and Residual Composition (LREERC). Specifically, we incrementally expand the model with task-specific low-rank experts and reuse historical knowledge through adaptive residual composition without modifying previously learned experts. To avoid negative transfer, we further introduce a gradient-fingerprint-based expert retrieval mechanism, which selects optimization-compatible historical experts according to adversarial gradient similarity, together with a probability-weighted orthogonality regularization that discourages alignment between expert parameters during continual adaptation. Moreover, we propose a Dual-Channel Adaptive Perturbation Augmentation (DCAPA) strategy that jointly explores attack-specific perturbation directions guided by transferable experts and more universal vulnerability directions shared by the clean and current experts, broadening local perturbation sampling and improving robustness beyond observed attacks. Extensive experiment results demonstrate that our approach achieves state-of-the-art performance.

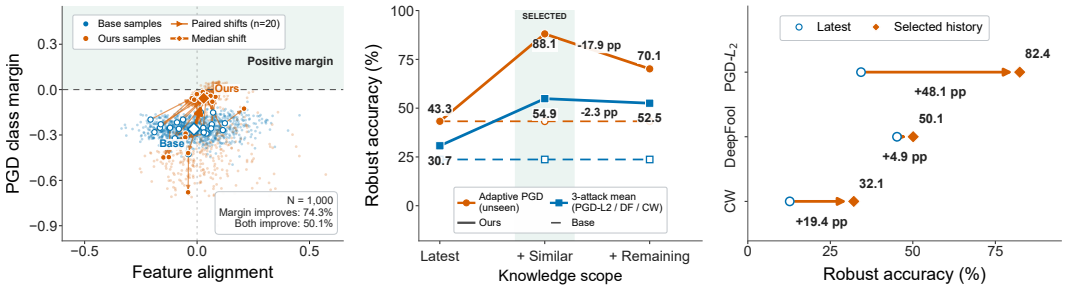
1 INTRODUCTION

Deep neural networks have achieved remarkable success in computer vision Krizhevsky et al. (2012), yet their vulnerability to adversarial perturbations remains a major challenge for reliable deployment in safety-critical applications. While adversarial training improves robustness against specific attacks, most methods assume a fixed attack distribution Zhou & Hua (2024). In reality, attacks continuously evolve, requiring defense models to adapt to unseen threats while retaining robustness against previous attacks Mi et al. (2025).

Continual learning provides a natural paradigm for addressing this challenge by learning from sequentially arriving tasks without requiring access to historical training data Zenke et al. (2017); Pan et al. (2020). Recent studies have begun to formulate continual adversarial defense as a task-incremental learning problem Zhou & Hua (2024), where each task corresponds to a newly observed adversarial attack. Nevertheless, directly extending conventional continual learning techniques to adversarial robustness remains highly challenging Ye et al. (2025). Unlike standard classification tasks whose distributions differ primarily at the semantic level Buzzega et al. (2020), successive adversarial attacks often exhibit subtle yet highly structured relationships in perturbation space.

Effective continual adversarial adaptation therefore requires simultaneously preserving previously acquired robustness, selectively transferring useful adversarial knowledge across related attacks, and efficiently adapting to newly emerging perturbation patterns under strict computational constraints. However, despite the promising progress made by existing Continual Adversarial Defense (CAD) methods Zhou & Hua (2024); Mi et al. (2025), they largely overlook two critical challenges: (1) how to effectively transfer and reuse previously learned knowledge for adapting to emerging adversarial tasks, and (2) how to fully leverage accumulated knowledge to enhance model robustness against

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) Feature Transfer (b) Knowledge Sharing (c) Unseen-Attack Gains

Figure 1: **Motivation of the proposed framework.** (a) At initialization for a newly encountered PGD attack, LREERC improves clean-to-PGD feature alignment and true-class margins relative to the standard LoRA baseline without historical-expert fusion (*Base*), supporting beneficial historical knowledge transfer. (b) Sharing selected similar historical experts' robust knowledge for training-time perturbation augmentation (+ *Similar*) improves unseen-attack robustness over the latest expert without such sharing (*Latest*), whereas further including the remaining experts (+ *Remaining*) reduces these gains. (c) Selected historical knowledge improves robustness across unseen PGD-ℓ_2, DeepFool, and CW attacks. Hollow circles and filled diamonds indicate results without and with this knowledge sharing, respectively; "pp" denotes percentage points. Detailed experimental protocols and metric definitions are provided in Appendix D.9.

future unknown and unseen adversarial attacks. Due to these limitations, existing approaches remain inadequate in achieving satisfactory robustness against both seen and unseen adversarial attacks under the CAD scenario.

To address the **first challenge**, we distinguish shared knowledge from the specialized information retained in individual adapters. CL-LoRA He et al. (2025) learns cross-task commonalities through shared adapters, while its separate task-specific modules preserve specialized knowledge. This separation limits the direct reuse of historical task-specific residuals by a new adapter, as shown in Figure 1a, motivating explicit retrieval and composition of such residuals. To this end, we propose Low-Rank Expert Expansion and Residual Composition (LREERC), which enables positive knowledge transfer through adaptive residual composition. Figure 1a illustrates this transfer at model initialization for a newly encountered PGD attack in which incorporating historical experts through a knowledge reuse mechanism yields positive median changes in feature alignment and PGD margin relative to the model before fusion. These changes are consistent with beneficial feature transfer, supporting the use of historical knowledge to initialize adaptation to new attacks. To make this knowledge reuse more efficient, we introduce a gradient-fingerprint-based historical expert selection strategy in the knowledge reuse process, in which each expert is represented by a compact gradient fingerprint computed on adversarial data after training, and cosine similarity between historical and current fingerprints serves as a proxy for optimization compatibility. Only experts with high fingerprint similarity participate in residual fusion, while those with low similarity are excluded. In addition, LREERC incorporates probability-weighted orthogonality regularization that encourages newly learned experts to capture complementary adversarial characteristics, preserving expert diversity during continual adaptation.

Beyond knowledge reuse, we address the **second challenge** by exploring the potential perturbation space. We achieve this by proposing a Dual-Channel Adaptive Perturbation Augmentation (DCAPA) strategy. The first channel of DCAPA generates attack-specific perturbations guided jointly by the current expert and selected historical experts, thereby incorporating transferable attack characteristics accumulated from previous tasks. The second channel of DCAPA explores more general vulnerability directions by jointly exploiting gradients from the clean base expert and the current expert. These two channels broaden local adversarial exploration, aiming to improve generalization beyond individual attack algorithms while maintaining robustness against previously encountered perturbations. Figure 1b provides evidence for the benefit of sharing accumulated robust knowledge during training in which perturbation generation guided by selected similar historical experts improves robustness on the evaluated unseen attacks relative to the latest expert without this knowledge sharing, whereas additionally incorporating the remaining experts reduces these gains. This comparison supports selectively using historical knowledge to guide perturbation generation. Fig-

ure 1c further shows improvements across unseen PGD-ℓ_2, DeepFool, and CW attacks, indicating that this knowledge sharing benefits robustness across different attack algorithms.

We construct a series of experiments and empirical results demonstrate that our approach achieves the state-of-the-art performance. The main contributions of this paper are summarized as follows: (1) We propose a LREERC framework that enables positive knowledge transfer through an adaptive residual composition design; (2) We introduce a gradient-fingerprint-based historical expert retrieval mechanism together with probability-weighted orthogonality regularization, allowing the model to selectively reuse optimization-compatible adversarial knowledge while maintaining complementary expert representations; (3) We propose a dual-channel adaptive perturbation augmentation strategy that jointly explores attack-specific and universal vulnerability directions, substantially enlarging the local adversarial neighborhood and improving continual robustness as well as generalization to unseen adversarial variations.

2 Related Work

Continual learning (CL) enables sequential learning while preserving previously acquired knowledge. Replay mitigates forgetting by revisiting stored examples Zhou et al. (2026); Chu et al. (2026); Deng et al. (2025); Skierś & Deja (2025); Eskandar et al. (2025); Wang et al. (2025b); Bang et al. (2022); Tiwari et al. (2022); Huo et al. (2024), regularization constrains updates to preserve previous knowledge Eskandar et al. (2025); Dohare et al. (2024); Martens & Grosse (2015); Chaudhry et al. (2019); Lopez-Paz & Ranzato (2017); Derakhshani et al. (2021); Shi et al. (2021); Wang et al. (2021b); Nguyen et al. (2018); Ahn et al. (2019); Deng et al. (2021); Jha et al. (2024), and knowledge distillation transfers teacher responses to preserve prior behavior Chu et al. (2026); Hasan et al. (2025); Skierś & Deja (2025); Yan et al. (2024). PDFK Zhou et al. (2026) further addresses knowledge fragility in online CL through temporal smoothing and consistency regularization.

Adversarial defense improves robustness to adversarial perturbations through ensembles, denoising, and input transformations Jin et al. (2019); Xie et al. (2017), adversarial training against input or feature perturbations Jia et al. (2022); Madry et al. (2017); Mustafa et al. (2020), and distillation with softened teacher targets Goldblum et al. (2020); Wang et al. (2019); Zhu et al. (2021); Wang et al. (2021a); Zi et al. (2021). Complementary expert representations strengthen ensemble defenses Deng & Mu (2024), while attack studies evaluate robustness across perturbation strategies Pang et al. (2020). Memory replay has also been explored for adversarial defense in CL Zhou & Hua (2024). AAOpt Wesego & Rooshenas (2026) couples clean-image priors with learned adversarial perturbation scores for test-time diffusion purification, whereas DRIFT Guesmi & Shafique (2026) learns filter ensembles to disrupt gradient consensus and reduce adversarial transferability. More discussions are provided in **Appendix** C.

3 Methodology

3.1 Problem Formulation

We consider continual adversarial defense under the task-incremental learning paradigm Zhou & Hua (2024). Let the clean training dataset be denoted as $\mathcal{D} = \{(\mathbf{x}_j, \mathbf{y}_j)\}_{j=1}^{n}$, where $\mathbf{x}_j$ and $\mathbf{y}_j$ represent the input sample and its corresponding label, respectively. Assume a sequence of adversarial attack algorithms $\{\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_T\}$, where each attack $\mathcal{B}_t$ transforms the clean dataset into an adversarial counterpart $\mathcal{D}_t = \{(\mathcal{B}_t(\mathbf{x}), \mathbf{y}) \mid (\mathbf{x}, \mathbf{y}) \in \mathcal{D}\}$. Based on these datasets, we define a sequence of learning tasks $\{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_T\}$, where the initial task $\mathcal{T}_1$ corresponds to learning from clean data, i.e., $\mathcal{D}_1 = \mathcal{D}$, while each subsequent task $\mathcal{T}_t$ ($t > 1$) is associated with the adversarial dataset generated by attack $\mathcal{B}_t$. The model is trained in a sequential manner, receiving only the data from the current task. Once training advances to task $\mathcal{T}_t$, the training samples from earlier tasks $\{\mathcal{T}_1, \ldots, \mathcal{T}_{t-1}\}$ are assumed to be unavailable. After completing the entire task sequence, the learned model is evaluated on separate test sets under both the clean setting and all previously encountered adversarial attacks.

3.2 Low-Rank Expert Expansion and Residual Composition

A straightforward solution for continual adversarial adaptation is to fine-tune the entire backbone whenever a new attack is observed. However, updating all model parameters not only incurs substantial computational overhead but also inevitably overwrites knowledge acquired from previous tasks, resulting in severe catastrophic forgetting. To enable efficient continual adaptation while preserving previously learned robustness, we instead maintain a frozen pretrained backbone and incrementally expand the model using lightweight task-specific residual experts.

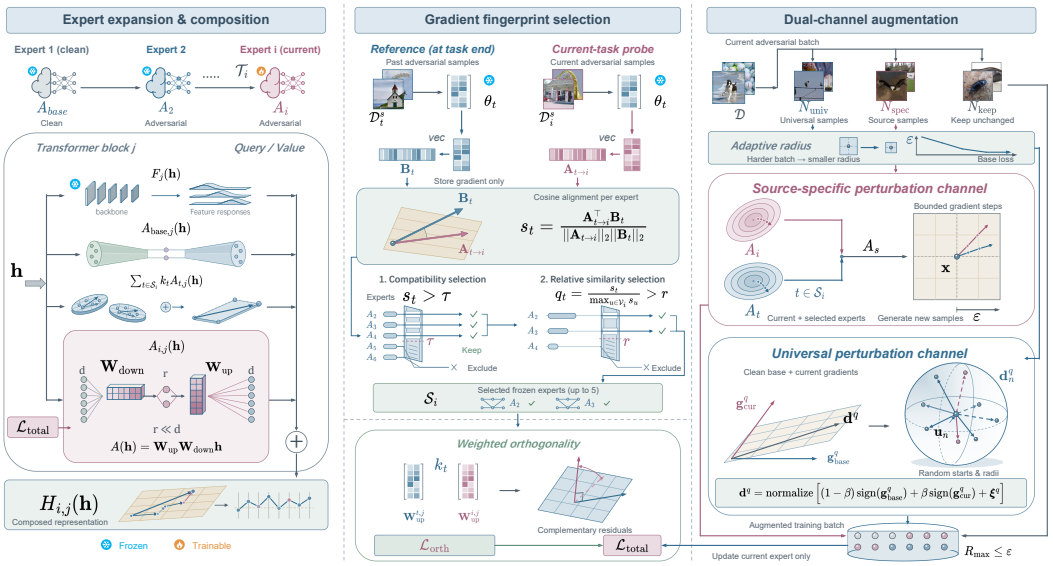


Figure 2: Our framework includes three core mechanisms (left to right): (1) Low-rank expert expansion and residual composition (LREERC) enables parameter-efficient adaptation to new attacks, while residual composition reuses historical knowledge with the backbone and previous experts frozen. (2) Gradient fingerprint selection retrieves optimization-compatible historical experts, while probability-weighted orthogonality regularization encourages the current expert to learn complementary representations. (3) Dual-channel adaptive perturbation augmentation explores expert-guided attack-specific directions and vulnerabilities shared by the clean and current experts, broadening the local adversarial neighborhood to improve generalization to unseen attacks.

Let F denote a pretrained Vision Transformer backbone whose parameters remain frozen throughout continual learning. Instead of fine-tuning the full backbone for every incoming task, we introduce a task-specific Low-Rank Adaptation (LoRA) expert into the query and value projections of each Transformer block. For a hidden representation $\mathbf{h} \in \mathbb{R}^d$, a LoRA module produces an additive correction $A(\mathbf{h}) = \mathbf{W}_{\text{up}}\mathbf{W}_{\text{down}}\mathbf{h},$, where $\mathbf{W}_{\text{down}} \in \mathbb{R}^{r\times d}$ and $\mathbf{W}_{\text{up}} \in \mathbb{R}^{d\times r}$ are the down-projection and up-projection matrices. $r \ll d$ is the low-rank dimension. Compared with learning a complete $d \times d$ adaptation matrix, which requires $\mathcal{O}(d^2)$ parameters, the low-rank formulation introduces only $\mathcal{O}(2dr)$ additional parameters. In our implementation, $\mathbf{W}_{\text{up}}$ is initialized to zero such that the LoRA branch initially produces no residual perturbation to the pretrained representation.

The first task learns a clean base expert A_{base}, which is then permanently frozen. For each subsequent task $\mathcal{T}_i$, a new expert A_i is independently optimized, while historical experts provide frozen residual corrections during inference. Unlike methods that merge parameters or repeatedly fine-tune previous modules, our framework preserves prior knowledge through residual composition. Specifically, for either the query or the value projection in Transformer block j, the composed mapping is:

$$H_{i,j}(\mathbf{h}) = F_j(\mathbf{h}) + A_{\text{base},j}(\mathbf{h}) + A_{i,j}(\mathbf{h}) + \sum\nolimits_{t\in\mathcal{S}_i} \{k_t A_{t,j}(\mathbf{h})\}, \tag{1}$$

Here F_j is the corresponding frozen pretrained projection, and the $A_{\cdot,j}$ terms are LoRA residuals at that projection. The expression applies separately to query and value; the resulting tensors enter the standard nonlinear attention computation. The set $\mathcal{S}_i$ contains the historical experts selected for task $\mathcal{T}_i$. The coefficient k_t controls the contribution of historical expert A_t. Historical knowledge is reused via additive residual composition, with previous experts frozen and only the new expert updated. This prevents parameter overwriting and mitigates catastrophic forgetting.

3.3 Gradient-Fingerprint-Based Historical Expert Selection

Although residual expert composition enables knowledge reuse across tasks, not every historical expert is beneficial for a newly encountered adversarial attack. Different attack algorithms often induce substantially different optimization characteristics, and indiscriminately combining all previously learned experts may introduce conflicting residual corrections that hinder adaptation. Therefore, before residual composition, we first estimate the transferability of each historical expert and retain only those that exhibit high optimization consistency with the current task.

To this end, we associate every historical expert with a compact gradient fingerprint, which characterizes the optimization direction learned during its original training. Since the final Transformer block contains the highest-level semantic representation while introducing only a small number of LoRA parameters, we use the up-projection matrix of the LoRA module in the last Transformer block as the fingerprint parameter. Specifically, let θ_t denote the corresponding parameter of historical expert A_t. After completing task $\mathcal{T}_t$, we compute and store a reference gradient on a sampled subset $\mathcal{D}_t^s$, expressed as:

$$\mathbf{B}_t = \mathrm{vec}\left(\nabla_{\theta_t}\mathcal{L}(A_t; \mathcal{D}_t^s)\right), \tag{2}$$

where $\mathrm{vec}(\cdot)$ vectorizes the gradient matrix and $\mathcal{L}(A_t, \mathcal{D}_t^s)$ is the cross-entropy loss function, defined as:

$$\mathcal{L}(A_t, \mathcal{D}_t^s) = -\mathbb{E}_{(\mathbf{x},\mathbf{y})\sim P_{\mathcal{D}_t^s}}\left[\sum_{c=1}^{C} \mathbf{y}[c] \log A_t(\mathbf{x})[c]\right]. \tag{3}$$

where $P_{\mathcal{D}_t^s}$ denotes the distribution of $\mathcal{D}_t^s$. $\mathbf{y}[c]$ and $A_t(\mathbf{x})[c]$ denote the c-th dimension of the class label $\mathbf{y}$ and the prediction from A_t. C represents the total number of classes. When a new task $\mathcal{T}_i$ arrives, the same frozen historical expert is evaluated using sampled current-task data $\mathcal{D}_i^s$:

$$\mathbf{A}_{t\to i} = \mathrm{vec}\left(\nabla_{\theta_t}\mathcal{L}(A_t, \mathcal{D}_i^s)\right). \tag{4}$$

The compatibility between the historical task and the new task is quantified using cosine similarity:

$$s_t = \frac{\mathbf{A}_{t\to i}^{\top}\mathbf{B}_t}{||\mathbf{A}_{t\to i}||_2||\mathbf{B}_t||_2}, \tag{5}$$

where $\|\cdot\|_2$ denotes the Euclidean norm. A larger similarity indicates closer alignment with the stored reference gradient. Its conditional connection to residual-transfer benefit is established in Proposition 4.2. Historical adversarial experts are indexed by $\mathcal{H}_i = \{2, \ldots, i-1\}$. The first threshold gives

$$\mathcal{V}_i = \{t \in \mathcal{H}_i \mid s_t > \tau\}. \tag{6}$$

For nonempty $\mathcal{V}_i$, softmax determines composition coefficients:

$$k_t = \frac{\exp(s_t)}{\sum_{u\in\mathcal{V}_i}\exp(s_u)}, \qquad t \in \mathcal{V}_i. \tag{7}$$

Let $s_i^\star = \max_{u\in\mathcal{V}_i} s_u$ and $q_t = s_t/s_i^\star$. The second threshold selects relatively strong matches:

$$\mathcal{S}_i = \mathrm{Top}^s_{K_{\mathrm{res}}}(\{t \in \mathcal{V}_i \mid q_t > r\}). \tag{8}$$

Here $\mathrm{Top}^s_{K_{\mathrm{res}}}$ retains at most K_{res} candidates with the largest similarities. We use $\tau = 0.1$, $r = 0.6$, and $K_{\mathrm{res}} = 5$. Retained coefficients are not renormalized. If $\mathcal{V}_i = \varnothing$, then $\mathcal{S}_i = \varnothing$ and all historical coefficients are zero. Otherwise, at least one best-matching expert is retained. Appendix A.4 provides the analysis. On the other hand, although residual composition encourages effective knowledge transfer, the newly learned expert may gradually converge toward parameter directions already represented by selected historical experts, thereby reducing expert diversity. To alleviate this issue, we further introduce a probability-weighted orthogonality regularization that penalizes alignment between selected parameter directions feature subspaces,

$$\mathcal{L}_{\mathrm{orth}} = \frac{\lambda}{L|\mathcal{S}_i|}\sum\nolimits_{t\in\mathcal{S}_i}\sum\nolimits_{j=1}^{L} k_t \left|\cos\left(\mathrm{vec}(\mathbf{W}_{\mathrm{up}}^{i,j}), \mathrm{vec}(\mathbf{W}_{\mathrm{up}}^{t,j})\right)\right|, \tag{9}$$

$\mathbf{W}_{\mathrm{up}}^{i,j}$ and $\mathbf{W}_{\mathrm{up}}^{t,j}$ are the query-LoRA up-projection matrices of the current and historical experts at layer j, respectively. $\cos(\cdot,\cdot)$ denotes the cosine similarity function. Since all historical matrices are frozen, this term only updates the current expert A_i at the i-th task learning. Consequently, residual fusion encourages knowledge reuse, whereas the orthogonality regularizer encourages the current expert to preserve complementary task-specific directions. The overall objective for optimizing the current active expert is defined as:

$$\mathcal{L}_{\mathrm{total}} = \mathcal{L}(A_i, \mathcal{D}_i) + \lambda_{\mathrm{orth}}\mathcal{L}_{\mathrm{orth}}, \tag{10}$$

where λ_{orth} is a hyperparameter that determines the importance of the orthogonality regularization term during the optimization process. Overall, the proposed gradient-fingerprint selection mechanism enables adaptive historical knowledge reuse by retrieving only optimization-compatible experts, while the orthogonality regularization discourages parameter alignment during continual learning. Together, these two components provide a stable and effective foundation for residual expert composition under continually evolving adversarial attacks.

3.4 DUAL-CHANNEL ADAPTIVE PERTURBATION AUGMENTATION

Even after retrieving optimization-compatible experts, the learned expert is still optimized using adversarial samples generated by a single attack algorithm. Consequently, the resulting expert may become highly specialized to one perturbation trajectory, limiting its transferability to neighboring adversarial regions. To alleviate this issue, we introduce a dual-channel adaptive perturbation augmentation strategy that explicitly enlarges the local adversarial neighborhood surrounding each observed example.

Unlike conventional adversarial augmentation methods that generate all perturbations using a single optimization objective, our approach simultaneously explores two complementary perturbation directions. The first channel focuses on attack-specific vulnerability inherited from the current and transferable historical experts, while the second channel explores more universal perturbation directions jointly revealed by the clean base expert and the current task expert. Together, these two channels provide complementary local exploration around the observed adversarial distribution.

Given a data batch $\mathbf{D}$ of size B, the number of augmented samples is defined as $N_{\text{pert}} = \text{round}(mB)$, where $m \in [0, 1]$ is the total augmentation ratio and $\text{round}(\cdot)$ is the random number generation function. These samples are further divided into two groups:

$$N_{\text{univ}} = N_{\text{pert}} - N_{\text{spec}}, N_{\text{keep}} = B - N_{\text{pert}}, N_{\text{spec}} = \text{round}(\rho N_{\text{pert}}) \tag{11}$$

where ρ controls the relative proportion of the source-specific channel and N_{univ} denotes the number of samples allocated to the general channel. N_{keep} denotes the number of samples for which the current adversarial examples are kept unchanged, without adding further perturbations. Instead of adopting a fixed perturbation radius, we further adjust the augmentation strength according to the difficulty of the current task measured by the frozen base expert. The perturbation radius is then computed as:

$$\epsilon = \phi \times \text{scale}\,, \text{scale} = \text{clip}\left(1 - \frac{\mathcal{L}(A_{\text{base}}, \mathbf{D})}{\log C}, \text{scale}_{\min}, 1\right), \tag{12}$$

where ϵ denotes the specific perturbation radius actually used in the current epoch and clip is a function that clamps the values in an array. $\mathcal{L}(A_{\text{base}}, \mathbf{D})$ is the average classification loss of the frozen base expert on the current adversarial batch, C is the number of classes, and ϕ is the maximum perturbation radius. $\text{scale}_{\min}$ denotes the non-zero lower bound for dynamic scaling. Even when the current attack is strong, a minimal incremental perturbation is retained rather than completely disabling the augmentation. A larger base loss $\mathcal{L}(A_{\text{base}}, \mathbf{D})$ leads to a smaller augmentation radius, preventing already difficult adversarial examples from being moved excessively far from the observed distribution.

Source-Specific Channel. The first augmentation channel explores perturbation directions associated with individual attack experts. Instead of relying solely on the current expert, perturbation gradients may also originate from selected historical experts whose transferability has been verified by the proposed gradient-fingerprint selection strategy. Consequently, the generated perturbations preserve attack-specific characteristics while simultaneously inheriting transferable adversarial knowledge accumulated during previous tasks. Starting from $\mathbf{x}^0 = \mathbf{x}$, the perturbation is iteratively updated as:

$$\mathbf{x}^{q+1} = \text{clip}_{[0,1]}\left\{\Pi_{[\mathbf{x}-\epsilon, \mathbf{x}+\epsilon]}\left[\mathbf{x}^q + \frac{\epsilon}{Q}\,\text{sign}\left(\nabla_{\mathbf{x}}\mathcal{L}(A_s, \mathbf{x}^q, \mathbf{y})\right)\right]\right\}, \tag{13}$$

Here q indexes the Q perturbation steps, $\text{sign}(\cdot)$ acts elementwise, and Π projects onto the ϵ-bounded perturbation set. The source pool Ω contains the current expert and at most five historical configurations, selected independently of $\mathcal{S}_i$. Historical sources are prioritized by decreasing first-stage weights k_t, followed by the remaining experts in historical order. We allocate $a_s = N_{\text{spec}} w_s / \sum_{u\in\Omega} w_u$ and $N_s = \lfloor a_s \rfloor + \delta_s$, where the largest-remainder rule assigns $\delta_s \in \{0, 1\}$ so that $\sum_{s\in\Omega} N_s = N_{\text{spec}}$. The current source has weight 1.0; historical sources use k_t above the numerical tolerance 10^{-8} and 0.02 otherwise. This design emphasizes the current attack while preserving limited rehearsal of previously observed adversarial directions.

Universal Perturbation Channel. While the source-specific channel focuses on preserving attack-dependent characteristics, it still follows optimization trajectories associated with individual experts. To further improve perturbation diversity, the second channel explores more universal vulnerability

Table 1: The average classification accuracy of various models on CIFAR10, CIFAR100, CUB200, and ImageNet-R.

Method	CIFAR10					CIFAR100				
	None	FGSM	PGD	CW	AutoAttack	None	FGSM	PGD	CW	AutoAttack
DER	96.02±0.42	43.90±4.99	0.19±0.13	9.57±4.31	0.00±0.00	84.36±1.70	8.33±4.12	0.04±0.06	0.39±0.44	0.00±0.00
DER++	95.66±0.77	45.85±8.72	0.13±0.13	10.99±3.95	0.00±0.00	85.19±4.34	10.49±6.69	0.03±0.04	0.41±0.18	0.00±0.00
DER+++refresh	94.97±0.58	73.91±3.35	2.68±1.05	26.10±6.70	0.01±0.01	80.34±1.10	**74.19±6.55**	0.55±0.44	1.24±0.00	0.43±0.61
DEF	91.25±0.27	68.36±7.93	12.14±4.32	37.88±2.36	0.07±0.04	76.08±1.98	69.58±19.77	0.76±0.17	6.18±5.57	0.20±0.23
SEMA	93.28±0.70	18.47±0.51	0.00±0.00	0.03±0.05	0.00±0.00	54.64±2.22	4.19±0.08	0.00±0.00	0.21±0.04	0.00±0.00
IGAT	69.87±0.28	37.94±0.57	21.04±0.41	42.08±0.11	44.96±0.21	34.44±3.37	15.74±2.12	**11.65±0.75**	30.78±1.27	**30.90±0.23**
AIR	76.65±0.61	28.20±0.78	15.41±1.10	66.58±0.38	12.36±0.94	50.65±6.83	9.21±7.38	5.52±2.85	27.52±20.68	3.96±2.44
DER++AFLC	87.03±0.41	36.16±1.00	12.83±0.28	25.93±0.54	1.38±0.06	73.41±1.56	16.23±1.57	2.83±0.54	2.13±0.30	0.41±0.04
AAOpt	75.13±0.35	38.89±0.24	**27.13±2.40**	69.16±0.42	27.31±1.42	51.15±1.71	14.35±1.06	10.14±1.56	40.98±2.04	6.62±1.44
PDFK	95.51±0.18	46.30±3.14	0.52±0.07	12.04±2.29	0.00±0.00	85.63±0.44	18.16±4.16	0.18±0.19	0.98±0.19	0.00±0.00
DRIFT	**97.81±0.35**	**74.30±4.20**	0.90±1.00	6.98±2.71	0.13±0.01	87.10±1.66	31.27±6.40	0.15±0.13	7.76±1.50	0.76±0.20
CL-LORA	97.77±0.11	51.31±0.60	13.17±1.61	**81.11±5.33**	**61.51±3.78**	**88.82±0.34**	21.29±0.27	0.33±0.11	**66.62±0.70**	10.95±0.09
OURS	**97.32±0.12**	**77.01±1.16**	**75.08±3.92**	**84.67±1.94**	**84.20±0.79**	**88.07±0.15**	**61.56±0.76**	**60.26±0.58**	**80.41±0.63**	**73.30±1.80**

Method	CUB200					ImageNet-R				
	None	FGSM	PGD	CW	AutoAttack	None	FGSM	PGD	CW	AutoAttack
DER	81.09±0.67	43.27±2.86	4.52±0.70	7.56±1.81	3.21±0.66	53.52±1.27	43.02±3.64	15.97±2.38	23.95±4.28	11.45±2.19
DER++	82.15±0.23	42.67±0.77	4.22±0.54	8.04±0.11	3.08±0.15	55.32±1.43	46.57±3.85	19.88±2.46	30.17±1.82	14.38±1.12
DER+++refresh	82.31±1.28	53.12±1.62	20.51±0.96	29.50±1.87	18.96±1.51	60.98±0.97	46.65±1.51	31.08±1.48	39.73±2.62	24.87±2.07
DEF	80.72±0.69	52.70±1.17	22.68±0.66	34.24±1.39	20.44±0.81	64.22±1.35	45.78±2.64	**29.40±2.12**	43.22±2.29	24.48±1.47
SEMA	**82.45±0.94**	8.35±0.31	0.00±0.00	0.02±0.03	0.00±0.00	65.36±1.73	6.25±0.48	0.02±0.03	0.03±0.04	0.01±0.02
IGAT	33.56±2.73	22.13±1.66	11.78±0.72	26.50±1.06	**27.32±0.29**	23.91±3.42	16.37±2.18	8.64±0.95	20.85±1.37	21.46±0.53
AIR	40.11±4.72	**54.28±3.43**	**26.32±3.18**	26.34±11.27	15.53±3.24	32.85±5.18	10.38±6.24	3.95±2.73	2.77±1.86	1.78±1.35
DER++AFLC	80.61±0.83	38.68±1.15	23.58±0.68	32.38±0.79	11.49±0.38	60.34±1.28	33.58±1.73	29.38±0.97	39.57±1.15	14.83±0.58
AAOpt	75.78±0.59	40.43±0.37	22.71±2.03	36.36±1.89	20.10±1.38	60.37±0.91	34.26±0.67	25.43±2.51	39.58±2.14	22.65±1.82
PDFK	80.38±0.36	32.38±3.49	0.74±0.12	5.16±1.13	0.00±0.00	53.05±0.52	33.18±3.94	3.05±0.24	17.28±1.62	0.00±0.00
DRIFT	81.33±0.64	43.68±2.07	9.24±0.56	15.19±1.23	7.93±0.71	58.12±1.00	**51.19±2.95**	13.46±1.65	26.25±2.34	9.99±1.36
CL-LORA	82.34±0.05	47.15±0.10	5.38±0.01	**56.36±0.37**	20.12±0.78	**67.44±0.01**	50.68±0.24	22.50±0.02	**54.07±0.28**	**31.05±0.12**
OURS	**82.59±0.03**	**51.93±0.35**	**22.89±1.51**	**61.37±0.97**	**28.53±1.73**	**67.42±0.05**	**47.84±0.06**	**36.60±0.13**	**54.93±0.13**	**41.30±0.02**

directions shared between the clean model and the current task. At perturbation step q, we first compute the gradient vectors using A_{base} and A_i:

$$\mathbf{g}^q_{\text{base}} = \nabla_{\mathbf{x}}\mathcal{L}(A_{\text{base}}, \mathbf{x}^q, \mathbf{y}), \qquad \mathbf{g}^q_{\text{cur}} = \nabla_{\mathbf{x}}\mathcal{L}(A_i, \mathbf{x}^q, \mathbf{y}), \tag{14}$$

Based on the gradient information, the update direction is constructed as:

$$\mathbf{d}^q = \text{normalize}\left[(1-\beta)\,\text{sign}(\mathbf{g}^q_{\text{base}}) + \beta\,\text{sign}(\mathbf{g}^q_{\text{cur}}) + \boldsymbol{\xi}^q\right], \tag{15}$$

where $\beta \in [0,1]$ balances clean-distribution sensitivity and current-task vulnerability, and $\boldsymbol{\xi}^q$ denotes a small random smoothing term. $\text{sign}(\cdot)$ is a mathematical function that extracts the sign of a real number and normalize denotes the L2 norm. For each augmented sample $\mathbf{x}_n$, an individual perturbation radius is randomly sampled $R_n \sim \mathcal{U}[R_{\min}, R_{\max}]$, where $R_{\max} \leq \epsilon$. A random initial offset is drawn as $\mathbf{u}_n \sim \mathcal{U}[-R_n, R_n]$, leading to $\mathbf{x}^0_n = \text{clip}_{[0,1]}(\mathbf{x}_n + \mathbf{u}_n)$. The subsequent update follows the iterative optimization process.

$$\mathbf{x}^{q+1}_n = \text{clip}_{[0,1]}\left\{\Pi_{|\mathbf{x}^{q+1}_n - \mathbf{x}^q_n|_\infty \leq R_n}\left[\mathbf{x}^q_n + \frac{R_n}{Q}\mathbf{d}^q_n\right]\right\}, \tag{16}$$

where $\mathbf{d}^q_n$ is the update direction of $\mathbf{x}_n$, calculated using Eq. (15). The combination of random radii, random initialization, and dynamically recomputed mixed gradients encourages the model to explore a broader local neighborhood around each observed adversarial example instead of repeatedly following a single deterministic perturbation trajectory. Together, the two channels distribute training examples across source-specific and mixed-gradient perturbation distributions. Their support properties are characterized in Appendix A, while their effects on continual robustness and unseen-attack performance are evaluated empirically.

4 Theoretical Analysis

We analyze the preservation of frozen historical configurations and the conditions under which gradient fingerprints support beneficial residual composition. Detailed assumptions, proofs, and additional results are provided in Appendix A.

4.1 Residual Composition and Catastrophic Forgetting

Proposition 4.1 (Zero Direct Interference with Frozen Configurations). *Let ℓ_t be a loss evaluated through the retained configuration of task $\mathcal{T}_t$ on fixed data. Suppose its backbone, expert parameters, composition coefficients, evaluation state, and class prototypes remain fixed. Then, for parameters Θ_i introduced at any later task, we have*

$$\frac{\partial \ell_t}{\partial \Theta_i} = \mathbf{0}, \qquad i > t. \tag{17}$$

Thus subsequent optimization leaves this configuration's predictions unchanged on fixed inputs.

Remark. The result guarantees preservation of a frozen configuration, rather than zero forgetting of the complete classifier. New configurations can change the maximum competing-class score without altering historical branches. Appendix A.2 provides the proof and the corresponding condition for preserving the final prediction.

4.2 Gradient-Fingerprint Selection as a Transfer Bound

We analyze the immediate effect of adding a frozen historical expert.

Proposition 4.2 (Conditional Transfer Benefit of Expert Composition). *Let $\Psi(\alpha)$ denote the current-task loss when all query/value residuals of A_t have coefficient α, with other quantities fixed. Set $a = \mathbf{A}_{t\to i} \neq 0$ and $b = \mathbf{B}_t \neq 0$. If Ψ' is K-Lipschitz on $[0, k_t]$, then, for $\gamma \geq 0$,*

$$\Delta\mathcal{L}_i^{(t)} := \Psi(0) - \Psi(k_t) \geq k_t\big(\gamma\|a\|_2\|b\|_2 s_t - \mathcal{E}_{t\to i}(\gamma)\big) - \frac{K k_t^2}{2}. \tag{18}$$

The discrepancy $\mathcal{E}_{t\to i}(\gamma)$ accounts for the difference between the fingerprint and the actual composition direction, including residual sites outside the fingerprint. Its definition and the proof are given in Appendix A.3. A positive lower bound certifies beneficial immediate composition. Similarity alone is insufficient; the discrepancy and curvature terms must also be controlled. This result provides a conditional interpretation of the selection statistic without assuming that the trained residual equals a terminal gradient.

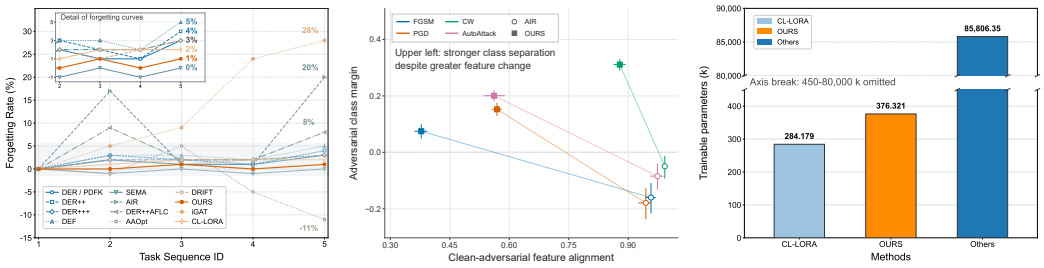
5 Experiments

We conduct experiments on four benchmarks: CIFAR-10 and CIFAR-100 Krizhevsky et al. (2009), CUB-200 Wah et al. (2011), and ImageNet-R Hendrycks et al. (2021). These datasets differ in class granularity and visual characteristics, enabling evaluation across a range of continual learning settings. For all datasets, we use a Vision Transformer (ViT) Dosovitskiy et al. (2020) pretrained on ImageNet-21K and subsequently fine-tuned on ImageNet-1K as the backbone. We provide the detailed experiment settings in **Appendix** D.

5.1 Comparison with SOTA

In this section, we evaluate our method against state-of-the-art continual learning and adversarial defense models, with a focus on experience replay-based methods (e.g., DER Buzzega et al. (2020), DER++ Buzzega et al. (2020), DER+++refresh Wang et al. (2024)), regularization-based methods (e.g., PDFK Zhou et al. (2026)), dynamic architecture-based methods (e.g., SEMA Wang et al. (2025a)), parameter-efficient adaptation methods (e.g., CL-LoRA He et al. (2025)), and robustness-aware approaches (e.g., DEF Buzzega et al. (2020), IGAT Deng & Mu (2023), AIR Zhou & Hua (2024), DER++AFLC Mi et al. (2025), AAOpt Wesego & Rooshenas (2025), DRIFT Guesmi & Shafique (2026)).

Main results. Table 1 reports final classification accuracies, while Fig. 3a presents clean-task forgetting rates on CIFAR-10. Our framework achieves the best overall adversarial robustness with low forgetting, ranking first in 12 of the 16 dataset-attack settings. Averaged over four attacks, it achieves 80.24%, 68.88%, 41.18%, and 45.17% accuracy on CIFAR-10, CIFAR-100, CUB-200, and ImageNet-R, respectively, outperforming all baselines. Under PGD, our method reaches 75.08% and 60.26% on CIFAR-10 and CIFAR-100, compared with the best baseline results of 27.13% and 11.65%. This advantage extends to AutoAttack, where our method reaches 73.30% on CIFAR-100 versus 30.90% for IGAT. It also retains higher final accuracy on clean samples (88.07% versus 34.44%), indicating a more favorable balance between clean and adversarial performance. On CIFAR-10, clean-task forgetting remains within 0%–1% throughout adaptation and ends at 1% after



(a) Forgetting Curve (b) Feature Geometry (c) Peak Trainable Parameters

Figure 3: (a): Forgetting rates on CIFAR-10 over the clean, FGSM, PGD, CW, and AutoAttack task sequence. (b): Feature alignment and class separation on CIFAR-10 under four attacks. Colors denote attacks; lines pair AIR (circles) with OURS (squares). The x-axis measures clean–adversarial feature cosine similarity; the y-axis measures an adversarial feature's cosine similarity to its true-class prototype minus the maximum across other-class prototypes. Points show means over 1,000 shared, randomly sampled test images (100/class); prototypes are normalized mean feature vectors from 200 separate clean images per class. (c): Peak trainable parameter counts on CIFAR-10, in thousands (k). Models with identical parameter counts are grouped as Others; the broken y-axis shows both the lightweight models and the full-backbone group.

learning all four adversarial tasks, compared with 2%, 20%, and 28% for CL-LoRA, AIR, and IGAT, respectively.

Feature distribution and Trainable parameters. Fig. 3b compares feature geometry between our framework and AIR under four attacks. Compared with the closely spaced class representations in AIR, Our method shows clearer class separation and larger inter-class distances. This implies that successfully attacking our framework requires a higher perturbation cost and stronger attacks, indicating our model's stronger robustness and explaining its consistent performance across varying attack strengths. Fig. 3c compares trainable parameters across methods. Our framework maintains parameter efficiency with only $\sim$92K more parameters than CL-LoRA, while achieving stronger attack adaptation and a better parameter efficiency–robustness trade-off than full-parameter methods.

5.2 Ablation Results

Table 2 demonstrates the contributions of historical knowledge reuse and complementary perturbation augmentation. Incorporating LREERC increases PGD accuracy from 14.77% to 36.32% and average adversarial accuracy from 50.21% to 56.61%. The larger gain on PGD than on the other attacks suggests that the benefit of historical knowledge reuse depends on the attack distribution. This improvement is consistent with transferring historical FGSM knowledge to the related PGD attack, highlighting the value of selective knowledge reuse during continual adaptation.

Table 2: Component ablation study. Final accuracies (%) are reported, with the best result in each column highlighted in bold.

Model Configuration	Clean	FGSM	PGD	CW	AutoAttack
LoRA Baseline	97.87	51.91	14.77	76.43	57.73
LoRA + LREERC	**98.01**	49.44	36.32	80.46	60.22
LoRA + LREERC + UPC	97.88	63.17	53.84	81.47	63.35
LoRA + LREERC + SSC	97.65	55.02	53.78	86.23	81.08
LoRA + LREERC + DCAPA	97.44	**75.85**	**78.99**	**86.60**	**84.98**

Universal Perturbation Channel (UPC) and Source-Specific Channel (SSC) capture shared vulnerabilities and expert-specific perturbation directions, respectively. UPC performs better on FGSM, whereas SSC achieves higher accuracy on CW and AutoAttack, suggesting that the channels provide distinct benefits across the evaluated attack distributions. Individually, they achieve PGD accuracies of 53.84% and 53.78%, while their combination reaches 78.99%, exceeding the additive reference of 71.30% and suggesting a superadditive interaction. Removing UPC or SSC decreases the average accuracy across four attacks from 81.61% to 69.03% or 65.46%, respectively, supporting the complementary contributions of both channels to the observed robustness.

6 Conclusion

This paper proposes LREERC for parameter-efficient continual adversarial defense, combining low-rank expert expansion with knowledge reuse through residual composition. Gradient-fingerprint selection guides this reuse by identifying optimization-compatible historical experts, while probability-weighted orthogonality regularization encourages complementary representations.

AI USE STATEMENT

Generative AI tools were used for language editing. All technical claims, implementations, experiments, and reported numbers were verified by the authors.

REFERENCES

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

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)*, pp. 9275–9284, June 2022.

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.

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

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 39–57, 2017. doi: 10.1109/SP.2017.49. URL https://arxiv.org/abs/1608.04644.

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.

Jie Chu, Pei Liu, Tong Su, Yunpeng Wu, Yu Song, and Zenglin Shi. Random dense knowledge distillation for continual learning. *ACM Trans. Multimedia Comput. Commun. Appl.*, 22(6), June 2026. ISSN 1551-6857. doi: 10.1145/3816731. URL https://doi.org/10.1145/3816731.

Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 2206–2216. PMLR, 2020. URL https://proceedings.mlr.press/v119/croce20b.html.

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.

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.

Yian Deng and Tingting Mu. Understanding and improving ensemble adversarial defense. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 58075–58087. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/b589d92785e39486e978fa273d0dc343-Paper-Conference.pdf.

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

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

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.

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.

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.

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

Amira Guesmi and Muhammad Shafique. Drift: Divergent response in filtered transformations for robust adversarial defense. In *International Conference on Learning Representations*, volume 2026, pp. 86810–86829, 2026.

Mahedi Hasan, Shoaib Meraj Sami, and Nasser Nasrabadi. Clface: A scalable and resource-efficient continual learning framework for lifelong face recognition. In *Proceedings of the Winter Conference on Applications of Computer Vision*, pp. 5082–5091, 2025.

Soufiane Hayou, Nikhil Ghosh, and Bin Yu. LoRA+: Efficient low rank adaptation of large models. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 17783–17806. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/hayou24a.html.

Jiangpeng He, Zhihao Duan, and Fengqing Zhu. Cl-lora: Continual low-rank adaptation for rehearsal-free class-incremental learning. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 30534–30544. IEEE, 2025.

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*, pp. 8340–8349, 2021.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

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, pp. 12698–12707, 2024.

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.

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*, pp. 13398–13408, 2022.

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)*, pp. 3842–3846. IEEE, 2019.

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

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Inf. Proc. Systems (NIPS)*, pp. 1097–1105, 2012.

Yan-Shuo Liang and Wu-Jun Li. Inflora: Interference-free low-rank adaptation for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23638–23647, June 2024.

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

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *stat*, 1050(9), 2017.

James Martens and Roger B. Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In Francis R. Bach and David M. Blei (eds.), *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*, pp. 2408–2417. JMLR.org, 2015. URL http://proceedings.mlr.press/v37/martens15.html.

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*, pp. 562–572, 2025.

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.

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.

Pingbo Pan, Siddharth Swaroop, Alexander Immer, Runa Eschenhagen, Richard Turner, and Mohammad Emtiyaz E Khan. Continual deep learning by functional regularisation of memorable past. In *Advances in Neural Information Processing Systems*, volume 33, pp. 4453–4464, 2020.

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

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.

Steffen Schotthöfer, Emanuele Zangrando, Gianluca Ceruti, Francesco Tudisco, and Jonas Kusch. Geolora: Geometric integration for parameter efficient fine-tuning. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 49852–49881, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/7ccae68f81c1bd9104c304a9d8967048-Paper-Conference.pdf.

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*, pp. 16674–16683, 2021.

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

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)*, pp. 99–108, June 2022.

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

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*, pp. 7658–7667, 2021a.

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*, pp. 10087–10098, 2025a.

Ruohan Wang, Marco Ciccone, Massimiliano Pontil, and Carlo Ciliberto. Schedule-robust continual learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–13, 2025b. doi: 10.1109/TPAMI.2025.3614868.

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*, pp. 184–193, 2021b.

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.

Zhengbo Wang, Jian Liang, Ran He, Zilei Wang, and Tieniu Tan. Lora-pro: Are low-rank adapters properly optimized? In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 93787–93808, 2025c. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/ea184f920a0f0f8d8030aa1bd7ac9fd4-Paper-Conference.pdf.

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

Daniel Wesego and Pedram Rooshenas. Adversary aware optimization for robust defense. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, pp. 38556–38582. Curran Associates, Inc., 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/3714a5dd84368335b36920d289ea9912-Paper-Conference.pdf.

Daniel Wesego and Pedram Rooshenas. Adversary aware optimization for robust defense. *Advances in Neural Information Processing Systems*, 38:38556–38582, 2026.

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

Hongwei Yan, Liyuan Wang, Kaisheng Ma, and Yi Zhong. Orchestrate latent expertise: Advancing online continual learning with multi-level supervision and reverse self-distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23670–23680, 2024.

Fei Ye, Yulong Zhao, Qihe Liu, Junlin Chen, Adrian G Bors, Rongyao Hu, et al. Dynamic siamese expansion framework for improving robustness in online continual learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Jui-Nan Yen, Si Si, Zhao Meng, Felix Yu, Venkata Sai Surya Subramanyam Duvvuri, Inderjit Dhillon, Cho-Jui Hsieh, and Sanjiv Kumar. Lora done rite: Robust invariant transformation equilibration for lora optimization. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 75815–75834, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/bcbc0f660d2dde42f9d1d0ecb14a6f9a-Paper-Conference.pdf.

F. Zenke, B. Poole, and S. Ganguli. Continual learning through synaptic intelligence. In *Proc. of Int. Conf. on Machine Learning, vol. PLMR 70*, pp. 3987–3995, 2017.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adaptive budget allocation for parameter-efficient fine-tuning. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=lq62uWRJjiY.

Dulan Zhou, Zijian Gao, and Kele Xu. Perturbing to preserve: Defending fragile knowledge in online continual learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 28937–28945, 2026.

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*, pp. 24263–24272, 2024.

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.

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*, pp. 16443–16452, 2021.

A Additional Analysis of the Theoretical Framework

A.1 Notation and Scope

The additive expression in Eq. equation 1 applies to query and value projections. The complete Transformer representation remains nonlinear. A historical residual refers to the expert's own LoRA modules, whereas a retained configuration includes the backbone, the clean base expert, its own modules, and its saved historical composition.

The fingerprint parameter is the query LoRA up-projection in the last Transformer block. Reference and query fingerprints are computed on historical and current adversarial subsets, respectively. They record gradients at retained parameter values rather than complete optimization trajectories. The notation $\mathcal{L}(A_t, \mathcal{D})$ denotes classification loss through the corresponding expert configuration and probe classifier; the prediction inside the cross-entropy expression is a class-probability vector.

Exact cosine statements require nonzero gradients. Numerical computation uses a positive normalization floor and assigns zero selection scores to zero or non-finite fingerprints. The exact identities below apply when the relevant normalization floors are inactive.

A.2 Proof of Proposition 4.1

Proof. Let $h_t(\mathbf{x})$ be the feature map of the configuration retained after task $\mathcal{T}_t$. It depends only on the frozen backbone, the clean base expert, the completed expert, and the historical residuals and coefficients saved for that configuration. No subsequently introduced parameter Θ_i, $i > t$, enters this computation graph. Therefore,

$$\frac{\partial h_t(\mathbf{x})}{\partial \Theta_i} = \mathbf{0}.$$

Let

$$\sigma_{t,c}(\mathbf{x}) = \mathrm{nrm}(h_t(\mathbf{x}))^\top \mathrm{nrm}(p_{t,c}),$$

where nrm denotes the stabilized normalization used by the classifier. Because the prototypes and evaluation state are also fixed, $\sigma_{t,c}(\mathbf{x})$ is invariant to changes in Θ_i. The chain rule then gives $\partial \ell_t / \partial \Theta_i = \mathbf{0}$ for a loss evaluated solely through this retained configuration on fixed data. □

From branch preservation to the final decision. The complete classifier takes a class-wise maximum over retained configurations. For a fixed labeled input, let T and V be the old collection's maximum correct-class and incorrect-class scores. Let t and v be the corresponding maxima contributed by newly added configurations. Then

$$M_{\mathrm{new}} - M_{\mathrm{old}} = [t - T]_+ - [v - V]_+, \qquad [u]_+ = \max\{u, 0\}. \tag{19}$$

Indeed, $M_{\mathrm{old}} = T - V$ and $M_{\mathrm{new}} = \max\{T, t\} - \max\{V, v\}$.

Consequently,

$$M_{\mathrm{new}} \geq M_{\mathrm{old}} - [v - V]_+.$$

The condition $M_{\mathrm{old}} > [v - V]_+$ is sufficient to retain a correct prediction. A corresponding robust statement requires this condition to hold throughout the evaluated perturbation set.

For example, old correct-class and incorrect-class maxima of 0.8 and 0.2 yield a correct prediction. A new configuration contributing 0.1 and 0.9 changes the final decision even though all historical scores are preserved. Thus zero direct parameter interference does not imply zero forgetting of the complete classifier.

A.3 Proof of Proposition 4.2

Actual residual insertion. Fix the current adversarial subset $\mathcal{D}_i^s$, the current classification head, all model parameters, and all composition coefficients other than the coefficient of candidate expert A_t. For its query and value injection sites, write

$$U_t^{p,j} = W_{\text{up}}^{t,p,j}, \qquad D_t^{p,j} = W_{\text{down}}^{t,p,j}, \qquad p \in \{Q, V\}, \quad j = 1, \ldots, L.$$

In the implementation, $L = 12$.

Introduce auxiliary up-projection slots $Z = \{Z^{p,j}\}_{p,j}$. At each site, insert

$$Z^{p,j} D_t^{p,j} h_j(Z)$$

into the corresponding query or value projection. The down-projections remain fixed. The hidden states $h_j(Z)$ are generated by the complete network, so the construction includes changes in attention and all downstream representations.

Let $f_i(Z)$ be the resulting classification loss on $\mathcal{D}_i^s$ and define

$$\Psi(\alpha) = f_i(\{\alpha U_t^{p,j}\}_{p,j}).$$

At $\alpha = 0$, the candidate's direct residual contribution is absent. At $\alpha = k_t$, its actual weighted residual is inserted at all query and value sites. Other contributions remain fixed. Hence

$$\Delta\mathcal{L}_i^{(t)} = \Psi(0) - \Psi(k_t)$$

measures the immediate gain from actual residual composition, not from a hypothetical parameter update.

Exact directional derivative. At the composition center, define

$$G^{p,j} = \nabla_{Z^{p,j}} f_i(Z)|_{Z=0}.$$

The chain rule through the complete computational graph yields

$$\Psi'(0) = \sum_{j=1}^{L} \sum_{p \in \{Q,V\}} \langle G^{p,j}, U_t^{p,j} \rangle_{\text{F}}. \tag{20}$$

This identity does not assume a linear Transformer.

The fingerprint represents the last query up-projection. Write

$$u = \text{vec}(U_t^{Q,L}), \qquad g = \text{vec}(G^{Q,L}),$$

and collect the other sites into

$$R_{\text{other}} = \sum_{(p,j) \neq (Q,L)} \langle G^{p,j}, U_t^{p,j} \rangle_{\text{F}}.$$

Thus

$$\Psi'(0) = g^\top u + R_{\text{other}}.$$

For $L = 12$, the remainder contains the other 23 query and value injection sites.

Connecting the actual residual to the fingerprints. Let $a = \mathbf{A}_{t \to i}$ and $b = \mathbf{B}_t$. Both vectors use the coordinates of the last query up-projection. The probe gradient a and the insertion gradient g are evaluated in different model configurations: the historical probe configuration and the receiving composition center.

For any $\gamma \geq 0$, define

$$e_\gamma = u + \gamma b, \qquad d = g - a,$$

and

$$\mathcal{E}_{t \to i}(\gamma) = \|a\|_2 \|e_\gamma\|_2 + \|d\|_2 \|u\|_2 + [R_{\text{other}}]_+. \tag{21}$$

The first term measures the difference between the trained up-projection and a scaled negative reference-gradient direction. The second measures the probe-to-composition gradient difference. The third retains potentially loss-increasing contributions from sites not represented by the fingerprint. No equality between a trained parameter and its terminal gradient is assumed.

Proof of Proposition 4.2. Using $u = -\gamma b + e_\gamma$ and $g = a + d$,

$$\begin{aligned}\Psi'(0) &= a^\top u + d^\top u + R_{\text{other}} \\ &= -\gamma a^\top b + a^\top e_\gamma + d^\top u + R_{\text{other}} \\ &\le -\gamma \|a\|_2 \|b\|_2 s_t + \mathcal{E}_{t\to i}(\gamma).\end{aligned}$$

The inequality follows from Cauchy–Schwarz and $R_{\text{other}} \le [R_{\text{other}}]_+$.

The assumed K-Lipschitz continuity of Ψ' on $[0, k_t]$ gives

$$\begin{aligned}\Psi(k_t) - \Psi(0) &= \int_0^{k_t} \Psi'(\alpha)\,\mathrm{d}\alpha \\ &\le k_t \Psi'(0) + \frac{K k_t^2}{2}.\end{aligned}$$

Substitution and rearrangement prove Eq. equation 18. □

Interpretation and scope. Beneficial immediate composition is certified when

$$\gamma \|a\|_2 \|b\|_2 s_t > \mathcal{E}_{t\to i}(\gamma) + \frac{K k_t}{2}.$$

The scale γ is an analytical quantity rather than a training hyperparameter. For example, the choice minimizing $\|u + \gamma b\|_2$ over $\gamma \ge 0$ is

$$\gamma^\star = \max\left\{0, -\frac{u^\top b}{\|b\|_2^2}\right\}.$$

This choice does not guarantee small remaining discrepancies.

If $u = -\gamma b$, $g = a$, and $R_{\text{other}} \le 0$, the discrepancy vanishes and the gradient-alignment term can dominate the curvature penalty. These conditions do not follow from storing a terminal gradient. The general bound explicitly accounts for their failure.

The discrepancy terms can be examined using frozen parameters and derivatives at the composition center. A rigorous finite-coefficient certificate additionally requires a valid curvature bound over the entire interval $[0, k_t]$. The reported accuracy results are not taken as verification of these sufficient conditions.

The proposition concerns immediate empirical loss change under actual residual insertion. It does not guarantee gains after further optimization or robustness of the final classifier. Taking expectations preserves the conditional inequality when its terms are integrable, but does not yield an unconditional proportionality between expected transfer gain and s_t.

A.4 Relative Selection and Bounded Historical Residual Reuse

Two-stage selection. The configured rule uses `fusion_gate=relative`, with $\tau = 0.1$, $r = 0.6$, and $K_{\text{res}} = 5$. The historical pool $\mathcal{H}_i = \{2, \dots, i-1\}$ contains completed adversarial experts and excludes the clean base and current expert. The first gate retains $\mathcal{V}_i = \{t \in \mathcal{H}_i : s_t > \tau\}$. Missing, non-finite, or non-positive similarities are assigned zero in the implementation. Its additional numerical validity threshold of 10^{-6} does not affect candidates admitted under $\tau = 0.1$.

For nonempty $\mathcal{V}_i$, define

$$s_i^\star = \max_{u \in \mathcal{V}_i} s_u, \qquad q_t = \frac{s_t}{s_i^\star}, \qquad \mathcal{C}_i = \{t \in \mathcal{V}_i : q_t > r\}.$$

Both gates use strict inequalities. Candidates in $\mathcal{C}_i$ are ranked by decreasing similarity, with ties resolved by increasing historical index. The first $\min(K_{\text{res}}, |\mathcal{C}_i|)$ are retained in $\mathcal{S}_i$. The cap therefore limits the number of admitted residuals without filling unused positions with candidates that fail either threshold. If $\mathcal{V}_i = \varnothing$, we set $\mathcal{C}_i = \mathcal{S}_i = \varnothing$.

Selection and weighting serve different roles. The relative score q_t determines admission, whereas k_t determines the contribution of an admitted expert. The coefficients are the unit-temperature softmax weights computed over the entire first-stage pool:

$$k_t = \frac{\exp(s_t)}{\sum_{u \in \mathcal{V}_i} \exp(s_u)}, \qquad t \in \mathcal{V}_i.$$

Coefficients outside $\mathcal{V}_i$ are zero. The retained coefficients are not renormalized after relative selection or truncation. In particular, the implemented score $q_t = s_t/s_i^\star$ differs from $k_t/\max_u k_u = \exp(s_t - s_i^\star)$. Selection is performed once at task initialization and remains fixed during that task.

Nonempty selection and bounded cardinality. Suppose $0 < \tau < 1$, $0 \le r < 1$, and $K_{\text{res}} \ge 1$. Whenever $\mathcal{V}_i \neq \varnothing$, its maximum satisfies $s_i^\star > \tau > 0$. Every maximizer therefore has relative score $q_t = 1 > r$ and belongs to $\mathcal{C}_i$. Retaining the highest-scoring candidates cannot remove every member of this nonempty set. Hence

$$\begin{aligned} \mathcal{S}_i = \varnothing \quad &\Longleftrightarrow \quad \mathcal{V}_i = \varnothing, \\ |\mathcal{S}_i| &= \min\{K_{\text{res}}, |\mathcal{C}_i|\} \le K_{\text{res}}. \end{aligned} \tag{22}$$

With the configured thresholds, a nonempty first-stage pool yields between one and five historical residuals. Zero reuse remains possible when no historical expert passes the absolute-similarity gate. In that case, the historical residual contribution and its associated orthogonality penalty are zero, while the clean base and current expert remain active.

The two admission conditions can equivalently be written as

$$t \in \mathcal{C}_i \quad \Longleftrightarrow \quad t \in \mathcal{H}_i \text{ and } s_t > \max\{\tau, r s_i^\star\}.$$

The absolute threshold requires sufficient alignment, while the relative threshold retains candidates close to the strongest eligible match. Their combination supports selective reuse of accumulated historical knowledge while allowing multiple compatible experts to contribute.

Resolution of probability-induced rejection. Under the alternative probability gate $k_t > r$, let $M_i = |\mathcal{V}_i| \ge 2$. Since $\tau < s_t \le 1$,

$$k_t = \left[1 + \sum_{u \in \mathcal{V}_i \setminus \{t\}} \exp(s_u - s_t)\right]^{-1} < \frac{1}{1 + (M_i - 1)e^{\tau - 1}}.$$

At $\tau = 0.1$, this upper bound is below 0.6 whenever $M_i \ge 3$. Furthermore, coefficients summing to one permit at most one value strictly above 0.6. Applying the second threshold directly to these coefficients can therefore disable residual reuse when several aligned candidates are available.

Relative admission removes this restriction because the scores q_t do not sum to one. For example, similarities $(0.9, 0.8, 0.7)$ yield relative scores $(1, 0.8889, 0.7778)$ and softmax coefficients approximately $(0.3672, 0.3322, 0.3006)$. All three candidates pass the relative threshold $r = 0.6$ and retain their original coefficients, whereas probability thresholding at the same value rejects all three. This algebraic example isolates the difference between the two admission rules.

Candidate growth and composition coefficients. For fixed existing similarities, adding candidates whose scores do not exceed the previous maximum leaves the existing relative scores unchanged. An enlarged softmax denominator therefore cannot by itself cause those candidates to fail the relative gate. The capped set can still change when new candidates outrank retained members, and a new maximum can change the relative reference.

The composition coefficients continue to depend on the entire first-stage pool. Additional candidates can reduce existing k_t even if they fail the relative gate. For M_i equally similar candidates, all relative scores equal one, while each coefficient equals $1/M_i$. The selected set then contains $\min(K_{\text{res}}, M_i)$ experts with total coefficient $\min(K_{\text{res}}, M_i)/M_i$. Thus the correction removes the structural rejection caused by probability thresholding while preserving the original softmax weighting mechanism.

Computational scope and historical knowledge pathways. The cap $K_{\text{res}} = 5$ limits the number of historical LoRA residuals inserted into a new configuration and the number of historical experts entering its orthogonality penalty. Each inserted term uses the selected expert's own LoRA modules, without recursively expanding its saved composition. This bounds the growth of direct residual reuse in longer task sequences. Historical fingerprint evaluation, library storage, and the number of retained prediction configurations remain separate costs.

In the five-task clean–FGSM–PGD–CW–AutoAttack sequence, at most three historical adversarial experts precede the final task, so the cap of five does not truncate the residual candidates. Its computational effect becomes relevant in longer sequences with more eligible historical experts.

The residual set $\mathcal{S}_i$, augmentation-source pool Ω, and prediction collection $\mathcal{E}$ have distinct roles. The independent parameter `perturb_max_hist=5` limits historical augmentation sources. These sources are prioritized by decreasing first-stage coefficients above 10^{-8}, followed by the remaining historical experts in stored order. Their allocation weights are k_t above this tolerance and 0.02 otherwise; the current source has weight 1.0. Passing the relative residual-selection gate is not required for inclusion in this pool.

Historical augmentation and prediction use saved complete configurations, including their original residual compositions. Consequently, excluding an expert from the current residual set does not discard its accumulated knowledge from the other pathways. Historical knowledge can continue to support perturbation generation and prediction without requiring every retained expert to participate in every subsequent residual composition.

A.5 Parameter-Subspace Control by the Orthogonality Penalty

Fix layer j and let $\mathbf{w}_i^j = \mathrm{vec}(\mathbf{W}_{\text{up}}^{i,j})$ and $\mathbf{w}_t^j = \mathrm{vec}(\mathbf{W}_{\text{up}}^{t,j})$ be the query up-projection parameters. The following result includes the scale and historical-frame terms needed for a valid projection bound.

Proposition A.1 (Weighted parameter-subspace bound)**.** *Suppose $\mathcal{S}_i \neq \varnothing$ and $k_t > 0$ for its members. Define*

$$\bar{\mathbf{w}}_i = \frac{\mathbf{w}_i^j}{\max\{\|\mathbf{w}_i^j\|_2, \varepsilon_{\text{n}}\}}, \qquad \mathbf{v}_t = \frac{\mathbf{w}_t^j}{\max\{\|\mathbf{w}_t^j\|_2, \varepsilon_{\text{n}}\}},$$

and

$$G_j = \sum_{t \in \mathcal{S}_i} k_t \mathbf{v}_t \mathbf{v}_t^\top, \qquad \mathcal{W}_j = \mathrm{span}\{\mathbf{v}_t : t \in \mathcal{S}_i\}.$$

If $\mathcal{W}_j \neq \{0\}$, let $\mu_j = \lambda_{\min}^+(G_j) > 0$. Then

$$\begin{aligned} \frac{\|\operatorname{Proj}_{\mathcal{W}_j} \mathbf{w}_i^j\|_2^2}{\max\{\|\mathbf{w}_i^j\|_2, \varepsilon_{\text{n}}\}^2} &\leq \frac{1}{\mu_j} \sum_{t \in \mathcal{S}_i} k_t (\bar{\mathbf{w}}_i^\top \mathbf{v}_t)^2 \\ &\leq \frac{1}{\mu_j} \sum_{t \in \mathcal{S}_i} k_t |\bar{\mathbf{w}}_i^\top \mathbf{v}_t|. \end{aligned} \tag{23}$$

If $\mathcal{W}_j = \{0\}$, the projection is zero.

Proof. Positive weights imply that the range of G_j is $\mathcal{W}_j$. Its spectral decomposition gives

$$z^\top G_j z \geq \mu_j \|\operatorname{Proj}_{\mathcal{W}_j} z\|_2^2.$$

Taking $z = \bar{\mathbf{w}}_i$ and rescaling yields the first inequality. Since both stabilized normalized vectors have norm at most one, $|\bar{\mathbf{w}}_i^\top \mathbf{v}_t| \leq 1$. The inequality $u^2 \leq |u|$ for $|u| \leq 1$ then yields the second bound. □

The result does not require an orthogonal historical basis. Its geometry enters through μ_j, which cannot be omitted. With inactive normalization floors, the left-hand side is the fraction of the current parameter vector lying in the historical span. The absolute-cosine penalty controls an upper bound on this quantity. Absolute-cosine and squared-cosine minimization are not generally equivalent, and a decreasing upper bound does not imply a monotonically decreasing realized projection.

The statement concerns the chosen LoRA parameter coordinates. For any invertible matrix R of compatible size,

$$U \mapsto UR, \qquad D \mapsto R^{-1}D$$

preserves UD but can change the angles between vectorized up-projections. Parameter separation therefore does not independently establish feature-space separation or complementary input sensitivities. The functional conditions used in the margin analysis are stated separately in Appendix B. For $\mathcal{S}_i = \varnothing$, the regularizer is defined as zero.

A.6 Distributional Coverage of Dual-Channel Augmentation

We characterize the support of the generated probability measures. This formulation allows discrete components and boundary mass introduced by clipping, without assuming that every component has a density.

For fixed channel counts with $N_{\text{pert}} > 0$, let μ_{spec} and μ_{univ} denote the marginal laws of labeled examples produced by the two channels. These laws include source allocation and perturbation randomness. The actual mixing ratio for a uniformly selected augmented example is

$$\rho_* = \frac{N_{\text{spec}}}{N_{\text{pert}}},$$

which accounts for the rounding in Eq. equation 11.

Proposition A.2 (Support and loss properties of the mixture)**.** *If* $0 < \rho_* < 1$*, then*

$$\mu_{\text{dual}} = \rho_* \mu_{\text{spec}} + (1 - \rho_*)\mu_{\text{univ}}$$

satisfies

$$\operatorname{supp}(\mu_{\text{dual}}) = \operatorname{supp}(\mu_{\text{spec}}) \cup \operatorname{supp}(\mu_{\text{univ}}). \tag{24}$$

For every nonnegative measurable loss ℓ*,*

$$\begin{aligned} \mathbb{E}_{\mu_{\text{spec}}}[\ell] &\le \frac{\mathbb{E}_{\mu_{\text{dual}}}[\ell]}{\rho_*}, \\ \mathbb{E}_{\mu_{\text{univ}}}[\ell] &\le \frac{\mathbb{E}_{\mu_{\text{dual}}}[\ell]}{1 - \rho_*}. \end{aligned} \tag{25}$$

Proof. Any open neighborhood with positive probability under either component has positive probability under the mixture. If a point lies outside both component supports, each component has a zero-probability open neighborhood of that point. The intersection of these neighborhoods has zero mixture probability. This proves Eq. equation 24.

Linearity of integration gives

$$\mathbb{E}_{\mu_{\text{dual}}}[\ell] = \rho_* \mathbb{E}_{\mu_{\text{spec}}}[\ell] + (1 - \rho_*)\mathbb{E}_{\mu_{\text{univ}}}[\ell].$$

Dropping either nonnegative term proves the respective loss bound. □

Strict enlargement over the source-specific component requires the universal component to contribute points outside that component's support. Additional random inputs, sampled radii, and smoothing noise do not alone guarantee this condition. The result compares the fixed component laws above, rather than an independently reconfigured single-channel baseline.

The proposition requires no independence between channels. It does not establish an increase in differential entropy or coverage of an entire threat region. The loss inequalities apply to the same predictor and loss in every expectation, so they do not directly convert current-expert training loss into final-classifier robustness.

Including the examples that receive no additional perturbation, the full-batch marginal law is

$$\mu_{\text{train}} = \frac{N_{\text{keep}}}{B}\mu_{\text{keep}} + \frac{N_{\text{spec}}}{B}\mu_{\text{spec}} + \frac{N_{\text{univ}}}{B}\mu_{\text{univ}}.$$

The same proof shows that its support is the union of the positive-weight component supports. DCAPA therefore provides exposure to several generated distributions; its benefit on unseen attacks is evaluated empirically.

Corollary A.3 (Bounds on the adaptive radius)**.** *For $C > 1$, $\phi > 0$, and $0 < \mathrm{scale}_{\min} \le 1$, the adaptive schedule satisfies*

$$\phi\,\mathrm{scale}_{\min} \le \epsilon \le \phi,$$

and ϵ is nonincreasing in the measured base-expert loss.

Proof. Both statements follow from the range and monotonicity of the clipping operation applied to $1 - \mathcal{L}(A_{\mathrm{base}}, \mathbf{D})/\log C$. □

The quantity $\log C$ is the cross-entropy of a uniform predictor, not an upper bound on cross-entropy. The radius reaches its lower floor when

$$\mathcal{L}(A_{\mathrm{base}}, \mathbf{D}) \ge (1 - \mathrm{scale}_{\min}) \log C.$$

A positive floor preserves a permitted augmentation budget. It does not ensure a nonzero realized displacement, strict support enlargement, or robustness to an arbitrary unseen attack.

B Additional Theoretical Analysis of The Higher Accuracy under CW and AutoAttack

B.1 The comparison to be explained

Table 1 reports final accuracies of 77.01%, 75.08%, 84.67%, and 84.20% on CIFAR-10 under FGSM, PGD, CW, and AutoAttack, respectively. The corresponding CIFAR-100 results are 61.56%, 60.26%, 80.41%, and 73.30%. Thus, both CW and AutoAttack yield higher accuracy than either FGSM or PGD. We explain how this ordering can arise from the classification geometry learned by LREERC under the evaluation protocol described in the experimental settings.

The relevant distinction is between an attack's optimization objective and the damage its returned examples cause to the learned decision margin. LREERC retains historical representations, selectively reuses their residual corrections, and trains each new expert on a mixture of adversarial neighborhoods. These operations can preserve class evidence on different attack distributions to different degrees. We first quantify that preservation, then connect the training neighborhoods to attack-conditioned error, and finally state sufficient conditions for the observed ordering. The analysis fixes the final model and the clean test population throughout.

B.2 Preservation of the final classification margin

Let $\mathcal{E}$ index the expert configurations retained after the final task. Each configuration retains the frozen base expert and its selected historical residual corrections through Eq. equation 1. As described in Section 3.2, the LoRA modules are introduced into the query and value projections, while the complete Transformer representation $h_e(\mathbf{x})$ remains nonlinear. Write

$$u_e(\mathbf{x}) = \frac{h_e(\mathbf{x})}{\|h_e(\mathbf{x})\|_2}, \qquad v_{e,c} = \frac{p_{e,c}}{\|p_{e,c}\|_2}, \qquad \sigma_{e,c}(\mathbf{x}) = \langle u_e(\mathbf{x}), v_{e,c} \rangle,$$

where $p_{e,c}$ is the class prototype for configuration e. Numerical normalization at zero gives vectors of norm at most one, which is sufficient for the bounds below. The final classifier uses

$$\begin{aligned} z_c(\mathbf{x}) &= \max_{e \in \mathcal{E}} \sigma_{e,c}(\mathbf{x}), \quad M(\mathbf{x}, y) = z_y(\mathbf{x}) - \max_{c \ne y} z_c(\mathbf{x}), \\ V(\mathbf{x}, y) &= \max_{\substack{e \in \mathcal{E} \\ c \ne y}} \sigma_{e,c}(\mathbf{x}), \quad \rho_e(\mathbf{x}, y) = \sigma_{e,y}(\mathbf{x}) - V(\mathbf{x}, y). \end{aligned} \tag{26}$$

Consequently, $M = \max_e \rho_e$. A positive supporting margin ρ_e means that expert e's correct-class score exceeds every incorrect-class score in the collection. Correct classification by that expert in isolation is a weaker condition. We count zero-margin ties as errors in the analysis; when ties have probability zero, $1 - \Pr[M \le 0]$ is the usual classification accuracy.

For a perturbation $\boldsymbol{\delta}$, define

$$d_e(\mathbf{x}, \boldsymbol{\delta}) = \|u_e(\mathbf{x} + \boldsymbol{\delta}) - u_e(\mathbf{x})\|_2, \qquad D(\mathbf{x}, \boldsymbol{\delta}) = \max_e d_e(\mathbf{x}, \boldsymbol{\delta}).$$

Proposition B.1 (Preservation of correct-class evidence)**.** *For fixed expert configurations and prototypes,*

$$M(\mathbf{x}+\boldsymbol{\delta}, y) \geq \mathcal{B}(\mathbf{x}, y, \boldsymbol{\delta}) \\ := \max_{e}\{\rho_e(\mathbf{x}, y) - d_e(\mathbf{x}, \boldsymbol{\delta})\} - D(\mathbf{x}, \boldsymbol{\delta}). \tag{27}$$

In particular, the prediction remains correct whenever some expert satisfies $\rho_e(\mathbf{x}, y) > d_e(\mathbf{x}, \boldsymbol{\delta}) + D(\mathbf{x}, \boldsymbol{\delta})$.

Proof. Cauchy–Schwarz gives $|\sigma_{e,c}(\mathbf{x}+\boldsymbol{\delta}) - \sigma_{e,c}(\mathbf{x})| \leq d_e$. The correct-class score of expert e therefore decreases by at most d_e, and the global incorrect-class maximum increases by at most D. Subtracting and maximizing over e proves the claim. □

This bound identifies the two quantities that matter: the supporting margins before perturbation and their sensitivity to the returned displacement. In particular, $\mathcal{B} > 0$ certifies survival of the prediction, whereas $\mathcal{B} \leq 0$ alone does not establish an error. A simpler consequence is $M(\mathbf{x}+\boldsymbol{\delta}, y) \geq M(\mathbf{x}, y) - 2D$.

For feature norms locally above the numerical normalization floor and a locally Lipschitz Jacobian $J_e = \partial h_e / \partial \mathbf{x}$, normalization gives the expansion

$$u_e(\mathbf{x}+\boldsymbol{\delta}) - u_e(\mathbf{x}) = \frac{(I - u_e u_e^\top) J_e \boldsymbol{\delta}}{\|h_e(\mathbf{x})\|_2} + O(\|\boldsymbol{\delta}\|_2^2). \tag{28}$$

Thus, to first order, only the component perpendicular to the feature vector changes prototype similarities. This explains why input-space distortion and classification damage need not have the same ordering. It does not assume that any particular attack produces predominantly radial changes.

B.3 How continual learning supplies complementary evidence

B.3.1 Retention and selective reuse

Proposition 4.1 preserves each fixed historical configuration. Equation equation 19 characterizes the effect of adding new configurations to the complete classifier: the margin changes according to the balance between additional correct-class and competing-class evidence. Historical branch preservation therefore retains existing support without guaranteeing nonnegative backward transfer of the final prediction rule.

Gradient fingerprints determine which historical corrections enter the new representation. For nonzero gradients in Eq. equation 5, positive s_t means that the current-task and historical reference gradients form an acute angle. This directional compatibility motivates the two-stage selection rule in Section 3.3. Its contribution to the present analysis is assessed through the resulting classification margins. The weighted absolute-cosine penalty in Eq. equation 9 discourages alignment with selected historical parameter directions. To connect this parameter-level objective to robustness, the next proposition explicitly states the required condition on input sensitivities.

Historical experts have three roles: contributing residuals through $\mathcal{S}_i$, supplying perturbation gradients through the source set Ω, and contributing retained configurations to $\mathcal{E}$. These roles follow different selection rules. Appendix A.4 derives the consequences of the stated residual-selection thresholds. In particular, $r = 0.6$ permits at most one historical residual, and three or more candidates passing the $\tau = 0.1$ similarity gate yield an empty residual-composition set. This does not remove historical configurations from the other two knowledge pathways. The margin analysis therefore does not assume simultaneous residual fusion of all retained experts.

B.3.2 The cost of removing complementary support

Parameter separation is useful for robustness when it translates into different input sensitivities of positive supporting margins. The following result makes that additional condition explicit and allows curvature in the local margin functions.

Proposition B.2 (Simultaneous suppression of supporting margins)**.** *Fix an input with K positive supporting margins $\rho_e(\mathbf{x}, y)$, indexed here by $e = 1, \ldots, K$. Suppose that throughout the ball*

$\|\boldsymbol{\delta}\|_2 \le r_0$,

$$\rho_e(\mathbf{x}+\boldsymbol{\delta}, y) \ge \rho_e(\mathbf{x}, y) + a_e^\top \boldsymbol{\delta} - \frac{L_e}{2}\|\boldsymbol{\delta}\|_2^2, \qquad a_e \ne 0, \quad L_e \ge 0.$$

Set $g_e = a_e/\|a_e\|_2$, $b_e = \rho_e(\mathbf{x}, y)/\|a_e\|_2$, and $\eta_e = L_e/(2\|a_e\|_2)$. If $|g_e^\top g_f| \le \mu$ for $e \ne f$, with $0 \le \mu \le 1$, then any perturbation of norm $r \le r_0$ that makes all these margins nonpositive must satisfy

$$\left(\sum_{e=1}^{K} [b_e - \eta_e r^2]_+^2\right)^{1/2} \le \sqrt{1 + (K-1)\mu}\, r. \tag{29}$$

For affine supporting margins, $L_e = 0$, this reduces to

$$\|\boldsymbol{\delta}\|_2 \ge \frac{(\sum_e b_e^2)^{1/2}}{\sqrt{1 + (K-1)\mu}}.$$

Proof. Nonpositive supporting margins require $g_e^\top \boldsymbol{\delta} \le -b_e + \eta_e r^2$. For indices with $b_e - \eta_e r^2 > 0$, squaring gives a lower bound on the corresponding entries of $G\boldsymbol{\delta}$, where G has rows $g_e^\top$. Therefore

$$\sum_e [b_e - \eta_e r^2]_+^2 \le \|G\boldsymbol{\delta}\|_2^2 \le \|G\|_2^2 r^2 \le [1 + (K-1)\mu] r^2.$$

The last step follows from the row-sum bound on the Gram matrix $GG^\top$. □

A strict reverse inequality at a radius $r \le r_0$ certifies that at least one supporting margin remains positive throughout that ball: the left-hand side decreases with the radius and the right-hand side increases. For equal affine thresholds $b_e = b$, orthogonal sensitivities yield the requirement $\sqrt{K}\, b$, compared with b for fully aligned sensitivities. The Taylor lower bound holds, for example, where the supporting margins have Lipschitz gradients; near a change of the maximizing competitor, it must be established separately. The proposition concerns input sensitivity. The parameter regularizer encourages complementary adaptation but does not by itself establish this condition.

B.4 DCAPA AND THE MARGIN QUALITY OF TRAINING NEIGHBORHOODS

Let $\mathbf{a}$ denote an adversarial example supplied by the current task. DCAPA adds perturbations around $\mathbf{a}$. Its radius therefore controls the additional displacement, not the total distance from the original clean image. For fixed batch size B, the marginal law of a uniformly selected output example is

$$\begin{aligned} \mathcal{Q}_i &= \alpha_0 \mathcal{Q}_{i,0} + \alpha_s \mathcal{Q}_{i,s} + \alpha_u \mathcal{Q}_{i,u}, \\ (\alpha_0, \alpha_s, \alpha_u) &= \frac{(B - N_{\text{pert}}, N_{\text{spec}}, N_{\text{univ}})}{B}. \end{aligned} \tag{30}$$

The three components retain the current examples, perturb them with source-specific gradients, and perturb them with mixed base/current gradients. In the specified run, $m = \rho = 0.5$. The training loop uses $B = 32$ for FGSM, PGD, and CW, giving counts $(16, 8, 8)$, and $B = 8$ for AutoAttack, giving $(4, 2, 2)$. Both full-batch cases have weights $(1/2, 1/4, 1/4)$. The component laws include batch selection and perturbation randomness; independence within a batch is unnecessary.

The source-specific component uses the current expert and the historical sources admitted to Ω in Section 3.4. Source weights and largest-remainder allocation determine how many examples each source receives. A positive source weight need not yield a sample in every batch. The update in Eq. equation 13 follows the source's signed loss gradient, the maximizer of its first-order loss increase over an unconstrained ℓ_∞ step box. Historical sources thus supply previously learned vulnerability directions on current data without replaying old training examples.

The universal component combines these sensitivities with stochastic exploration. Using the methodology's mixing coefficient β, write the whole selected minibatch direction as

$$V^q = (1-\beta)\,\mathrm{sign}(g_{\text{base}}^q) + \beta\,\mathrm{sign}(g_{\text{cur}}^q) + \xi^q, \qquad d^q = \frac{V^q}{\|V^q\|_{\mathrm{F}} + \varepsilon_0}.$$

Here $\xi^q \sim \mathcal{N}(0, 0.1^2 I)$ in the specified run, $\varepsilon_0 = 10^{-8}$, and the Frobenius norm is taken over the selected minibatch. For the current gradient of sample n,

$$\langle g^q_{\mathrm{cur},n}, d^q_n \rangle \geq \frac{(2\beta - 1)\|g^q_{\mathrm{cur},n}\|_1 - \|g^q_{\mathrm{cur},n}\|_2 \|\xi^q_n\|_2}{\|V^q\|_{\mathrm{F}} + \varepsilon_0}. \tag{31}$$

The inequality follows by bounding the base-gradient contribution below by $-(1-\beta)\|g^q_{\mathrm{cur},n}\|_1$ and applying Cauchy–Schwarz to the noise term. At $\beta = 0.7$, the deterministic numerator is $0.4\|g^q_{\mathrm{cur},n}\|_1$. Thus, the unprojected noiseless direction retains positive alignment with the current loss while incorporating base sensitivity. Random starts, noise, and per-example radii add further directions and scales; projection does not guarantee that every realized step increases loss.

The adaptive radius in the methodology satisfies

$$\epsilon_i = \phi \,\mathrm{clip}\left(1 - \frac{\bar{\ell}_{\mathrm{base},i}}{\log C}, \mathrm{scale}_{\min}, 1\right), \qquad \phi\,\mathrm{scale}_{\min} \leq \epsilon_i \leq \phi.$$

Here $\bar{\ell}_{\mathrm{base},i}$ denotes the average base-expert classification loss $\mathcal{L}(A_{\mathrm{base}}, \mathbf{D})$ used to set the radius for the current epoch in Section 3.4. In the specified configuration, $\phi = 0.0313$ and $\mathrm{scale}_{\min} = 0.05$. The universal radii lie between $\min(0.0078, R_{\max,i})$ and $R_{\max,i} = \min(0.0157, \epsilon_i)$; both channels use two steps.

The margin bound must account for the displacement of the complete augmentation trajectory. The source-specific projection in Eq. equation 13 gives $\|\mathbf{v}\|_\infty \leq \epsilon_i$. For the universal update as written in Eq. equation 16, let $\mathbf{a}_n$ be the original anchor and $\mathbf{a}^q_n$ its successive iterates, including the random initialization $\|\mathbf{a}^0_n - \mathbf{a}_n\|_\infty \leq R_n$. Since the normalized directions satisfy $\|d^q_n\|_\infty \leq 1$, summing the Q update steps yields

$$\|\mathbf{a}^Q_n - \mathbf{a}_n\|_\infty \leq R_n + \frac{R_n}{Q}\sum_{q=0}^{Q-1} \|d^q_n\|_\infty \leq 2R_n. \tag{32}$$

Thus, a common displacement bound for the two channels is

$$b_{\mathrm{aug},i} := \max\{\epsilon_i, 2R_{\max,i}\} \leq 2\epsilon_i.$$

If $d_F(\mathbf{a}+\mathbf{v}, \mathbf{a}) := \max_e \|u_e(\mathbf{a}+\mathbf{v}) - u_e(\mathbf{a})\|_2 \leq K_\infty \|\mathbf{v}\|_\infty$ throughout the resulting neighborhood, Proposition B.1 gives

$$M(\mathbf{a}+\mathbf{v}, y) \geq M(\mathbf{a}, y) - 2K_\infty b_{\mathrm{aug},i}. \tag{33}$$

For the source-specific channel alone, the tighter term $2K_\infty \epsilon_i$ applies. The adaptive radius controls these bounds through the measured base-expert loss, without imposing an ordering by attack name or task position.

For any nonnegative loss ℓ and any component with $\alpha_k > 0$, the mixture also satisfies

$$\mathbb{E}_{\mathcal{Q}_{i,k}}[\ell] \leq \frac{\mathbb{E}_{\mathcal{Q}_i}[\ell]}{\alpha_k}. \tag{34}$$

Positive mixture weights preserve each component's support: any open set with positive probability under a component also has positive probability under the mixture. Both this property and the loss inequality follow directly from the mixture definition. DCAPA therefore distributes training exposure across the original domain and the source-specific and universal neighborhoods described in Section 3.4. The following result connects the final margin quality on these neighborhoods to attack-conditioned classification error.

B.5 From neighborhood margins to an accuracy ordering

The preceding mechanisms affect the margin distribution of the final classifier. To relate that distribution to an evaluated attack, use the feature pseudometric

$$d_F(\mathbf{a}, \mathbf{b}) = \max_e \|u_e(\mathbf{a}) - u_e(\mathbf{b})\|_2.$$

Cauchy–Schwarz and the stability of a maximum imply $|M(\mathbf{a}, y) - M(\mathbf{b}, y)| \leq 2d_F(\mathbf{a}, \mathbf{b})$, even when the maximizing expert changes. Let $\mathcal{P}_\mathcal{A}$ be the labeled test distribution returned by attack $\mathcal{A}$,

and let $\mathcal{Q}_\mathcal{A}$ be a reference law formed from one or more DCAPA neighborhoods with the same label marginal. The reference law is an analytical construction; it does not require storing historical data. Define

$$W_F(\mathcal{P}_\mathcal{A}, \mathcal{Q}_\mathcal{A}) = \inf_{\pi \in \Pi_y(\mathcal{P}_\mathcal{A}, \mathcal{Q}_\mathcal{A})} \mathbb{E}_\pi[d_F(\mathbf{a}, \mathbf{b})],$$

where the couplings in Π_y preserve labels. For $\gamma > 0$, introduce two ramp functions,

$$\ell_\gamma^-(m) = \min\{1, \max\{0, -m/\gamma\}\}, \qquad \ell_\gamma^+(m) = \min\{1, \max\{0, 1 - m/\gamma\}\}.$$

They satisfy $\ell_\gamma^-(m) \le \mathbf{1}\{m \le 0\} \le \ell_\gamma^+(m)$.

Proposition B.3 (Two-sided transfer of margin error)**.** *For the fixed final classifier, let* $\mathcal{R}_\mathcal{A} = \Pr_{\mathcal{P}_\mathcal{A}}[M \le 0]$ *and* $w_\mathcal{A} = W_F(\mathcal{P}_\mathcal{A}, \mathcal{Q}_\mathcal{A})$*. Then*

$$\begin{aligned} L_\mathcal{A}(\gamma) &:= \max\left\{0, \mathbb{E}_{\mathcal{Q}_\mathcal{A}}[\ell_\gamma^-(M)] - \frac{2w_\mathcal{A}}{\gamma}\right\} \\ &\le \mathcal{R}_\mathcal{A} \le \min\left\{1, \mathbb{E}_{\mathcal{Q}_\mathcal{A}}[\ell_\gamma^+(M)] + \frac{2w_\mathcal{A}}{\gamma}\right\} =: U_\mathcal{A}(\gamma). \end{aligned} \tag{35}$$

In particular, $U_\mathcal{A}(\gamma) < L_\mathcal{B}(\gamma)$ *is sufficient for* $\mathrm{Acc}_\mathcal{A} > \mathrm{Acc}_\mathcal{B}$.

Proof. Both ramp functions are $1/\gamma$-Lipschitz. Their composition with the margin is therefore $2/\gamma$-Lipschitz in d_F. For either sign and any label-preserving coupling,

$$\left|\mathbb{E}_{\mathcal{P}_\mathcal{A}}[\ell_\gamma^\pm(M)] - \mathbb{E}_{\mathcal{Q}_\mathcal{A}}[\ell_\gamma^\pm(M)]\right| \le \frac{2}{\gamma}\mathbb{E}_\pi[d_F(\mathbf{a}, \mathbf{b})].$$

Taking the infimum over couplings and using the indicator bounds proves both inequalities. The ordering follows from $\mathcal{R}_\mathcal{A} \le U_\mathcal{A} < L_\mathcal{B} \le \mathcal{R}_\mathcal{B}$. □

The upper bound is small when the final model has positive margins on the reference neighborhoods and the evaluated attack remains close to them in feature space. The lower bound detects neighborhoods with persistently negative margins. Retained support and useful residual corrections affect the margin terms; complementary sensitivities affect their stability; DCAPA determines the reference neighborhoods. This gives an explicit sufficient condition for an accuracy reversal, rather than inferring one from two upper bounds. The reference losses are evaluated with the final classification margin. The cross-entropy used to train an expert cannot be substituted for them without establishing the corresponding margin behavior.

B.6 Consequences for CW, AutoAttack, and task order

B.6.1 CW and the geometry of distortion

The configured CW attack penalizes squared ℓ_2 distortion with $c = 1$ and confidence $\kappa = 0$, whereas FGSM and PGD use an ℓ_∞ budget of $8/255$. These are different optimization problems Carlini & Wagner (2017). For a locally affine decision margin $m + v^\top \boldsymbol{\delta}$, ignoring input-box clipping,

$$\max_{\|\boldsymbol{\delta}\|_\infty \le \epsilon} (-v^\top \boldsymbol{\delta}) = \epsilon \|v\|_1, \qquad \max_{\|\boldsymbol{\delta}\|_2 \le r} (-v^\top \boldsymbol{\delta}) = r\|v\|_2. \tag{36}$$

The distinction persists for a penalized objective. For $m > 0$, $v \ne 0$, and $c > 0$, consider the local CW-type problem

$$\min_{\boldsymbol{\delta}} \ \|\boldsymbol{\delta}\|_2^2 + c[m + v^\top \boldsymbol{\delta}]_+.$$

For a prescribed margin reduction $t = -v^\top \boldsymbol{\delta} \ge 0$, the minimum squared distortion is $t^2/\|v\|_2^2$. Minimizing $t^2/\|v\|_2^2 + c[m - t]_+$ gives

$$t^* = \min\{m, c\|v\|_2^2/2\}.$$

Hence, whenever $c\|v\|_2^2/2 < m < \epsilon\|v\|_1$, this idealized penalized optimum retains a positive margin while an ℓ_∞ margin-minimizing step crosses the boundary. This example establishes the absence of a universal ordering even before finite optimization is considered; it does not replace the actual CW search by a linear model.

For LREERC, distortion matters through the normalized displacements in Proposition B.1. If those displacements are bounded by $K_2\|\boldsymbol{\delta}\|_2$, a clean margin exceeding $2K_2\|\boldsymbol{\delta}\|_2$ survives. Complementary supporting margins can further increase the displacement needed to suppress all correct-class evidence, as quantified by Proposition B.2. CW examples can therefore be well classified when they preserve this support or remain near positive-margin DCAPA neighborhoods. This explanation requires properties of the learned representation and the realized examples, rather than a presumption that CW always produces less harmful perturbations.

B.6.2 AutoAttack and the distribution of surviving margins

AutoAttack combines adaptive PGD variants, FAB, and Square Attack Croce & Hein (2020). Its comparison with PGD at a common ℓ_∞ budget cannot be explained by the norm distinction above. The relevant question is whether its returned examples more often retain positive margins in the fixed final classifier. A sufficient condition is provided by Proposition B.3: AutoAttack has a lower error rate if its upper error bound lies below the lower bound for FGSM or PGD. Historical support and the augmented neighborhoods specify the model-dependent quantities in that condition; membership in an attack ensemble does not determine their values.

There is also an exact characterization on paired test examples. Let $\mathcal{E}_{\mathcal{A}} = \{M(\mathbf{x} + \boldsymbol{\delta}_{\mathcal{A}}, y) \le 0\}$, with attack randomness included in the joint experiment. For $\mathcal{B} \in \{\mathrm{FGSM}, \mathrm{PGD}\}$,

$$\mathrm{Acc}_{\mathcal{A}} - \mathrm{Acc}_{\mathcal{B}} = \Pr(\mathcal{E}_{\mathcal{B}} \setminus \mathcal{E}_{\mathcal{A}}) - \Pr(\mathcal{E}_{\mathcal{A}} \setminus \mathcal{E}_{\mathcal{B}}). \tag{37}$$

The observed advantage means that correct-class support survives on more examples uniquely misclassified under the comparison attack than vice versa. This identity links the accuracy difference to the margin analysis without requiring a universal ranking of finite attacks. It also fixes the interpretation of a stronger evaluation: if a failure under either attack is retained according to the final classifier, then

$$\mathrm{Acc}_{\mathcal{A}\cup\mathcal{B}} = 1 - \Pr(\mathcal{E}_{\mathcal{A}} \cup \mathcal{E}_{\mathcal{B}}) \le \min\{\mathrm{Acc}_{\mathcal{A}}, \mathrm{Acc}_{\mathcal{B}}\}.$$

The explanation concerns separately returned attack distributions under the stated protocol, not increasing accuracy as the retained adversarial candidate set grows.

B.6.3 Training history and the scope of the explanation

The task order is clean, FGSM, PGD, CW, and AutoAttack. CW and AutoAttack experts are consequently trained after more historical representations and perturbation sources have become available. Selective residual reuse supplies compatible corrections during their adaptation, and historical sources can guide augmentation on their current examples. This asymmetry concerns training: every attack in the final comparison is scored by the same retained expert collection. A larger historical pool alone does not imply a larger positive margin, because residual selection may reject candidates and new experts can introduce competing scores.

Together these results give a conditional, testable explanation. Historical retention preserves class evidence; useful residual reuse and complementary adaptation can strengthen its resistance to perturbation; DCAPA trains on several nearby distributions; the final margins and feature distances then determine attack-conditioned error. The reported accuracies establish the ordering, while Propositions B.1–B.3 identify sufficient conditions under which the learned geometry produces it. Establishing which mechanism accounts for each observed gain additionally requires final-margin distributions, supporting-margin survival rates, and matched component ablations. Figure 1 supports the benefit of selective knowledge reuse, but does not by itself measure these conditions for the CW and AutoAttack comparisons.

C Additional Information about Related Work

C.1 Additional information about low-rank adaptation

Low-rank adaptation (LoRA) Hu et al. (2021) parameterizes updates to frozen pre-trained weights with low-rank factors, reducing adaptation and storage costs without additional inference latency when merged. AdaLoRA Zhang et al. (2023) allocates ranks by importance, while LoRA+

Hayou et al. (2024) uses factor-specific learning rates. LoRA-Pro Wang et al. (2025c) approximates full fine-tuning updates through gradient correction, LoRA-RITE Yen et al. (2025) applies transformation-invariant preconditioning, and GeoLoRA Schotthöfer et al. (2025) adapts ranks via geometric low-rank integration. In CL, InfLoRA Liang & Li (2024) suppresses cross-task interference through low-rank subspaces while preserving plasticity, whereas CL-LoRA He et al. (2025) combines continuously updated shared adapters with task-specific adapters for rehearsal-free knowledge accumulation and task-specific plasticity.

C.2 Additional information about baselines

This section describes the architectures, optimization objectives, and operating mechanisms of the baselines evaluated in our experiments. These descriptions clarify the technical differences among the methods and provide context for interpreting the empirical comparisons presented in the preceding sections.

DER Buzzega et al. (2020) addresses catastrophic forgetting in General Continual Learning (GCL) through replay of previously recorded network outputs. It stores logits collected along the optimization trajectory and encourages the current model to reproduce them on replayed examples. This preserves information about earlier predictions beyond their hard class assignments, helping maintain consistency with past experience. DER requires neither explicit task boundaries during training nor a task oracle at inference, making it applicable to the GCL setting.

DER++ Buzzega et al. (2020) augments DER's logit-matching objective with supervised replay using ground-truth labels. The additional conditional-likelihood term complements the information retained in stored logits by reinforcing correct predictions on past examples. This is particularly useful under abrupt distribution shifts, where stored logits may be biased and provide insufficient guidance on their own. DER++ retains DER's independence from task boundaries and test-time task information while improving resilience to distribution changes, especially in domain-incremental learning.

DEF Buzzega et al. (2020) incorporates adversarial training into DER's replay procedure to support continual learning under adversarial perturbations. FGSM generates adversarial examples during both training on incoming data and memory replay, while a mean squared error (MSE) objective encourages agreement between clean and adversarial outputs. Incorporating adversarial examples into the replay buffer allows the model to revisit earlier experiences together with their perturbed counterparts. This extends DER's knowledge-retention mechanism to adversarial robustness while preserving its ability to operate without explicit task boundaries in GCL.

DER+++refresh Wang et al. (2024) combines DER++ with Refresh Learning for robust continual learning. Its objective couples label-consistency regularization, weighted by β, with logit matching, weighted by α, through cross-entropy and MSE terms, respectively. FGSM examples are generated for both incoming and replayed data. Together, these components reinforce label information during replay and encourage consistent predictions between clean inputs and their adversarial counterparts.

SEMA Wang et al. (2025a) determines when to reuse existing adapters and when to expand the model by monitoring distribution changes at different representation levels. Each module pairs a functional adapter with a representation descriptor (RD), which assesses whether incoming features are compatible with the distribution represented by that module. A new adapter is introduced at a given layer only when all existing RDs indicate a sufficiently novel distribution. An expandable router, optimized jointly with the adapters, combines their outputs through learned mixture weights, allowing previously acquired knowledge to be reused across distributions. This selective expansion balances adaptation to new data with retention of established representations. By adding capacity only when needed, SEMA achieves sublinear model growth and reduces the resource demands associated with continually expanding adapter collections.

iGAT Deng & Mu (2023) improves adversarial training for ensembles through coordinated example assignment and regularization. Its theoretical motivation comes from a result showing that an ensemble of two multilayer perceptrons can attain lower empirical 0–1 loss than a single classifier on an ambiguous pair set comprising challenging, weakly separable examples. Building on this analysis, iGAT introduces two mechanisms that can complement existing ensemble adversarial training methods. First, a distribution rule assigns globally generated adversarial examples to indi-

vidual classifiers according to their predicted probabilities for the ground-truth class. Assignment either selects the highest-scoring classifier or uses a ranking-based roulette-wheel strategy. This encourages classifiers to develop their respective areas of competence and supports the acceptability condition underlying the theoretical analysis. Second, a misclassification regularizer penalizes the most erroneous prediction of the weakest classifier on each example. This term targets regions of shared vulnerability where the ensemble members fail together, complementing the specialization encouraged by example assignment.

AIR Zhou & Hua (2024) uses self-distillation with pseudo-replay to mitigate forgetting in continual adversarial defense without accessing data from previous attacks. It combines isotropic and anisotropic augmentation to preserve earlier defensive knowledge while learning from newly encountered attacks. Isotropic replay encourages prediction consistency within the neighborhood distribution of incoming data, indirectly aligning predictive preferences across tasks. Anisotropic replay introduces mixed semantic content to construct a compromise data manifold that supports resilience to future attacks. An output-alignment regularizer connects learning on new and previous tasks, helping balance the acquisition of new defensive capabilities with retention of existing robustness.

DER++AFLC Mi et al. (2025) addresses the accelerated forgetting that can arise when adversarial training is incorporated into continual learning. Built on DER++, it calibrates logits differently for past, current, and future classes to reduce adverse gradient effects on previously learned tasks while retaining the capacity to learn new ones. Its Further Prior (FP) adjustment assigns future classes a calibration value obtained by averaging those associated with past and current classes. This regulates gradient allocation across the three class groups and limits the interference introduced by adversarial examples. The approach improves the balance between robustness acquisition and knowledge retention without requiring additional data or elaborate architectural changes.

AAOpt Wesego & Rooshenas (2025) treats test-time adversarial purification as maximum a posteriori (MAP) estimation, combining a pretrained diffusion prior with a learned model of adversarial perturbations. The method is motivated by evidence that such perturbations exhibit shared structure, including universal adversarial directions and recurring decision-boundary geometry, which generic Gaussian noise models do not capture. A lightweight score-based diffusion network is trained on residuals $\gamma = x_{\text{clean}} - x_{\text{adv}}$ obtained from a mixture of ℓ_∞ and ℓ_2 attacks. This network estimates the score associated with the perturbation likelihood $\log p(\hat{x} \mid x)$. Given a potentially corrupted input $\hat{x}$, purification proceeds through a small number of gradient-based refinement steps. The optimization combines a diffusion-prior term that favors natural images with a perturbation term that accounts for learned adversarial structure. Stop-gradient operations on the score networks reduce the computational cost of these updates. AAOpt requires no classifier retraining or costly reverse-diffusion sampling, and its reported evaluations demonstrate generalization to unseen classifier architectures and attack norms.

PDFK Zhou et al. (2026) targets knowledge fragility in online continual learning (OCL), where correctly learned examples become misclassified after only a few minor parameter updates. It relates this behavior to interacting spatial and temporal factors. Spatially, vulnerable examples occupy high-curvature regions of the loss landscape, where small parameter changes can alter predictions. Temporally, noisy online optimization produces rapid oscillations that repeatedly move the model through these sensitive regions. PDFK addresses the temporal component using an exponential moving average (EMA) of model parameters as a self-ensemble teacher. This teacher smooths short-term fluctuations and supplies soft distillation targets for replayed examples. To address the spatial component, PDFK introduces lightweight structured parameter perturbations and enforces agreement between perturbed and unperturbed predictions on correctly classified buffer examples. This regularization encourages a flatter local loss landscape and enlarges the fragility radius of vulnerable examples, reducing their susceptibility to forgetting under subsequent updates.

DRIFT Guesmi & Shafique (2026) is a stochastic adversarial defense that learns lightweight, differentiable input filters while keeping the pretrained classifier frozen. It is motivated by gradient consensus: alignment between gradients across input transformations can enable adversarial perturbations to transfer between them. Each filter consists of a shallow residual convolutional module that preserves the input dimensions. Training combines clean-data cross-entropy, two gradient-separation objectives, and an adversarial loss. The separation objectives penalize squared cosine similarity between randomly probed vector–Jacobian products, measuring alignment in the filters'

input–output sensitivities and in the input sensitivities of classifier logits. The latter also includes comparisons with the unfiltered classifier, accounting for adversarial directions derived directly from the backbone. For adversarial training, projected gradient descent generates perturbations against the frozen classifier, and the filters minimize the maximum classification loss across ensemble members on these examples. At inference, one filter is sampled for each forward pass. The resulting defense seeks to preserve clean predictive performance while reducing adversarial directions shared across transformations, without retraining the backbone.

CL-LoRA He et al. (2025) addresses rehearsal-free class-incremental learning by combining shared and task-specific LoRA modules within a frozen pretrained vision transformer. Shared adapters are placed in earlier transformer blocks, while task-specific adapters occupy later blocks; both modify the query and value projections of self-attention. Each shared adapter uses a fixed random orthogonal down-projection and a trainable up-projection that is updated across tasks. To preserve accumulated knowledge, CL-LoRA applies distillation at the transition between shared and task-specific blocks, matching softened predictions obtained from the current and previous shared adapters. The distillation gradients are reweighted using the ℓ_2 norms of the previous up-projection weight vectors, strengthening preservation in dimensions associated with greater prior adaptation. For task-specific learning, trainable block scaling factors regulate adapter contributions, while an orthogonality penalty discourages overlap between the scaling vectors of different tasks. At inference, the shared representation is computed once and processed by the task-specific branches. Classification selects the highest cosine similarity between these features and their corresponding class prototypes. This design supports cross-task knowledge retention and task-specific adaptation without replaying stored examples from earlier tasks.

C.2.1 Scope of the contribution.

Our framework builds on established ideas of parameter-efficient adaptation, historical knowledge retention, and expert reuse. CL-LoRA He et al. (2025) explicitly learns cross-task commonalities through shared adapters, while preserving specialized information in separate task-specific modules. Our distinction concerns the direct reuse of that specialized information: historical attack experts remain frozen and can be retrieved as residual contributors to a new configuration. Their configurations can also guide perturbation generation on current adversarial inputs.

AIR Zhou & Hua (2024) already uses historical knowledge and complementary pseudo-replay mechanisms for continual defense. Accordingly, our contribution is not the general idea of knowledge sharing or multiple augmentation channels. It is the use of adversarial gradient compatibility to organize frozen historical experts for residual composition and expert-guided perturbation generation. Residual selection, perturbation-source allocation, and final prediction retain distinct roles, as specified in Appendix A.4.

D Additional Information on Experiment Settings

We conduct experiments on four benchmarks: CIFAR-10 and CIFAR-100 Krizhevsky et al. (2009), CUB-200 Wah et al. (2011), and ImageNet-R Hendrycks et al. (2021). These datasets differ in class granularity and visual characteristics, enabling evaluation across a range of continual learning settings. For all datasets, we use a Vision Transformer (ViT) Dosovitskiy et al. (2020) pretrained on ImageNet-21K and subsequently fine-tuned on ImageNet-1K as the backbone.

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.

D.1 Framework configurations

Pretrained Vision Transformer. We use ViT-B/16 with 12 Transformer blocks, an embedding dimension of 768, and 12 attention heads. The backbone is initialized from a pretrained ViT weight, corresponding to the configuration with supervised ImageNet-21K pretraining followed by ImageNet-1K fine-tuning. The backbone parameters remain frozen during continual adaptation.

Table 3: Detailed dataset division.

Dataset	Class Number	Phase	Division
CIFAR-10	10	Training	5 tasks, 10 classes per task
		Testing	5 tasks, 10 classes per task
CIFAR-100	100	Training	5 tasks, 100 classes per task
		Testing	5 tasks, 100 classes per task
CUB-200	200	Training	5 tasks, 200 classes per task
		Testing	5 tasks, 200 classes per task
ImageNet-R	200	Training	5 tasks, 200 classes per task
		Testing	5 tasks, 200 classes per task

Table 4: Basic information and transformations for datasets.

Dataset	Original Size	Phase	Resizing & Cropping
CIFAR-10	$3 \times 32 \times 32$	Training	Resize to 224×224, random crop (padding 28), horizontal flip
		Testing	Resize to 224×224
CIFAR-100	$3 \times 32 \times 32$	Training	Resize to 224×224, random crop (padding 28), horizontal flip
		Testing	Resize to 224×224
CUB-200	$3 \times 224 \times 224$	Training	Resize to 224x224, Center crop, Horizontal Flip
		Testing	Resize to 224x224, Center crop
ImageNet-R	$3 \times 64 \times 64$	Training	Resize to 224x224, Random crop (padding 28), Horizontal Flip
		Testing	Resize to 224x224

D.2 Dataset configurations

This section provides comprehensive dataset configuration details as detailed in Tab. 3 and Tab. 4, including per-dataset class counts, segmentation of data across various phases, image resolution and preprocessing protocols.

CIFAR-10 images are resized to 224×224 for both training and testing. Training additionally applies random cropping with padding 28 and horizontal flipping. Inputs are represented in $[0, 1]$ with identity normalization, and attack budgets are measured on these resized inputs. Preprocessing for the remaining datasets is summarized in Table 4.

D.3 Methods configurations

To ensure consistent experimental conditions, we set the replay buffer capacity to 500 samples for methods that use replay and adopt a current-task training batch size of 32 across all methods. For the AutoAttack task, the training batch size is uniformly reduced to 8 to accommodate its memory and computational demands, while the replay buffer capacity remains unchanged. For AIR, we apply dataset-specific noise augmentation to clean training samples. We adapt SEMA from its original domain-incremental setting to our task-incremental scenario and further extend iGAT and AAOpt to the same continual learning protocol, evaluating their ability to retain adversarial robustness across sequentially presented tasks. These adaptations align the methods with the shared experimental setting to support fair comparisons.

D.4 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 DEF, we use $\alpha = 0.5$.

(3) For iGAT, we adopt the ensemble adversarial defense framework with $N = 4$ base classifiers under the soft distributing rule. We configure the global adversarial loss weight as $\gamma = 5.0$ and the misclassification regularization weight as $\eta = 10.0$, consistent with the settings recommended for ADP/CLDL/DVERGE backbones in the original paper. Adversarial training is performed via PGD with 5 steps at a step size of 2.0.

(4) For AIR, to adapt to our experimental environment, we adjust the hyperparameters based on the recommended settings in the original papers: we set $\lambda_{\text{SD}} = 3.0$, $\lambda_{\text{Reg}} = 2.0$, $\alpha_{\min} = 0.2$, $\alpha_{\max} = 0.5$.

(5) For AAOpt, the DER component inherits the configuration from (1), i.e., $\alpha = 0.3$. The AAOpt purification mechanism, introduces the parameter λ that governs the trade-off between the pretrained diffusion prior and the learned perturbation model; we set $\lambda = 0.5$, adhering to the value recommended for BPDA-EOT evaluation in the original work.

(6) For PDFK, we set the spatial perturbation coefficient to 0.5 and the temporal consistency coefficient to 1.0, corresponding to η_p and η_c of the original framework. The EMA decay rate is fixed at 0.999. The structured perturbation mechanism adopts an initial noise scale of 0.01 and a perturbation step size of 0.05, with a single perturbation iteration per update.

(7) For DRIFT, we retain its core design of a learnable filter ensemble preceding a frozen classifier, together with classification, Jacobian separation, logit-VJP separation, and base-model adversarial training objectives. We adapt this design to sequential attack tasks under our shared training protocol. The target classifier is first adapted on clean data using SGD with a learning rate of 0.03, matching the learning-rate setting of the other models for a fair comparison. It is then frozen, while the filters retain the original AdamW learning rate of 0.001. To accommodate GPU memory constraints, both separation losses are evaluated on random sub-batches of eight samples. Logit-VJP optimization uses a stop-gradient approximation that treats classifier-side vector–Jacobian products as constants while retaining differentiation through the filter Jacobians. Sequential backpropagation accumulates component gradients before a single optimizer update, and activation checkpointing and smaller attack-generation batches further reduce peak memory usage. For continual adaptation, a buffer of 500 clean samples supports label supervision, consistency with the frozen classifier's logits, and an identity-preservation penalty weighted by 0.5. These samples also anchor the internal PGD adversaries, carrying the original base-model adversarial training mechanism into the continual setting. Filter residuals are smoothly bounded to approximately $8/255$ per pixel, with outputs clipped to $[0, 1]$. External attacks use the mean ensemble logits as a deterministic surrogate.

(8) For all other methods, we adopt the hyperparameters recommended by the original authors to ensure consistent and reproducible evaluation under standardized settings.

(9) Finally, for our framework, the LoRA rank is set to 10. Historical residual selection uses `fusion_gate=relative`, with similarity threshold $\tau = 0.1$, relative threshold $r = 0.6$, and maximum residual count $K_{\text{res}} = 5$. Historical adversarial experts first satisfy $s_t > \tau$ to enter $\mathcal{V}_i$ in Eq. equation 6. Their composition coefficients are the softmax weights in Eq. equation 7. The second gate requires $q_t = s_t / \max_{u \in \mathcal{V}_i} s_u > r$, after which Eq. equation 8 retains at most five candidates in decreasing similarity order, resolving ties by increasing historical index. The retained coefficients are not renormalized. If $\mathcal{V}_i = \varnothing$, no historical residual is inserted and the orthogonality penalty is zero. The effective orthogonality coefficient remains $\lambda\lambda_{\text{orth}} = 10^{-3}$, controlling the contribution of the probability-weighted orthogonality penalty relative to the classification loss.

For Dual-Channel Adaptive Perturbation Augmentation (DCAPA), the total augmentation ratio is $m = 0.5$, and the source-specific channel ratio is $\rho = 0.5$ (Eq. 11), retaining half of each adversarial mini-batch without additional perturbations and allocating one quarter to each augmentation channel. The maximum augmentation radius is $\phi = 0.0313$ (approximately $8/255$), with

$\text{scale}_{\min} = 0.05$ (Eq. 12). Accordingly, the adaptive radius satisfies $\epsilon \in [0.05\phi, \phi]$, reducing augmentation strength as the frozen base expert's average classification loss increases while preserving a nonzero lower bound. Within the source-specific channel, `perturb_max_hist=5` independently limits the number of historical sources. Sources with first-stage coefficients $k_t > 10^{-8}$ are prioritized in decreasing coefficient order, followed by the remaining historical experts in stored order; membership in $\mathcal{S}_i$ is not required. The allocation weight is 1.0 for the current expert, k_t for historical sources above this tolerance, and 0.02 otherwise. For the universal channel, the gradient-mixing coefficient is $\beta = 0.7$ (Eq. 15), assigning weights of 0.7 and 0.3 to the signed gradients of the current and clean base experts, respectively. The smoothing term follows $\boldsymbol{\xi}^q \sim \mathcal{N}(\mathbf{0}, 0.1^2\mathbf{I})$ and is added before direction normalization. Per-sample radii are drawn from $\mathcal{U}[R_{\min}, R_{\max}]$, with $R_{\max} = \min(0.0157, \epsilon)$ and $R_{\min} = \min(0.0078, R_{\max})$, ensuring that the sampling interval contracts with the adaptive radius. Random initialization is enabled within each sampled radius. Both channels use $Q = 2$ perturbation steps, with step sizes ϵ/Q and R_n/Q for the source-specific and universal updates, respectively (Eqs. 13 and 16).

D.5 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.

D.6 Inference and attack target.

Let $\mathcal{E}_i$ denote the expert configurations retained after task $\mathcal{T}_i$. Each configuration includes its saved residual composition. Let $h_e(\mathbf{x})$ and $p_{e,c}$ denote its feature representation and class-c prototype, respectively, and define $\text{nrm}(v) = v/\max\{\|v\|_2, \varepsilon_{\text{n}}\}$. The complete classifier uses

$$z_{i,c}(\mathbf{x}) = \max_{e \in \mathcal{E}_i} \text{nrm}(h_e(\mathbf{x}))^\top \text{nrm}(p_{e,c}), \qquad \widehat{y}_i(\mathbf{x}) = \arg\max_c z_{i,c}(\mathbf{x}). \tag{38}$$

Prediction requires no task or attack identity. All external evaluation attacks target this complete score function, including the retained expert configurations, prototype normalization, and class-wise maximum. At each evaluation checkpoint, adversarial examples are generated against the classifier at that checkpoint, with its parameters and prototypes fixed during attack generation and evaluation.

D.7 Adversarial attack configurations

We evaluate FGSM with an ℓ_∞ budget of $8/255$; PGD with the same budget, step size $2/255$, and 10 iterations; CW with $c = 1$, confidence parameter $\kappa = 0$, 100 iterations, and learning rate 0.01; and AutoAttack with an ℓ_∞ budget of $8/255$. The PGD configuration starts from the unperturbed input. The configured CW objective penalizes ℓ_2 distortion and does not impose the same ℓ_∞ constraint as the other three procedures.

The white-box attack target is the complete classifier in Eq. equation 38. Gradient-based attacks differentiate through the retained expert configurations, normalized prototype similarities, and class-wise maximum; query-based components use the same final scores. For every evaluated checkpoint, attacks are regenerated against that checkpoint's classifier on the held-out test inputs. The returned examples are evaluated using this identical prediction rule, without an oracle for selecting an expert.

The resulting accuracies measure performance under the specified attack procedures. They do not constitute certificates of worst-case robustness. In particular, the ordering of separately evaluated PGD and AutoAttack accuracies does not determine the accuracy obtained by retaining successful candidates from their per-example union.

D.8 Evaluation metrics

Final accuracy. We measure final accuracy after the model has completed training on the entire task sequence $\{\mathcal{T}_t\}_{t=1}^T$, where $T = 5$. The five tasks correspond to clean data, FGSM, PGD, CW, and AutoAttack, respectively. After learning $\mathcal{T}_T$, the same final model is evaluated sequentially on the test sets of $\mathcal{T}_1, \ldots, \mathcal{T}_T$, with its parameters fixed throughout evaluation.

Let $\mathcal{D}_t^{\text{test}}$ denote the held-out test set associated with task $\mathcal{T}_t$, and let $\widehat{y}_T^{(s)}(\mathbf{x}) \in \{1, \ldots, C\}$ denote the class predicted by the complete model after learning all T tasks in run s. The final classification accuracy on task $\mathcal{T}_t$ is defined as

$$\text{Acc}_{T,t}^{(s)} = \frac{100}{|\mathcal{D}_t^{\text{test}}|} \sum_{(\mathbf{x},\mathbf{y}) \in \mathcal{D}_t^{\text{test}}} \mathbf{1}\left[\widehat{y}_T^{(s)}(\mathbf{x}) = \underset{c \in \{1,\ldots,C\}}{\arg\max}\ \mathbf{y}[c]\right], \tag{39}$$

where $\mathbf{1}[\cdot]$ is the indicator function, $\mathbf{y}$ is the one-hot class label used in the methodology, and C is the number of classes. Here, $t \in \{1, \ldots, 5\}$ indexes the evaluated task, while the first subscript $T = 5$ specifies that evaluation uses the model obtained after the final training task. Thus, the five task columns in Table 1 report $\text{Acc}_{5,1}, \ldots, \text{Acc}_{5,5}$, respectively, characterizing retained clean accuracy and adversarial robustness after the complete learning sequence.

For each method and dataset, we repeat the complete training and evaluation procedure three times ($s = 1, 2, 3$). Each entry in Table 1 is reported as the mean $\pm$ standard deviation of the corresponding task's final accuracy across these three runs. The statistics are computed separately for each task, thereby capturing both final performance and its variability across runs.

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:

$$FR_i^{(j)} = \frac{A_{i,i} - A_{i,j}}{A_{i,i}} \times 100 \tag{40}$$

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 four distinct attack sequences as the evaluation metric, in order to systematically characterize the resilience to forgetting of different models throughout the entire training process.

Adversarial accuracy. We also employ the adversarial accuracy to specifically assess the model's adversarial defense capability. Let $A_i^{(j)}$ be defined as in the average accuracy metric. After training on the j-th task (i.e., the $(j-1)$-th adversarial attack), the adversarial accuracy $\overline{A}_{\text{adv}}^{(j)}$ is computed as the mean accuracy over all adversarial tasks excluding the clean data task $\mathcal{T}_0$:

$$\overline{A}_{\text{adv}}^{(j)} = \frac{1}{j} \sum_{i=1}^{j} A_i^{(j)}, \tag{41}$$

where $j \geq 1$ (i.e., at least one adversarial task has been trained). This metric provides a focused evaluation of the model's robustness against a sequence of adversarial attacks, decoupled from its performance on clean samples.

Silhouette Score. To quantitatively assess the quality of the learned feature representations in terms of intra-class compactness and inter-class separation, we adopt the Silhouette Score Rousseeuw (1987). For a dataset with n samples partitioned into k clusters, the silhouette coefficient of a sample i assigned to cluster C_I is defined as:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \tag{42}$$

where $a(i) = \frac{1}{|C_I|-1} \sum_{j \in C_I, j \neq i} d(i,j)$ is the mean intra-cluster distance (the average dissimilarity between i and all other samples in the same cluster), and $b(i) = \min_{J \neq I} \frac{1}{|C_J|} \sum_{j \in C_J} d(i,j)$ is the

mean nearest-cluster distance (the lowest average dissimilarity between i and all samples of any other cluster). The distance $d(i, j)$ is typically taken as the Euclidean distance in the feature space. The overall Silhouette Score $\overline{s}$ is the arithmetic mean of $s(i)$ over all n samples:

$$\overline{s} = \frac{1}{n} \sum_{i=1}^{n} s(i). \tag{43}$$

The Silhouette Score ranges from -1 to 1, with higher values indicating better-defined clusters. A score close to 1 signifies that samples are well-matched to their own cluster and poorly matched to neighboring clusters; a score near 0 indicates overlapping clusters; and negative values suggest that samples may have been assigned to the wrong cluster.

Calinski–Harabasz (CH) Score. As a complementary clustering quality metric, we employ the Calinski–Harabasz Index Caliński & Harabasz (1974), also known as the Variance Ratio Criterion. Let n be the total number of samples, k be the number of clusters, $\mathbf{z}_i$ denote the feature representation of the i-th sample, $\boldsymbol{\mu}_C$ be the centroid of cluster C, and $\boldsymbol{\mu}$ be the global centroid of all samples. The within-cluster scatter matrix $\mathbf{W}_k$ and the between-cluster scatter matrix $\mathbf{B}_k$ are defined as:

$$\mathbf{W}_k = \sum_{C=1}^{k} \sum_{i \in C} (\mathbf{z}_i - \boldsymbol{\mu}_C)(\mathbf{z}_i - \boldsymbol{\mu}_C)^\top, \quad \mathbf{B}_k = \sum_{C=1}^{k} |C| (\boldsymbol{\mu}_C - \boldsymbol{\mu})(\boldsymbol{\mu}_C - \boldsymbol{\mu})^\top. \tag{44}$$

The CH Score is then given by:

$$\text{CH} = \frac{\operatorname{tr}(\mathbf{B}_k)/(k-1)}{\operatorname{tr}(\mathbf{W}_k)/(n-k)}, \tag{45}$$

where $\operatorname{tr}(\cdot)$ denotes the trace operator. A higher CH Score indicates that clusters are dense and well-separated, i.e., the between-cluster variance is large relative to the within-cluster variance.

D.9 Experimental Details for Figure 1

Experimental objective. Figure 1 examines two complementary roles of historical adversarial knowledge: improving the initial representation for a newly encountered attack through residual composition, and improving generalization to unseen attacks through expert-guided perturbation augmentation during training. Panel (a) examines the former at PGD-task initialization, whereas panels (b) and (c) examine the latter through comparisons of historical knowledge sharing.

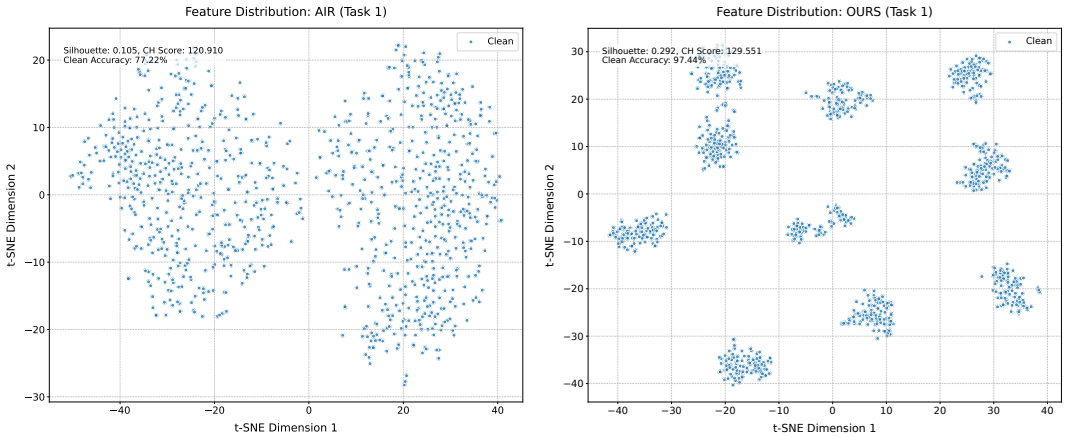
(a) Feature transfer at PGD-task initialization. We compare the standard LoRA model without historical-expert fusion, denoted by *Base*, with the model initialized using LREERC for the incoming PGD task. Historical expert residuals provide the reused knowledge in the LREERC configuration. The comparison is performed at initialization for this task, before adaptation to its training data. For each evaluated sample, we measure the consistency between its clean and PGD representations and its true-class margin under PGD. Let $h_M(\mathbf{x})$ denote the feature representation produced by model M, let $s_{M,c}(\mathbf{x})$ denote its classification score for class c, and let $(\mathbf{x}, \mathbf{x}^{\mathrm{PGD}}, y)$ denote a clean input, its PGD counterpart, and the ground-truth label. The two plotted quantities are

$$\begin{aligned} a_M(\mathbf{x}) &= \frac{h_M(\mathbf{x})^\top h_M(\mathbf{x}^{\mathrm{PGD}})}{\|h_M(\mathbf{x})\|_2 \|h_M(\mathbf{x}^{\mathrm{PGD}})\|_2}, \\ m_M(\mathbf{x}^{\mathrm{PGD}}, y) &= s_{M,y}(\mathbf{x}^{\mathrm{PGD}}) - \max_{c \neq y} s_{M,c}(\mathbf{x}^{\mathrm{PGD}}). \end{aligned} \tag{46}$$

The horizontal axis reports a_M, with larger values indicating stronger clean-to-PGD feature alignment. The vertical axis reports m_M, with larger values indicating greater separation of the true class from the strongest competing class; a positive margin means that the true-class score exceeds every competing score. Arrows connect measurements of the same sample under *Base* and LREERC. An upward-right displacement therefore indicates simultaneous improvements in feature consistency and class separation. The positive median shift summarizes the overall improvement and supports the motivation for historical knowledge reuse at initialization.

(b) Historical knowledge sharing for perturbation augmentation. We compare three scopes of expert knowledge used to guide perturbation generation during training. *Latest* uses the latest expert

Figure 4: Feature distributions of AIR and our model on clean samples (Task 1) on the CIFAR-10 dataset.



without historical knowledge sharing. + *Similar* additionally incorporates the robust knowledge of selected similar historical experts. + *Remaining* further incorporates the remaining historical experts beyond the selected subset. Historical knowledge affects training through the perturbations generated to augment the current training examples. The resulting models are evaluated on unseen attacks to examine whether this additional training signal transfers beyond the observed attack distributions. The improvement from *Latest* to + *Similar* supports selective historical knowledge sharing, while the reduction after adding the remaining experts indicates that broadening the source pool does not necessarily improve generalization.

(c) Per-attack generalization gains. We further compare the latest expert without historical knowledge sharing against the model trained with perturbation augmentation guided by selected historical experts, evaluating each on unseen PGD-ℓ_2, DeepFool, and CW attacks. For an evaluated model M and attack $\mathcal{B}$, classification accuracy is expressed as a percentage:

$$\mathrm{Acc}_M(\mathcal{B}) = \frac{100}{N} \sum_{n=1}^{N} \mathbf{1}\left[\widehat{y}_M(\mathbf{x}_n^{\mathcal{B}}) = y_n\right], \tag{47}$$

where N is the number of evaluated examples, $\mathbf{x}_n^{\mathcal{B}}$ is an adversarial input produced by attack $\mathcal{B}$, and $\widehat{y}_M$ is the model prediction. Hollow circles denote *Latest*, and filled diamonds denote the model trained with selected historical knowledge sharing. The annotated gain is $\Delta_{\mathcal{B}} = \mathrm{Acc}_{\mathrm{selected}}(\mathcal{B}) - \mathrm{Acc}_{\mathrm{latest}}(\mathcal{B})$, measured in percentage points rather than relative percentage improvement. Positive gains across the three attacks support the benefit of selected historical knowledge for cross-attack generalization.

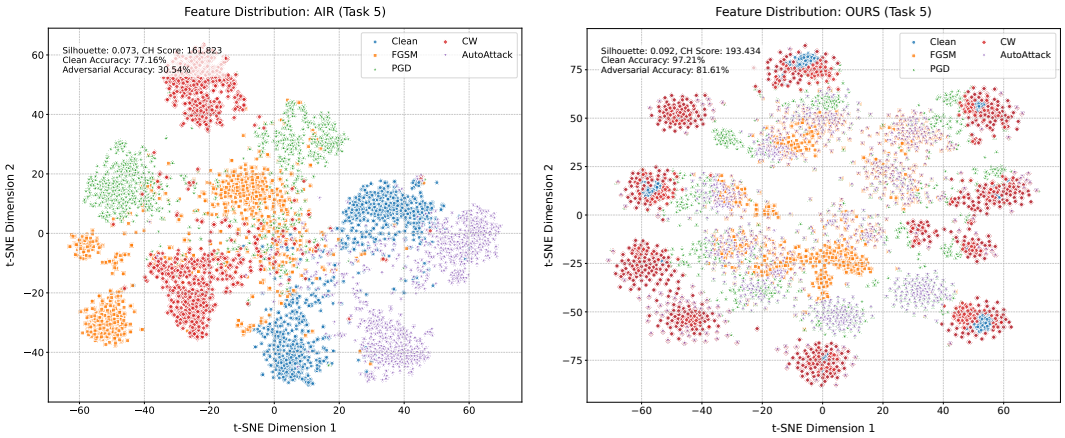
E Additional Experiment Results

E.1 Feature distribution

Fig. 4 compares the t-SNE visualizations of features extracted from clean samples by our method and AIR. Our method exhibits better clustering quality, consistent with its 20.22% advantage in clean accuracy and its higher silhouette and Calinski–Harabasz (CH) scores, which exceed those of AIR by 0.187 and 8.641, respectively. These results demonstrate the superior performance of our model on clean samples. They also confirm that our method does not compromise clean performance through excessive adversarial training from the outset of the first task, a practice adopted by some methods to maintain robustness against subsequent attacks.

Fig. 5 presents the feature visualizations of both models after five complete task sequences, providing further evidence of our model's robustness. Our method retains its advantage in clustering metrics, which is also apparent in the visualizations: compared with the loosely distributed clusters

Figure 5: Feature distribution of AIR and our model on CIFAR-10 after the complete None-FGSM-PGD-CW-AutoAttack task sequence (task 5).



produced by AIR, our model yields more compact clusters with greater inter-class separation. This greater separation implies that disrupting the model's classification requires greater adversarial effort and stronger attacks, consistent with our findings in the main paper. These advantages are also reflected in the quantitative results. On the final task, our method maintains a 20.05% lead in clean accuracy over AIR, demonstrating its strong resistance to forgetting. Our model also achieves an adversarial accuracy of 81.61%, exceeding AIR by 51.07%. Together, these results show that the robustness of our model is not merely explained by its underlying mechanisms but is also supported by strong empirical evidence at the level of feature distributions.

Table 5: The classification accuracy of various models on CIFAR-10 under the unknown JSMA-DeepFool-CW task sequence.

Method	**CIFAR-10**			
	JSMA	**DeepFool**	**CW**	**Average**
AIR	25.48	6.58	4.22	12.09
CL-LoRA	61.84	39.54	9.22	36.87
OURS	**80.75**	**51.24**	**34.18**	**55.39**

E.2 Robustness under unknown attack sequences

To assess generalization to attacks not encountered during training, we compare our framework with the replay-based AIR and parameter-efficient CL-LoRA methods discussed in the Introduction. Following training on the None–FGSM–PGD task sequence, we evaluate the models on the unseen JSMA–DeepFool–CW attack sequence without any additional training. Table 5 reports the classification accuracy under each attack and the average adversarial accuracy across the sequence.

Our framework achieves the highest accuracy under all three unseen attacks, with an average adversarial accuracy of 55.39%, compared with 12.09% for AIR and 36.87% for CL-LoRA. Although CL-LoRA consistently outperforms AIR, our method further improves accuracy under JSMA from 61.84% to 80.75% and under DeepFool from 39.54% to 51.24%. The largest improvement over CL-LoRA occurs under CW, which produces the lowest accuracy for all three methods in this evaluation. While AIR and CL-LoRA achieve only 4.22% and 9.22%, respectively, our framework reaches 34.18%. These improvements across all three attacks provide empirical evidence that our framework generalizes more effectively to the evaluated unseen attacks than both the replay-based AIR method and the parameter-efficient CL-LoRA method.

Table 6: 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

Table 7: Accuracy throughout continual adaptation on CIFAR-10. Each row evaluates the model after completing the indicated training task. AA denotes AutoAttack. The first five rows report accuracy (%); the last two rows report the final change from initial acquisition (Δ_i) and the nonnegative decline from the best preceding checkpoint (F_i^{peak}), both in percentage points (pp). Dashes indicate tasks not yet introduced or retention statistics that are inapplicable to the final task.

Training completed	Clean	FGSM	PGD	CW	AA
$\mathcal{T}_1$: Clean	97.44	–	–	–	–
$\mathcal{T}_2$: FGSM	97.39	44.33	–	–	–
$\mathcal{T}_3$: PGD	96.43	83.00	86.56	–	–
$\mathcal{T}_4$: CW	97.00	73.34	79.26	82.01	–
$\mathcal{T}_5$: AA	97.21	75.85	78.99	86.60	84.98
Δ_i (pp)	-0.23	$+31.52$	-7.57	$+4.59$	–
F_i^{peak} (pp)	0.23	7.15	7.57	0.00	–

E.3 Forgetting Rate of Long Task Sequences

The results in Tab. 6 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).

E.4 Retention of Historical Adversarial Robustness

Table 7 reports the accuracy trajectory on CIFAR-10 throughout the Clean–FGSM–PGD–CW–AutoAttack training sequence. Following the notation used for the forgetting rate, $A_{i,j}$ denotes the accuracy on task $\mathcal{T}_i$ after training through task $\mathcal{T}_j$. Rows therefore correspond to training stages j, whereas columns correspond to evaluation tasks i. At each stage, the complete classifier is evaluated on all tasks encountered so far using the protocol in Appendix D. Adversarial examples are regenerated against the classifier at the corresponding checkpoint, with attack configurations held fixed across stages. The table presents one training trajectory, rather than the three-run averages reported in Table 1.

To distinguish net backward transfer from the loss of previously attained performance, we report two complementary statistics. For each historical task $i < T$, with $T = 5$, define

$$\Delta_i = A_{i,T} - A_{i,i}, \qquad F_i^{\mathrm{peak}} = \left[\max_{i \le j < T} A_{i,j} - A_{i,T} \right]_+, \tag{48}$$

where $[x]_+ = \max(x, 0)$. The first statistic measures the final change relative to the accuracy immediately after learning the task. The second measures the decline from the best preceding checkpoint,

assigning zero when final accuracy matches or exceeds that peak. Both are expressed in percentage points, separately from the relative forgetting rate. We aggregate these statistics over the historical adversarial tasks $\mathcal{H} = \{2, 3, 4\}$:

$$\mathrm{BWT}_{\mathrm{adv}} = \frac{1}{|\mathcal{H}|} \sum_{i \in \mathcal{H}} \Delta_i = +9.51 \text{ pp}, \qquad F_{\mathrm{adv}}^{\mathrm{peak}} = \frac{1}{|\mathcal{H}|} \sum_{i \in \mathcal{H}} F_i^{\mathrm{peak}} = 4.91 \text{ pp}. \tag{49}$$

The clean task is excluded to isolate adversarial retention. AutoAttack is also excluded because no subsequent training stage is available to assess its retention.

The trajectories reveal different effects of subsequent adaptation on historical attacks. FGSM accuracy increases from 44.33% immediately after FGSM training to 83.00% after PGD adaptation. It then decreases to 73.34% after CW training before recovering to 75.85% at the final stage. Its final gain of 31.52 percentage points indicates positive backward transfer relative to initial acquisition. CW accuracy likewise improves from 82.01% to 86.60% after the final training stage, yielding a gain of 4.59 percentage points. In contrast, PGD accuracy decreases from 86.56% to 79.26% after CW training and then to 78.99% after AutoAttack training, yielding a final decline of 7.57 percentage points. Thus, the positive average backward transfer reflects gains on some historical attacks that outweigh the decrease on PGD; it does not imply that every historical task improves.

Comparison with earlier peaks provides a complementary view. Although final FGSM accuracy exceeds its initial level, it remains 7.15 percentage points below its peak after PGD training. Together with the decline on PGD, this produces an average peak-based forgetting of 4.91 percentage points across the three historical adversarial tasks. Clean accuracy, meanwhile, decreases from 97.44% to 96.43% after PGD training and subsequently recovers to 97.21%, ending 0.23 percentage points below its initial value. This small final decline on clean samples does not capture the larger, attack-dependent gains and losses observed during adaptation.

These results provide direct evidence of historical adversarial performance beyond clean-task retention. Under the specified evaluation protocol, subsequent adaptation improves final performance on FGSM and CW relative to initial acquisition, while retaining substantial PGD accuracy with a measurable decline. The observed fluctuations also distinguish preservation of frozen expert configurations from preservation of the complete classifier's accuracy, consistent with the scope of Proposition 4.1. The trajectory supports positive average backward transfer alongside nonzero forgetting, rather than uniform or lossless retention across attacks.

Table 8: Ablation study of our model on CIFAR-10 under the unknown JSMA-DeepFool-CW task sequence.

Method	**CIFAR-10**			
	JSMA	**DeepFool**	**CW**	**Average**
LoRA Baseline	70.13	43.51	8.52	40.72
LoRA + LREERC	74.08	47.92	15.88	45.96
LoRA + LREERC + SSC	77.74	48.64	29.78	52.05
LoRA + LREERC + UPC	77.20	50.16	33.30	53.55
LoRA + LREERC + DCAPA	**80.75**	**51.24**	**34.18**	**55.39**

E.5 Unseen AutoAttack Evaluation of the Full Predictor

Experimental setting. We evaluate generalization to an unseen attack ensemble using the checkpoints obtained after the four-stage sequence $\mathrm{Clean} \to \mathrm{FGSM} \to \mathrm{PGD} \to \mathrm{CW}$ on CIFAR-10. Both OURS and CL-LoRA He et al. (2025) are trained for five epochs on the clean task and one epoch on each subsequent adversarial task, following the remaining training settings described in Appendix D. Evaluation uses the checkpoint saved immediately after the CW stage, before any adaptation to the AutoAttack task. All model parameters, expert compositions, and class prototypes remain fixed throughout evaluation. This setting measures robustness acquired from the preceding tasks when the model encounters AutoAttack without additional training.

Evaluation data and threat model. We use the complete CIFAR-10 test set of $10{,}000$ images, comprising $1{,}000$ images per class. The same images are evaluated under clean conditions and at every perturbation budget for both methods. Each image is resized to 224×224 and con-

Table 9: Clean and AutoAttack accuracies (%) on the complete CIFAR-10 test set after $\mathrm{Clean} \rightarrow \mathrm{FGSM} \rightarrow \mathrm{PGD} \rightarrow \mathrm{CW}$ training. AutoAttack is unseen during training of these checkpoints. Perturbations are constrained in the resized 224×224 input space. Each budget is evaluated independently against the full frozen predictor. Bold highlights the higher accuracy where the methods differ.

Method	**Clean**	**AutoAttack (ℓ_∞)**			
		$\epsilon = 1/255$	$\epsilon = 2/255$	$\epsilon = 4/255$	$\epsilon = 8/255$
CL-LoRA	**97.90**	0.00	0.00	0.00	0.00
OURS	97.31	**3.36**	**1.74**	0.00	0.00

verted to a floating-point tensor in $[0, 1]$, without stochastic test-time augmentation. Adversarial perturbations are applied after resizing and satisfy $\|\widetilde{\mathbf{x}} - \mathbf{x}\|_\infty \leq \epsilon$, with $\widetilde{\mathbf{x}} \in [0, 1]^{3\times224\times224}$ and $\epsilon \in \{1/255, 2/255, 4/255, 8/255\}$. The resulting accuracies therefore describe robustness in the resized input space and are not directly interchangeable with CIFAR-10 evaluations that constrain perturbations in the original 32×32 space.

Attack configuration. We use the official **standard** version of AutoAttack Croce & Hein (2020), comprising untargeted APGD-CE, targeted APGD-DLR, targeted FAB, and Square Attack. APGD-CE uses 100 iterations and one restart. Targeted APGD-DLR and targeted FAB each use 100 iterations per target, nine target classes, and one restart. Square Attack uses one restart and a maximum of 5,000 queries per example. These settings remain unchanged across perturbation budgets and methods. We fix the evaluation seed to 42 and use an attack batch size of 16. At each budget, the official evaluator applies the attack components sequentially and forms new batches from the examples not yet misclassified. Each budget starts independently from the original clean images.

Full-predictor evaluation. Attacks are generated separately against each method's complete saved predictor, including all historical experts, their saved compositions, and the corresponding class prototypes. Classification follows the original class-wise maximum of normalized feature–prototype similarities. The gradient-based components differentiate through this complete computation, including the maximum operation, while Square Attack queries the same predictor. Thus, attack generation and candidate verification use the model actually evaluated, rather than an isolated expert or a separate surrogate classifier. Computation uses FP32 without mixed precision or TF32.

Accuracy measurement. An example is counted as robust only if it is correctly classified when clean and remains correctly classified after the complete AutoAttack evaluation at the specified budget. Clean errors remain in the denominator and count as failures. Let $\widehat{y}_{\mathrm{CW}}$ denote the class predicted by the frozen model after the CW stage, and let $\widetilde{\mathbf{x}}_j^\epsilon$ denote the corresponding example returned by AutoAttack. We report

$$\widehat{\mathrm{RA}}(\epsilon) = \frac{100}{N} \sum_{j=1}^{N} \mathbf{1}[\widehat{y}_{\mathrm{CW}}(\mathbf{x}_j) = y_j]\, \mathbf{1}\big[\widehat{y}_{\mathrm{CW}}(\widetilde{\mathbf{x}}_j^\epsilon) = y_j\big]\,, \qquad N = 10{,}000. \tag{50}$$

This statistic measures accuracy against the complete attack suite; it is not an average of the four component accuracies. Results are reported separately for each budget, without averaging or accumulating failures across budgets. All entries below are single-run measurements.

Results and interpretation. Together with the improvements under CW reported in Table 1, Table 9 shows that the benefit of our framework also extends to an attack ensemble not encountered during training of the evaluated checkpoints. At $\epsilon = 1/255$ and $2/255$, OURS achieves accuracies of 3.36% and 1.74%, respectively, whereas CL-LoRA attains 0.00% at both budgets. These gains correspond to 336 and 174 test examples remaining correctly classified after the complete AutoAttack evaluation. Moreover, this setting is particularly demanding because the frozen predictor is directly evaluated by complementary gradient-based and query-based attacks without further adaptation. Although the absolute gains are modest, retaining nonzero accuracy where CL-LoRA is fully compromised provides evidence of improved robustness to an unseen attack ensemble at small perturbation budgets. The result complements the CW improvements by showing a limited but measurable benefit beyond the attacks used for continual adaptation.

This advantage remains budget-dependent. Both methods attain 0.00% at $\epsilon = 4/255$ and $8/255$, indicating that the acquired robustness does not withstand the larger perturbation budgets considered here. The findings therefore support improved small-budget generalization to unseen AutoAttack, rather than broad robustness across all evaluated budgets.

E.6 Ablation study under unknown attack sequences

To further assess how each module affects our model's generalization to unseen attacks, we conducted an ablation study. Specifically, we trained the model on the None–FGSM–PGD task sequence and evaluated it on the unseen JSMA–DeepFool–CW attack sequence without any additional training. We recorded the classification accuracy under each attack and the final average adversarial accuracy. JSMA was run in batch mode for 30 iterations, modifying 16 pixels per step; DeepFool was run in batch mode for 20 steps with an overshoot parameter of 0.02; and the Carlini–Wagner (CW) attack was run for 50 steps.

The results in Tab. 8 show that each module contributes to the model's generalization to unseen attacks. Specifically, the LoRA Baseline, which includes only the basic LoRA module, performs poorly overall on unseen attacks, particularly under the strong CW attack, where its accuracy is only 8.52%. Incorporating LREERC improves CW accuracy by 7.36 percentage points through effective Expert Expansion and Residual Composition. DCAPA, however, makes the largest contribution to robustness against unseen attacks. This finding is also consistent with our analysis in the main paper. Incorporating its Source-Specific Channel (SSC) or Universal Perturbation Channel (UPC) individually further improves the framework's average adversarial accuracy by 6.09 and 7.59 percentage points, respectively. These gains are again most pronounced under CW, with accuracy increases of 13.90 and 17.42 percentage points, respectively, further highlighting the essential role of these mechanisms in improving generalization. Finally, the full model incorporating both LREERC and DCAPA further raises the average adversarial accuracy on unseen attacks to 55.39%, achieving the highest overall robustness.

Table 10: Average classification accuracy of our model on CIFAR-10 under different training epoch budgets.

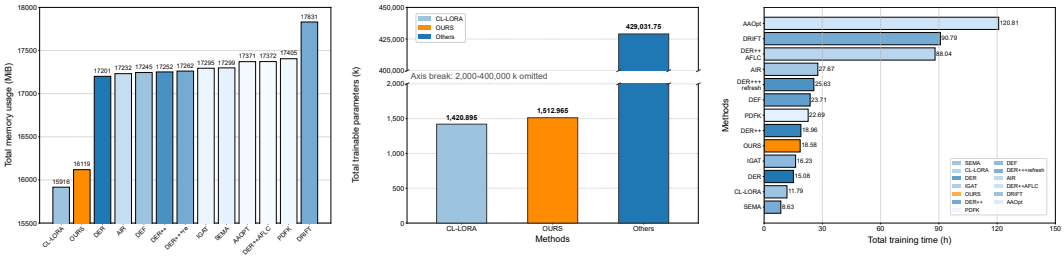
Epoch (Clean)	**CIFAR-10**				
	None	**FGSM**	**PGD**	**CW**	**AutoAttack**
1	97.10	33.51	2.90	81.43	7.50
5	97.44	**75.85**	78.99	**86.60**	**84.98**
10	**98.03**	73.57	**79.05**	82.19	84.56

E.7 Training Epoch Ablation Results

We conducted an ablation study to examine how the number of training epochs on the first task, consisting of clean samples, affects model performance. We did not perform a corresponding ablation on subsequent attack tasks because our continual learning protocol is designed to assess robustness in a realistic setting, where a model first learns from clean samples and then encounters a sequence of adversarial attacks. In this setting, the budget for learning, adaptation, and model updates is limited, so training is restricted to one epoch per subsequent attack task. Allowing unrestricted training would give the model sufficient time to learn each new attack and converge, undermining the purpose of this protocol.

As shown in Tab. 10, although the pretrained ViT enables good performance on clean samples after only 1 epoch of training, our LoRA-based expert architecture has not yet learned sufficiently rich representations of these samples, limiting its performance under subsequent attacks. Increasing the number of epochs to 5, consistent with our default protocol, resolves this limitation: adequate training of the initial expert on clean samples enables effective knowledge sharing during subsequent tasks. However, increasing the number of epochs does not necessarily yield continued performance gains. Extending training from 5 to 10 epochs improves accuracy on clean samples by only 0.59%, while overall performance under subsequent attacks remains largely unchanged. Accuracy under FGSM, CW, and AutoAttack even decreases by 2.28%, 4.41%, and 0.42%, respectively. These results show that 5 epochs provide sufficient training for our model under the adopted configura-

tion. Further increasing the number of epochs yields no meaningful performance gains and instead substantially increases computational cost.



(a) Memory Usage (b) Total Trainable Parameters (c) Total Training Time

Figure 6: Computational resource comparison across methods. (a) Total memory usage, in MiB. (b) Total trainable parameters, in thousands (k). Methods with identical parameter totals are grouped as Others; the broken y-axis separates the lightweight methods from the full-backbone group. (c) Total training time, in hours. OURS is highlighted in orange.

E.8 Computational Cost

Fig. 6 compares the resource consumption of our method with the evaluated baselines in terms of total memory usage, total trainable parameters, and total training time over the continual adversarial learning sequence. These measurements characterize the computational cost associated with the robustness gains reported in our experiments.

E.8.1 Total memory usage

As shown in Fig. 6a, our model consumes 16,119 MiB, the second-lowest memory footprint among the thirteen evaluated methods. Compared with CL-LORA, which requires 15,916 MiB, the additional memory consumption is 203 MiB, corresponding to an increase of only 1.28%. Our model requires less memory than every other evaluated baseline. In particular, it reduces memory consumption by 1,253 MiB (7.21%) relative to DER++AFLC and by 1,712 MiB (9.60%) relative to DRIFT. These results show that our method maintains a memory footprint close to the most memory-efficient baseline while supporting its stronger overall adversarial robustness.

E.8.2 Total trainable parameters

Fig. 6b reports the total trainable parameter counts accumulated over the task sequence, expressed in thousands (k). Our method requires 1,512.965 k, compared with 1,420.895 k for CL-LORA and 429,031.750 k for each full-backbone baseline represented by Others. Our method therefore reduces the cumulative trainable parameter count by 99.65% relative to the full-backbone group. Compared with CL-LORA, the increase is 92.070 k, or 6.48%. Thus, the robustness gains are obtained with a trainable parameter budget close to that of CL-LORA and substantially below that of the full-backbone baselines. These cumulative counts describe trainable parameters across tasks, rather than the number of unique parameters stored in the final model.

E.8.3 Total training time

As shown in Fig. 6c, our method completes training in 18.58 h, yielding the fifth-shortest duration among the thirteen evaluated methods. It is slightly faster than DER++, which requires 18.96 h, and substantially faster than DER++AFLC, DRIFT, and AAOpt, which require 88.04 h, 90.79 h, and 120.81 h, respectively. Relative to these three methods, our framework reduces training time by 78.90%, 79.54%, and 84.62%. In particular, it uses only 21.10% of the training time required by DER++AFLC. These measurements establish that our method achieves its robustness results with a lower training-time budget than eight of the twelve baselines.

E.8.4 RESOURCE-COST EFFICIENCY TRADE-OFF

The comparison with CL-LORA illustrates the trade-off across resource dimensions. Our method increases memory usage and total trainable parameters by 1.28% and 6.48%, respectively, while requiring an additional 6.78 h of training, a 57.49% increase. The additional cost relative to CL-LORA is therefore concentrated in training time. By comparison, our method reduces all three reported resource costs relative to DER++AFLC. This resource profile combines near-minimal memory consumption and a compact trainable parameter budget with a training duration below that of most evaluated baselines.

E.8.5 EFFICIENCY–ROBUSTNESS TRADE-OFF

The robustness experiments show that our method achieves the best overall robustness across the evaluated known and unknown attack settings. Considered together with the resource measurements, this result demonstrates that strong continual adversarial defense can be achieved within a competitive computational budget. Relative to CL-LORA, the stronger overall robustness entails additional training time, accompanied by limited increases in memory usage and trainable parameters. Relative to DER++AFLC and DRIFT, our method combines stronger overall robustness with lower memory consumption and substantially shorter training. These comparisons support a favorable efficiency–robustness trade-off under the evaluated continual learning protocol, balancing robustness to known and unknown attacks with the resources required to acquire and maintain that robustness.