From Standard to Robust: A Universal Framework for Continual Adversarial Defense

Qian Wang[1], **Hefei Ling**[1,*], **Yingwei Li**[2], **Qihao Liu**[2] and **Ning Yu**[3]
[1]School of Computer Science and Technology, Huazhong University of Science and Technology, China
[2]Johns Hopkins University, USA
[3]Eyeline Studios, USA
{yqwq1996, lhefei}@hust.edu.cn, {yingwei.li, qliu45}@jhu.edu, ningyu.hust@gmail.com

Abstract

Continual adversarial defense (CAD) aims to defend target models against continuously emerging attacks. However, existing CAD methods typically rely on maintaining large amounts of replay data or multiple expert modules to mitigate catastrophic forgetting, or suffer from reduced robustness against adversarial examples or degraded performance on clean images. As a result, they often fail to provide clear advantages over standalone robust models. To address these limitations, we formulate four fundamental principles for CAD: (1) *continual adaptation to new attacks without catastrophic forgetting*, (2) *few-shot adaptation*, (3) *memory-efficient adaptation*, and (4) *high classification accuracy on both clean and adversarial data*. Guided by these principles, we explore and integrate cutting-edge techniques from continual learning, few-shot learning, and ensemble learning, and propose **Universal Continual Adversarial Defense** (**UCAD**), a universal framework that enables both standard and robust models to perform effective defense under the CAD setting. Extensive experiments validate the effectiveness of UCAD against multi-stage adversarial attacks and demonstrate significant improvements over a wide range of baseline methods. Moreover, we observe that as the number of encountered attacks increases, UCAD becomes increasingly robust, consistently enhancing the defense capability of existing robust models until saturation.

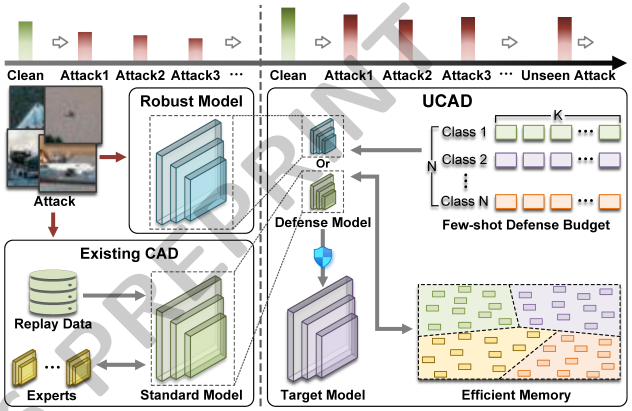

Figure 1: Existing CAD methods rely on maintaining large amounts of replay data or multiple expert modules but fail to provide clear advantages over robust models. In contrast, UCAD enables both standard and robust models to perform effective defense using efficient memory and a few-shot defense budget.

1 Introduction

Adversarial attacks [Madry *et al.*, 2017] aim to deceive deep neural networks (DNNs) by adding subtle perturbations to input images, seriously jeopardizing the reliability of DNNs, particularly in security- and trust-sensitive domains. To protect DNNs, researchers have proposed adversarial training (AT) [Zhang *et al.*, 2019] and purification techniques [Yoon *et al.*, 2021], which defend against adversarial attacks through one-shot training—where the model enters a static phase after a single defensive stage [Zhou and Hua, 2024]. Unlike the one-shot defense assumption, real-world defense systems operate in dynamic environments with increasingly aggressive and diverse attacks [Li *et al.*, 2023]. A well-matched approach to the dynamic environments is a constantly evolving defense strategy that continuously adapts to newly emerging adversarial attacks [Croce *et al.*, 2022]. However, current model adaptation defenses [Chen *et al.*, 2021] focus on adjusting defense strategies only in response to ongoing attacks rather than leveraging past defense experiences, leading to suboptimal performance.

To address the threat of continual attacks, continual adversarial defense (CAD) [Cho *et al.*, 2025] has been proposed to protect target models by continuously adapting defense mechanisms to newly emerging attacks. AIR [Zhou and Hua, 2024] and DDeR [Wang *et al.*, 2025] treat adversarial attacks as a sequence of tasks and aim to train robust models through lifelong learning under a white-box setting. SSEAT [Wang *et al.*, 2024b] further employs a replay mechanism together with a consistency regularization strategy to incrementally learn defenses against new attacks in a gray-box setting. However, these methods typically rely on maintaining large amounts

*Corresponding author.

of replay data or multiple expert modules to mitigate catastrophic forgetting, or suffer from reduced robustness against adversarial examples and degraded performance on clean images. As a result, they often fail to provide clear advantages over standalone robust models.

To address these limitations, we formulate four principles for CAD: (1) *Continual adaptation to new attacks without catastrophic forgetting.* (2) *Few-shot adaptation.* (3) *Memory-efficient adaptation.* (4) *High classification accuracy on both clean and adversarial data.* These principles require the defender to continually adapt to new attacks using a few-shot defense budget and efficient memory, while maintaining high classification accuracy on both clean and adversarial data and avoiding catastrophic forgetting.

Guided by these principles, we propose **Universal Continual Adversarial Defense** (**UCAD**). As shown in Figure 1, UCAD enables both standard and robust models to perform effective defense under the CAD setting. In the first stage, we use adversarial data from an initial attack (e.g., PGD) to adversarially train an initial defense model that is specifically designed to classify adversarial examples and complement the target model's predictions. Inspired by continual learning (CL), we treat the new classes introduced by each attack as incremental and expand the classifier accordingly, followed by fine-tuning the expanded layer using a few-shot defense budget. To address the overfitting issue inherent in few-shot fine-tuning [Zhou *et al.*, 2022], we augment existing robust models with a lightweight side branch for subsequent adaptation, and pre-assign virtual classes to compress the embeddings of past attacks, thereby reserving embedding space in the defense model for future attacks. To achieve memory efficiency, we employ prototype augmentation [Zhu *et al.*, 2022], which preserves decision boundaries from previous stages without explicitly storing any defense budget. Simultaneously, a lightweight model is employed to ensemble the defense model and the target model by estimating reliable logits for the input, thereby maintaining high classification accuracy on both clean and adversarial data.

Extensive experiments validate the effectiveness of UCAD against multi-stage adversarial attacks and demonstrate significant improvements over a wide range of baseline methods. Moreover, we observe that UCAD's defense performance tends to saturate as the number of encountered attacks increases, suggesting that once adapted to a sufficiently diverse attack set, it can act as a persistent defense mechanism applicable to both standard and robust models.

Our main contributions can be summarized as follows:

- To address the limitations of existing CAD methods, we formulate four principles for CAD: continual adaptation without catastrophic forgetting, few-shot adaptation, memory-efficient adaptation, and high classification accuracy on both clean and adversarial data.
- Guided by these principles, we explore and integrate cutting-edge techniques from continual learning, few-shot learning, and ensemble learning, and propose **Universal Continual Adversarial Defense** (**UCAD**), a universal framework that enables both standard and robust models to perform effective defense under the CAD setting.
- Extensive experiments validate the effectiveness of UCAD against multi-stage adversarial attacks and demonstrate significant improvements over a wide range of baseline methods. Moreover, we observe that as the number of encountered attacks increases, UCAD becomes increasingly robust, consistently enhancing the defense capability of existing robust models until saturation.

2 Related Work

2.1 Adversarial Defense

Researchers have developed various general defenses against adversarial attacks [Wu *et al.*, 2021; Lin *et al.*, 2019], including adversarial training [Grathwohl *et al.*, 2020; Laidlaw *et al.*, 2020; Jiang *et al.*, 2023], which enhances the robustness of the target model by training it with adversarial examples; data purification [Hill *et al.*, 2020; Pei *et al.*, 2025; Nie *et al.*, 2022], which removes potential adversarial perturbations or noise that could deceive the model; and dynamic model adaptation [Kang *et al.*, 2021; Yin *et al.*, 2024; Li *et al.*, 2022], which continuously adjusts model parameters, states, or activations when encountering attacks to enhance robustness. However, these methods still show limited robustness and often degrade classification performance on both adversarial and clean data. In contrast, our UCAD framework effectively counters a wide spectrum of stage-wise emerging attacks while maintaining robust clean-data performance.

2.2 Continual Learning

Continual learning (CL) [Liu *et al.*, 2022] aims to learn from a sequence of new classes while preserving knowledge of previously learned ones. Many works have been proposed for CL [Rebuffi *et al.*, 2017; Zhu *et al.*, 2021a; Zhu *et al.*, 2021b]. In recent years, some methods have attempted to address the CL problem without relying on data preservation (i.e., non-exemplar methods) [Zhu *et al.*, 2022], as well as few-shot scenarios [Zhang *et al.*, 2021; Zhou *et al.*, 2022], in which only a small number of samples from new classes are available. Attempting to break the one-shot defense assumption, continual adversarial defense (CAD) methods [Cho *et al.*, 2025; Zhou and Hua, 2024; Wang *et al.*, 2024b; Wang *et al.*, 2025] have been proposed to protect target models by continuously adapting defense mechanisms to newly emerging attacks. In this paper, we formulate four principles for CAD, which require the defense mechanism to leverage a limited defense budget and memory-efficient adaptation strategies.

3 Principles for Continual Adversarial Defense

Task Definition. Continual Adversarial Defense (CAD) is formulated under the N-way K-shot classification paradigm. In the gray-box setting [Taran *et al.*, 2019], the attacker has access to the architecture of the target model $f_t : \mathcal{X} \to \mathbb{R}^N$, but not to the defense mechanism, where $\mathcal{X}$ denotes the image space. In the white-box setting, the attacker has access to both the target model and the defense mechanism. The target model is trained on $\mathcal{D}_{\text{train}}$ and evaluated on $\mathcal{D}_{\text{test}}$. The defender is allowed to use $\mathcal{D}_{\text{train}}$ and has full knowledge of the

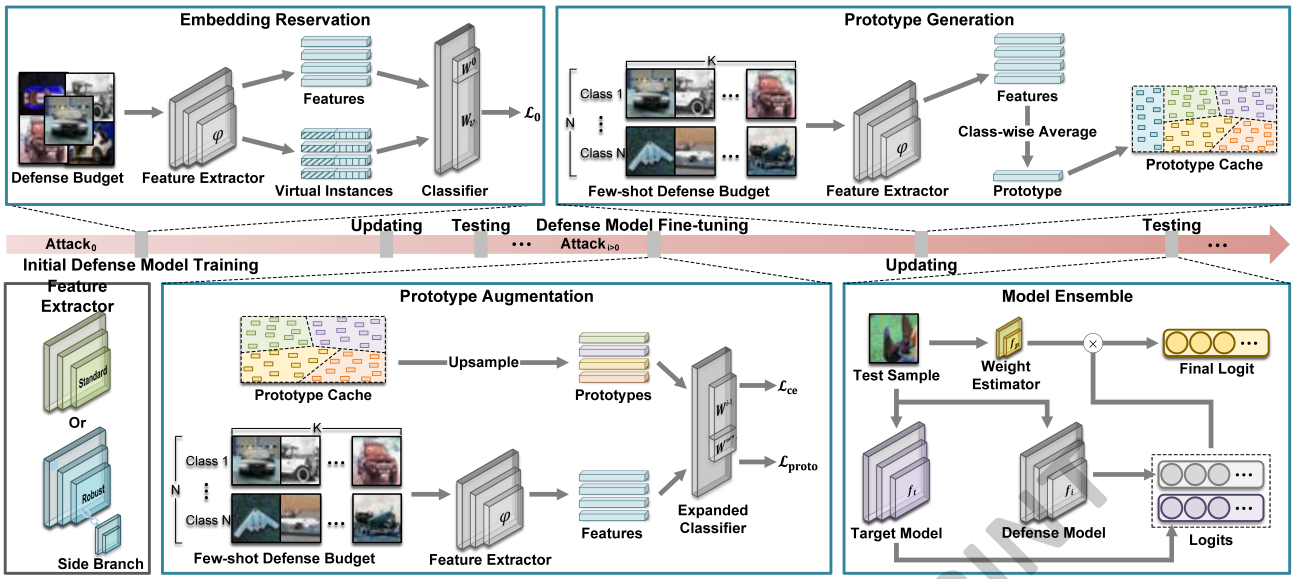

Figure 2: Illustration of UCAD. In stage 0, we reserve embedding space while training the initial defense model to mitigate future overfitting. In stage i ($i > 0$), we fine-tune the defense model using a few-shot defense budget, and store class-wise prototypes in a cache to achieve memory efficiency. Finally, the target model and the defense model are ensembled via a weight estimator to enable robust classification.

initial attack $A_0(\cdot)$ at stage 0. At stage i ($i = 1, 2, \cdots, T$), new attacks $A_i(\cdot)$ emerge, and the defender can utilize a defense budget $\mathcal{A}^i_{\text{train}} = \{(\mathbf{x}^i_{\text{adv}}, y) | \mathbf{x}^i_{\text{adv}} = A_i(\mathbf{x}), (\mathbf{x}, y) \in \mathcal{D}_{\text{train}}\}$[1] consisting of $N \times K$ samples—i.e., K samples per class for N classes—to adapt to new attacks, where y denotes the ground-truth label of sample $\mathbf{x}$. Evaluations are conducted on $\mathcal{D}_{\text{test}}$ and $\{\mathcal{A}^k_{\text{test}}\}_{k=0,1,\ldots,i}$ at each stage i, where $\mathcal{A}^i_{\text{test}} = \{(\mathbf{x}^i_{\text{adv}}, y) | \mathbf{x}^i_{\text{adv}} = A_i(\mathbf{x}), (\mathbf{x}, y) \in \mathcal{D}_{\text{test}}\}$.

Principles for CAD. Taking practical considerations into account, a defense mechanism should satisfy the following principles:

Principle 1. *Continual adaptation to new attacks without catastrophic forgetting.*

The defense mechanism must be capable of adapting to a range of new attacks that emerge over time while preserving knowledge of previously encountered ones.

Principle 2. *Few-shot adaptation.*

An increase in the defense budget indicates a heightened risk of attacks on the target model. Therefore, it is crucial to initiate adaptation before the defense budget grows excessively.

Principle 3. *Memory-efficient adaptation.*

Over time, the growing influx of attacks leads to the accumulation of the defense budget, which may eventually exceed memory limits, posing practical challenges under constrained capacity.

Principle 4. *High classification accuracy on both clean and adversarial data.*

A robust defense should not compromise performance on benign inputs for the sake of adversarial robustness. Therefore, ensuring high classification accuracy on both clean and adversarial data is paramount.

4 Universal Continual Adversarial Defense Framework

We propose **Universal Continual Adversarial Defense** (**UCAD**), a universal framework that enables both standard and robust models to perform effective defense under the CAD setting. As shown in Figure 2, UCAD comprises four key components, each designed to satisfy a corresponding CAD principle: (1) continual adaptation to new attacks to defend the target model while mitigating catastrophic forgetting; (2) embedding reservation to alleviate overfitting caused by few-shot adaptation; (3) prototype augmentation to achieve memory efficiency; and (4) model ensemble to ensure high classification accuracy on both clean and adversarial data.

4.1 Continual Adaptation to New Attacks

In response to Principle 1, we employ continual adaptation to defend against new attacks while mitigating catastrophic forgetting. At the initial stage (stage 0), an initial defense model $f_0 : \mathcal{X} \to \mathbb{R}^N$, which consists of a feature extractor $\varphi : \mathcal{X} \to \mathbb{R}^d$ and a classifier $g^0_c : \mathbb{R}^d \to \mathbb{R}^N$, is adopted to handle adversarial examples, thereby complementing the target model. The defense model can be either an off-the-shelf robust model or a standard model that is optimized under full supervision using an adversarial dataset $\mathcal{A}^0_{\text{train}} = \{(\mathbf{x}^0_{\text{adv}}, y) | \mathbf{x}^0_{\text{adv}} = A_0(\mathbf{x}), (\mathbf{x}, y) \in \mathcal{D}_{\text{train}}\}$. At stage i ($i = 1, 2, \cdots, T$), the defense model $f_i : \mathcal{X} \to \mathbb{R}^N$ adapts to new attacks using the defense budget $\mathcal{A}^i_{\text{train}}$, while

[1] In the adversarial attack formula $\mathbf{x}^i_{\text{adv}} = A_i(\mathbf{x}, y, f_t)$, we omit the target model f_t and the ground-truth label y for brevity.

mitigating catastrophic forgetting through cached prototypes P_e extracted from past attacks.

Inspired by continual learning (CL) [Liu *et al.*, 2022], we treat the new classes introduced by each attack as incremental classes and expand the classifier from $g_c^{i-1} : \mathbb{R}^d \to \mathbb{R}^{N\times i}$ to $g_c^i : \mathbb{R}^d \to \mathbb{R}^{N\times(i+1)}$ at stage i ($i = 1, 2, \cdots, T$). The parameters of the expanded classifier consist of those inherited from the previous classifier and newly initialized weights, i.e. $W^i = [W^{i-1}, W^{\text{new}}]$, where $W^i = [\mathbf{w}_1^i, \cdots, \mathbf{w}_{N_i}^i]$ denotes the parameter matrix of g_c^i. Meanwhile, the ground-truth label of each adversarial example $\mathbf{x}_{\text{adv}}^i \in \mathcal{A}_{\text{train}}^i$ is reassigned to correspond to the incremental classes introduced by the new attack [2]:

$$y^i = y + N \times i. \tag{1}$$

Due to the intrinsic similarity among certain adversarial attacks [Wang *et al.*, 2023a], a defense model trained on one attack can partially generalize to others, potentially rendering some class expansions unnecessary. Moreover, the recurrence of past attacks in later stages can also lead to unnecessary adaptation. Therefore, before fine-tuning at each stage, we evaluate the model to filter out redundant samples in the defense budget—that is, samples that are already correctly classified.

For evaluation, we focus solely on the class assignment of the input image, rather than the specific attack to which it is subjected. Therefore, the prediction for an instance $\mathbf{x}$ is:

$$y_{\text{pred}} = (\arg\max f_i(\mathbf{x})) \bmod N. \tag{2}$$

In the following section, we detail the training procedure of the initial defense model f_0. After that, we freeze the feature extractor φ and fine-tune the expanded portion of the classifier for each new attack A_i using the few-shot budget $\mathcal{A}_{\text{train}}^i$.

4.2 Embedding Space Reservation for Few-Shot Adaptation

A challenge introduced by the few-shot budget in Principle 2 is overfitting, which is a well-known issue in few-shot learning [Snell *et al.*, 2017]. To mitigate this issue, we pre-assign virtual classes to compress the embeddings of past attacks, thereby reserving embedding space in the defense model for future attacks.

First, before training the initial defense model f_0, several virtual classes $W_v = [\mathbf{w}_v^1, \cdots, \mathbf{w}_v^V] \in \mathbb{R}^{d\times V}$ are pre-assigned in the classifier [Zhang *et al.*, 2021], where $V = N \times T$ is the number of virtual classes, i.e., the reserved classes for future attacks. Accordingly, the output of the current defense model is given by $f_0(\mathbf{x}) = g(\varphi(\mathbf{x}))$, where $g : \mathbb{R}^d \to \mathbb{R}^{N\times(T+1)}$ is the new classifier parameterized by $[W^0, W_v]^\top$. After training, W_v is used to initialize the new parameters of the expanded classifier.

Second, virtual instances are constructed via manifold mixup [Verma *et al.*, 2019]:

$$\mathbf{z} = h_2(\omega h_1(\mathbf{x}_{\text{adv},r}^0) + (1-\omega)h_1(\mathbf{x}_{\text{adv},s}^0)), \tag{3}$$

where $\mathbf{x}_{\text{adv},r}^0$ and $\mathbf{x}_{\text{adv},s}^0$ belong to different classes r and s, and $\omega \sim \text{B}(\alpha, \beta)$ is a trade-off parameter, following [Zhou *et al.*, 2022]. h_1 and h_2 are decoupled hidden layers of feature extractor, i.e. $\varphi(\mathbf{x}) = h_2(h_1(\mathbf{x}))$.

Finally, the embedding space reservation is performed by training f_0 with the following loss function:

$$\begin{aligned}\mathcal{L}_0 =& F_{\text{ce}}(f_0(\mathbf{x}_{\text{adv}}^0), y^0) + \gamma F_{\text{ce}}(\mathsf{Mask}(f_0(\mathbf{x}_{\text{adv}}^0), y^0), \hat{y}) + \\ &+ F_{\text{ce}}(f_0(\mathbf{z}), \hat{y}) + \gamma F_{\text{ce}}(\mathsf{Mask}(f_0(\mathbf{z}), \hat{y}), \hat{\hat{y}}),\end{aligned} \tag{4}$$

where $\hat{y} = \arg\max_j \mathbf{w}_v^{j\top}\varphi(\mathbf{x}_{\text{adv}}^0) + N$ is the virtual class with maximum logit, and serves as a pseudo label. $\hat{\hat{y}} = \arg\max_k \mathbf{w}_k^{0\top}\mathbf{z}$ is the pseudo label among current known classes. γ is a trade-off parameter, F_{ce} represents the standard cross-entropy loss [Zhang and Sabuncu, 2018], and the function $\mathsf{Mask}(\cdot)$ masks out the logit corresponding to the ground-truth:

$$\mathsf{Mask}(f_0(\mathbf{x}_{\text{adv}}^0), y^0) = f_0(\mathbf{x}_{\text{adv}}^0) \otimes (\mathbf{1} - \text{onehot}(y^0)), \tag{5}$$

where $\otimes$ is Hadamard product and $\mathbf{1}$ is an all-ones vector.

In Eq. 4, the loss function pushes the adversarial instance $\mathbf{x}_{\text{adv}}^0$ toward its ground-truth class (item 1) and aligns it with the nearest virtual class (item 2). Simultaneously, it drives the virtual instance $\mathbf{z}$ toward the corresponding virtual class (item 3) and toward the nearest known class (item 4). When trained with $\mathcal{L}0$, the embeddings of the initial benign classes become more compact, and the embedding space for virtual classes is effectively reserved [Zhou *et al.*, 2022]. This reserved space allows the defense model to be adapted more easily in future stages and alleviates overfitting caused by the few-shot budget.

When using a robust model as the defense model, we freeze its backbone, attach a lightweight side branch, replace the original classifier with a new classifier g, and perform embedding space reservation.

4.3 Prototype Augmentation for Memory Efficiency

In response to Principle 3, we adopt prototype augmentation to achieve memory efficiency.

When learning new classes, the decision boundaries of previously learned classes can change significantly, resulting in a severely biased unified classifier [Zhu *et al.*, 2021b]. To mitigate this issue, many CL methods store a subset of past data and jointly train the model with both past and current data. However, preserving the past defense budget may also lead to memory shortages.

To achieve memory efficiency, we adopt prototype augmentation [Zhu *et al.*, 2022] to preserve the decision boundaries from previous stages without storing any defense budget. Specifically, we store one prototype in the deep feature space for each class under each attack, and over-sample (i.e., U_B) the prototypes $\mathbf{p}_B$ together with their corresponding ground-truth labels y_B to match the batch size, thereby calibrating the classifier:

$$\mathbf{p}_B = U_B(P_e), \quad \mathcal{L}_{\text{proto}} = F_{\text{ce}}(g_c^i(\mathbf{p}_B), y_B), \tag{6}$$

where $P_e = \{\mathbf{p}_e^j\}_{j=0}^{N\times i}$ denotes the prototype cache, i.e., the set of class-wise average embeddings [Snell *et al.*, 2017], defined as:

$$\mathbf{p}_e^j = \frac{1}{K}\sum_{k=1}^{|\mathcal{A}_{\text{train}}^i|}\mathbb{I}(y_k^i = j)\varphi(\mathbf{x}_{\text{adv},k}^i), \tag{7}$$

[2] We omit the network parameters in formulas for brevity.

		Method	0:PGD	1:SNIM	2:BIM	3:RFGSM	4:MIM	5:DIM	6:NIM	7:VNIM	8:VMIM	Clean
		None-defense	0.62	1.31	0.00	0.00	0.00	0.00	0.08	0.00	0.00	96.42
White-box	CAD	Vanilla AT	36.21	35.79	34.52	34.24	34.36	34.62	34.24	34.21	34.23	70.18
		SSEAT [Wang *et al.*, 2024b]	51.85	46.25	45.69	45.21	45.32	44.83	44.62	44.70	44.71	72.49
		UCAD (ours)	51.74	**48.73**	**49.52**	**50.33**	**50.63**	**50.72**	**50.81**	**51.08**	**51.09**	**94.32**
		UCAD†(ours)	51.74	51.83	51.76	51.98	51.64	51.75	51.83	51.92	51.95	96.02
	Robust Models	TRADES [Zhang *et al.*, 2019]	52.65	51.62	52.66	52.65	51.73	51.86	51.64	52.26	52.25	84.92
		TRADES + UCAD	51.33	**51.78**	**53.25**	**53.42**	**53.44**	**53.73**	**53.92**	**54.23**	**54.24**	**90.53**
		GAIRAT [Zhang *et al.*, 2020]	59.25	58.90	59.22	59.20	58.84	58.92	58.94	59.17	59.18	89.36
		GAIRAT + UCAD	58.65	**59.02**	**59.43**	**59.86**	**60.17**	**60.42**	**60.76**	**61.38**	**61.42**	**92.39**
		DMAT [Wang *et al.*, 2023b]	67.25	67.01	67.22	67.29	66.86	66.92	66.98	67.19	67.16	92.44
		DMAT + UCAD	66.53	**67.08**	**67.45**	**67.97**	**68.26**	**68.55**	**68.94**	**69.06**	**69.12**	**94.28**
Gray-box	CAD	Vanilla AT	42.51	40.93	41.82	41.68	41.29	40.74	39.66	40.83	40.77	70.18
		SSEAT [Wang *et al.*, 2024b]	78.62	74.37	75.25	75.19	73.70	72.94	73.68	72.71	72.58	81.92
		UCAD (ours)	**94.52**	**85.41**	**93.44**	**94.72**	**94.98**	**93.74**	**93.74**	**95.07**	**93.68**	**95.82**
		UCAD†(ours)	94.52	95.51	95.60	94.83	95.04	94.32	95.10	95.14	95.71	95.98
	Robust Models	TRADES [Zhang *et al.*, 2019]	55.84	60.22	57.43	55.90	55.82	57.09	57.07	58.85	57.34	84.82
		TRADES + UCAD	55.47	**62.81**	**63.74**	**63.85**	**65.08**	**64.73**	**65.19**	**65.47**	**65.51**	**90.62**
		GAIRAT [Zhang *et al.*, 2020]	66.93	70.42	66.44	66.91	66.73	66.49	66.51	68.63	66.42	89.38
		GAIRAT + UCAD	65.32	68.81	**70.25**	**70.67**	**71.54**	**71.92**	**72.33**	**72.58**	**72.61**	**92.80**
		DMAT [Wang *et al.*, 2023b]	85.77	85.42	85.69	85.70	85.33	85.31	85.82	84.84	85.03	92.44
		DMAT + UCAD	83.63	**85.51**	**85.72**	**85.92**	**86.71**	**87.24**	**87.67**	**87.92**	**87.90**	**93.76**

Table 1: Classification accuracy against ℓ_∞ attacks at each stage on CIFAR-10 in the white-box setting and the gray-box setting. The column "Clean" denotes the standard accuracy on clean images. **UCAD†** denotes Premium-UCAD, which serves as the oracle upper bound of UCAD. Robust models and vanilla AT in the gray-box setting are evaluated under a black-box setting, since these models themselves serve as the attack targets. **Bold** indicates the best result among the corresponding baselines.

where $\mathbb{I}(\cdot)$ is the indicator function, K denotes the budget size per class, and $(\mathbf{x}^i_{\text{adv},k}, y^i_k)$ represents the k-th sample in $\mathcal{A}^i_{\text{train}}$.

After each adaptation, prototypes of the current attack are added to the cache P_e. At each stage, the defense model is adapted using the following loss function:

$$\mathcal{L}_f = \mathcal{L}_{\text{ce}} + \lambda \mathcal{L}_{\text{proto}}, \text{ where } \mathcal{L}_{\text{ce}} = F_{\text{ce}}(f_i(\mathbf{x}^i_{\text{adv}}), y^i). \quad (8)$$

4.4 Model Ensemble for Clean Data Classification

To satisfy Principle 4, which requires high classification accuracy on both clean and adversarial data, we introduce a model ensemble as the final component of UCAD. We extend Ensemble Adversarial Training (EAT) [Tramèr *et al.*, 2018] to the CAD scenario by introducing a lightweight weight estimator f_p that adaptively fuses the logits of the defense model and the target model. This design ensures strong classification performance on clean data.

We adopt self-perturbation [Wang *et al.*, 2023a] to train a robust weight estimator f_p at the first stage. When trained with self-perturbation, the weight estimator becomes agnostic to specific attacks and can effectively distinguish between clean and adversarial images. The weight estimator f_p outputs a weight vector $\mathbf{w} \in \mathbb{R}^2$ to ensemble the defense model and the target model:

$$\text{logit}_i(\mathbf{x}) = \mathbf{w} \cdot [f_t(\mathbf{x}), f_i(\mathbf{x})]^\top, \text{ where } \mathbf{w} = f_p(\mathbf{x}). \quad (9)$$

In this manner, clean and adversarial inputs are routed to their corresponding models, thereby increasing the likelihood of correct classification.

5 Experiment

To evaluate the performance of UCAD against various adversarial attacks under the CAD setting, we conduct extensive experiments on CIFAR-10 and ImageNet-100 [Tian *et al.*, 2020] and compare UCAD with multiple baseline methods. For evaluation, we use two metrics: (1) classification accuracy, including robust accuracy on adversarial examples and standard accuracy on clean images after each adaptation step, to assess performance; and (2) Average Incremental Accuracy (AIAcc) [Rebuffi *et al.*, 2017], which averages accuracy over attacks encountered up to the current stage to measure resistance to catastrophic forgetting.

5.1 Experiment Settings

Attacks PGD-ℓ_∞ [Madry *et al.*, 2017] acts as the initial attack. Subsequently, eight ℓ_∞ attacks—namely SNIM [Lin *et al.*, 2019], BIM [Kurakin *et al.*, 2016], RFGSM [Tramèr *et al.*, 2018], MIM [Dong *et al.*, 2018], DIM [Wu *et al.*, 2021], NIM [Lin *et al.*, 2019], VNIM, and VMIM [Wang and He, 2021]—compose the attack pool for evaluating defense performance. In addition, another attack pool consisting of eight adversarial attacks: ℓ_∞ attacks VMIM and SNIM, ℓ_2 attacks CW [Carlini and Wagner, 2017] and DeepFool [Moosavi-Dezfooli *et al.*, 2016], ℓ_1 attacks EAD and EADEN [Chen *et al.*, 2018], and ℓ_0 attacks OnePixel [Su *et al.*, 2019] and SparseFool [Modas *et al.*, 2019], is used to evaluate UCAD's effectiveness in mitigating catastrophic forgetting.

Defense Baselines To evaluate defense performance in the white-box setting, we compare UCAD with vanilla AT under the CAD setting and a CAD method SSEAT [Wang *et al.*, 2024b]. In addition, we evaluate existing robust models and

		Method	0:PGD	1:SNIM	2:BIM	3:RFGSM	4:MIM	5:DIM	6:NIM	7:VNIM	8:VMIM	Clean
None-defense			0.21	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	89.76
White-box	CAD	Vanilla AT	20.18	19.53	20.12	20.16	19.24	19.26	19.45	19.58	19.55	46.19
		SSEAT [Wang *et al.*, 2024b]	28.32	24.51	24.62	24.47	24.18	24.02	23.66	23.52	23.58	50.36
		UCAD (ours)	28.27	**26.44**	**27.83**	**27.76**	**27.85**	**27.94**	**27.99**	**28.06**	**28.10**	**86.47**
		UCAD†(ours)	28.27	28.23	28.35	28.29	28.19	28.23	28.30	28.29	28.28	88.92
	Robust Models	TRADES [Zhang *et al.*, 2019]	31.25	30.87	31.16	31.12	30.65	30.72	30.93	31.08	31.10	52.27
		TRADES + UCAD	31.02	**31.15**	**31.43**	**31.72**	**31.80**	**31.98**	**32.42**	**32.61**	**32.65**	**57.49**
		GAIRAT [Zhang *et al.*, 2020]	46.53	46.28	46.38	46.42	45.96	46.05	46.17	46.43	46.49	65.22
		GAIRAT + UCAD	46.20	**46.43**	**46.62**	**46.83**	**46.92**	**47.01**	**47.25**	**47.51**	**47.83**	**69.27**
		DMAT [Wang *et al.*, 2023b]	56.27	56.05	56.28	56.31	56.11	56.08	56.12	56.25	56.26	71.36
		DMAT + UCAD	56.09	**56.12**	**56.35**	**56.57**	**56.63**	**57.12**	**57.40**	**57.74**	**57.92**	**74.28**
Gray-box	CAD	Vanilla AT	39.65	38.57	39.42	39.58	39.29	38.83	37.92	38.70	39.06	65.37
		SSEAT [Wang *et al.*, 2024b]	74.86	71.63	72.44	72.08	70.06	69.22	69.87	68.85	69.13	75.42
		UCAD (ours)	**83.72**	**83.44**	**82.04**	**82.53**	**83.46**	**82.03**	**81.90**	**82.24**	**81.88**	**88.13**
		UCAD†(ours)	83.72	83.85	83.96	82.64	84.33	83.76	84.22	82.24	82.04	88.81
	Robust Models	TRADES [Zhang *et al.*, 2019]	46.53	45.52	49.63	46.22	48.04	47.68	46.69	48.20	49.48	76.21
		TRADES + UCAD	46.27	**46.82**	**49.75**	**50.21**	**51.07**	**51.86**	**52.36**	**52.82**	**53.34**	**81.42**
		GAIRAT [Zhang *et al.*, 2020]	59.84	60.15	58.43	59.22	58.77	58.92	58.86	59.73	59.84	74.12
		GAIRAT + UCAD	59.02	60.23	**60.52**	**60.71**	**61.03**	**61.45**	**61.82**	**62.55**	**62.69**	**78.96**
		DMAT [Wang *et al.*, 2023b]	64.31	65.10	66.17	65.86	65.43	64.44	64.13	65.82	65.63	75.88
		DMAT + UCAD	64.15	64.98	**66.27**	**66.95**	**67.21**	**67.46**	**67.82**	**67.98**	**68.17**	**79.68**

Table 2: Classification accuracy against ℓ_∞ attacks at each stage on ImageNet-100 in the white-box setting and the gray-box setting.

	Method	0:PGD	1:SNIM	2:VNIM	3:CW	4:DeepF.	5:EAD	6:EADEN	7:OnePixel	8:SparseF.
CL	EWC [Kirkpatrick *et al.*, 2017]	**95.45**	68.20	67.01	23.74	23.63	23.68	26.90	25.92	22.21
	SSRE [Zhu *et al.*, 2022]	89.13	82.47	78.14	65.82	58.41	50.63	47.52	42.44	37.78
FCL	FACT [Zhou *et al.*, 2022]	94.52	84.23	86.41	81.54	80.88	78.62	78.43	78.17	74.62
	OrCo [Ahmed *et al.*, 2024]	94.63	87.11	86.50	81.17	81.51	80.23	79.29	78.91	75.22
UCAD(ours)		94.52	**89.11**	**90.54**	**84.13**	**84.78**	**81.10**	**82.13**	**82.83**	**78.24**
UCAD†(ours)		94.52	95.53	95.12	95.22	94.94	85.28	94.63	95.97	95.23

Table 3: Average incremental accuracy (AIAcc) against $\ell_{p=0,1,2,\infty}$ attacks at each stage on CIFAR-10 in the gray-box setting. DeepF. and SparseF. denote DeepFool and SparseFool respectively. **UCAD†** denotes Premium-UCAD, serving as the oracle upper bound.

their corresponding UCAD variants, which use these models as defense models under the CAD setting. The robust models include TRADES [Zhang *et al.*, 2019], GAIRAT [Zhang *et al.*, 2020], and DMAT [Wang *et al.*, 2023b].To assess UCAD's ability to mitigate catastrophic forgetting, we compare it with CL methods EWC [Kirkpatrick *et al.*, 2017] and SSRE [Zhu *et al.*, 2022], as well as few-shot continual learning (FCL) methods FACT [Zhou *et al.*, 2022] and OrCo [Ahmed *et al.*, 2024], in the gray-box setting. Additionally, we propose Premium-UCAD, a variant of UCAD without memory or defense budget constraints that serves as an oracle upper bound, where attack-specific defense models are trained with an abundant defense budget and ensembled with the target model via a weight estimator.

5.2 Comparisons to Baselines

We compare UCAD with other defense methods in both white-box and gray-box setting. As shown in Table 1 and Table 2, in the white-box setting, UCAD achieves average accuracies of 50.52 on CIFAR-10 and 27.79 on ImageNet-100 across nine attacks, outperforming vanilla AT (34.71 and 19.67) and SSEAT (45.91 and 24.54). For clean images, UCAD achieves accuracies of 94.32 and 86.47 on CIFAR-10 and ImageNet-100, respectively, substantially exceeding both baselines. These results highlight UCAD's ability to continually adapt to newly emerging threats and enable robust models to perform effective defense under the CAD setting.

Compared with robust models, UCAD employs a model ensemble, which leads to slightly lower robust accuracy on the initial attack. However, after adapting to multiple attacks, UCAD improves classification performance on clean images by 1.84–5.61 percentage points on CIFAR-10 and by 2.92–5.22 percentage points on ImageNet-100. Moreover, it further enhances the defense performance of robust models by 1.96–2.24 percentage points on CIFAR-10 and by 1.55–1.66 percentage points on ImageNet-100.

UCAD performs even better in the gray-box setting. On CIFAR-10 and ImageNet-100, it achieves average adversarial accuracies of 93.26 and 82.58, substantially outperforming SSEAT (74.34 and 70.90, respectively) and surpassing the best robust models by 7.83 and 17.37 percentage points. In addition, UCAD maintains clean accuracies of 95.82 and 88.13 on CIFAR-10 and ImageNet-100, respectively, close to the non-defense model (96.42 and 89.76). These results confirm the superiority of UCAD in the gray-box setting.

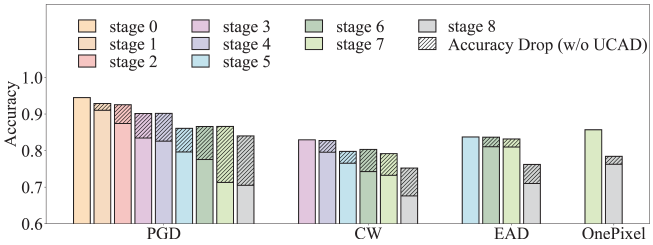

Figure 3: Verification of catastrophic forgetting in UCAD and a vanilla AT on CIFAR-10 under the attack sequence from Table 3.

Ablation	0:PGD	1:SNIM	2:CW	3:EAD	4:OnePixel	Clean
w/o $\mathcal{L}_0$	94.52	80.43	82.44	76.10	71.82	95.82
w/o $\mathcal{L}_{\text{proto}}$	94.43	88.65	89.52	80.04	78.49	95.82
w/o f_p	94.52	89.11	89.93	86.64	85.31	55.87
UCAD	**94.52**	**89.11**	**89.93**	**86.64**	**85.31**	**95.82**

Table 4: AIAcc against adversarial attacks and accuracy on clean images for ablation study. "w/o $\mathcal{L}_0$", "w/o $\mathcal{L}_{\text{proto}}$", and "w/o f_p" denote UCAD without embedding space reservation, prototype augmentation, and model ensemble, respectively.

5.3 Analyses and Ablation Study

Catastrophic Forgetting

To evaluate UCAD's ability to mitigate catastrophic forgetting, we compare it with CL and FCL methods on CIFAR-10. As shown in Table 3, UCAD maintains robust defense performance despite varying attack types. However, as the diversity of attacks increases, its performance gradually declines from 94.52 to 78.24. This suggests that defending against an expanding array of attack types remains a significant challenge for continual defense methods. Compared to the top baselines, UCAD achieves an average AIAcc that is $3.30x$ higher than FACT and 2.42 higher than OrCo across all stages. Furthermore, we compare UCAD with a vanilla model that serves as an empirical lower bound. As shown in Figure 3, UCAD significantly reduces forgetting–without it, error rates on PGD and CW exceed 50% by stage 8.

Ablation Study

As argued, embedding reservation facilitates smoother adaptation and reduces overfitting, prototype augmentation preserves decision boundaries from previous stages, and model ensembling ensures high classification accuracy for both clean and adversarial images. To validate the contribution of each component, we conduct an ablation study using four attacks: PGD (ℓ_∞), CW (ℓ_2), EAD (ℓ_1), and OnePixel (ℓ_0). As shown in Table 4, removing embedding space reservation ($\mathcal{L}_0$ in Eq. 4) leads to a significant drop in AIAcc at each stage due to the absence of a mechanism tailored for few-shot scenarios. In stage 4, AIAcc drops to 71.82, which is 13.49 lower than the original setting. When prototype augmentation ($\mathcal{L}_{\text{proto}}$ in Eq. 6) is removed, UCAD loses its ability to prevent catastrophic forgetting, resulting in a noticeable decrease in AIAcc after stage 1, reaching 78.49 in stage 4, which is 6.82 lower than the original. Additionally, removing model ensembling (f_p in Eq. 9) severely impairs UCAD's classification performance on clean images, reducing accuracy by 39.95 compared to the original. These results confirm the effectiveness of each component.

Attack (unseen)	0:PGD	1:SNIM	2:BIM	3:RFGSM	4:MIM
DIM	72.83	73.89	75.42	76.65	77.53
NIM	72.50	73.62	75.32	76.78	77.31
VNIM	70.81	72.44	74.52	75.21	76.44
VMIM	71.61	72.56	74.88	75.10	76.92

Table 5: Robust accuracy of UCAD against unseen attacks after each adaptation.

Attack (unseen)	0	10	20	30	40
DIM	72.83	80.52	87.04	89.25	89.01
NIM	72.50	80.75	86.33	89.21	89.44
VNIM	70.81	79.63	85.51	87.92	88.37
VMIM	71.61	79.83	85.38	88.25	88.22

Table 6: Robust accuracy of UCAD against unseen attacks after each adaptation with a simulated attack sequence.

Defense Against Unseen Attack and Saturation

As shown in Table 5, adaptation to known attacks enables UCAD to generalize to unseen ones. After four stages, UCAD learns diverse perturbation patterns, improving accuracy on unseen attacks from 71.94 to 77.05. Eventually, when UCAD has adapted to a sufficiently diverse set of attacks, its defense performance may reach a saturation point. However, due to the current limited availability of attack types, this saturation point cannot yet be empirically determined. Therefore, we construct simulated attacks using perturbation forgery [Wang *et al.*, 2024a] and evaluate UCAD's generalization ability after adapting to these synthetic attacks. As shown in Table 6, after 20 adaptation stages, UCAD achieved an average classification accuracy of over 86.07 on unseen attacks. After 30 stages, the average accuracy on unseen attacks reached 88.66 and saturated—indicating that further adaptation did not yield additional improvements. This surpasses all baseline methods. As mentioned earlier, UCAD requires pre-allocated virtual classes for future classifier expansion. The observed saturation behavior suggests that, with a sufficient number of pre-allocated classes, UCAD can reach a stage at which it defends against unseen attacks without further adaptation, thereby alleviating concerns about its scalability.

6 Conclusion

We formulate four fundamental principles for CAD: (1) *continual adaptation to new attacks without catastrophic forgetting*, (2) *few-shot adaptation*, (3) *memory-efficient adaptation*, and (4) *high classification accuracy on both clean and adversarial data*, to address the limitations of existing CAD methods. Guided by these principles, we propose **Universal Continual Adversarial Defense** (**UCAD**), a universal framework that enables both standard and robust models to defend effectively under the CAD setting. Extensive experiments against multi-stage adversarial attacks show that UCAD significantly outperforms diverse baselines. Moreover, as more attacks are encountered, UCAD progressively strengthens existing robust models until performance saturation.

Acknowledgments

This work was supported in part by the Natural Science Foundation of China under Grant 62372203 and 62302186, in part by the Major Scientific and Technological Project of Shenzhen (202316021), in part by the National key research and development program of China(2022YFB2601802), in part by the Major Scientific and Technological Project of Hubei Province (2022BAA046, 2022BAA042).

References

[Ahmed *et al.*, 2024] Noor Ahmed, Anna Kukleva, and Bernt Schiele. Orco: Towards better generalization via orthogonality and contrast for few-shot class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28762–28771, 2024.

[Carlini and Wagner, 2017] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. Ieee, 2017.

[Chen *et al.*, 2018] Pin-Yu Chen, Yash Sharma, Huan Zhang, Jinfeng Yi, and Cho-Jui Hsieh. Ead: elastic-net attacks to deep neural networks via adversarial examples. In *AAAI*, volume 32, 2018.

[Chen *et al.*, 2021] Zhuotong Chen, Qianxiao Li, and Zheng Zhang. Towards robust neural networks via close-loop control. In *International Conference on Learning Representations*, 2021.

[Cho *et al.*, 2025] Seungju Cho, Hongsin Lee, and Changick Kim. Enhancing robustness in incremental learning with adversarial training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 2518–2526, 2025.

[Croce *et al.*, 2022] Francesco Croce, Sven Gowal, Thomas Brunner, Evan Shelhamer, Matthias Hein, and Taylan Cemgil. Evaluating the adversarial robustness of adaptive test-time defenses. In *International Conference on Machine Learning*, pages 4421–4435. PMLR, 2022.

[Dong *et al.*, 2018] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *CVPR*, pages 9185–9193, 2018.

[Grathwohl *et al.*, 2020] Will Grathwohl, Kuan-Chieh Wang, Jörn-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. Your classifier is secretly an energy based model and you should treat it like one. In *ICLR*, 2020.

[Hill *et al.*, 2020] Mitch Hill, Jonathan Craig Mitchell, and Song-Chun Zhu. Stochastic security: Adversarial defense using long-run dynamics of energy-based models. In *ICLR*, 2020.

[Jiang *et al.*, 2023] Yulun Jiang, Chen Liu, Zhichao Huang, Mathieu Salzmann, and Sabine Susstrunk. Towards stable and efficient adversarial training against l_1 bounded adversarial attacks. In *ICML*, pages 15089–15104. PMLR, 2023.

[Kang *et al.*, 2021] Qiyu Kang, Yang Song, Qinxu Ding, and Wee Peng Tay. Stable neural ode with lyapunov-stable equilibrium points for defending against adversarial attacks. *Advances in Neural Information Processing Systems*, 34:14925–14937, 2021.

[Kirkpatrick *et al.*, 2017] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

[Kurakin *et al.*, 2016] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale, 2016.

[Laidlaw *et al.*, 2020] Cassidy Laidlaw, Sahil Singla, and Soheil Feizi. Perceptual adversarial robustness: Defense against unseen threat models. In *ICLR*, 2020.

[Li *et al.*, 2022] Xiyuan Li, Zou Xin, and Weiwei Liu. Defending against adversarial attacks via neural dynamic system. *Advances in Neural Information Processing Systems*, 35:6372–6383, 2022.

[Li *et al.*, 2023] Yushu Li, Xun Xu, Yongyi Su, and Kui Jia. On the robustness of open-world test-time training: Self-training with dynamic prototype expansion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11836–11846, 2023.

[Lin *et al.*, 2019] Jiadong Lin, Chuanbiao Song, Kun He, Liwei Wang, and John E Hopcroft. Nesterov accelerated gradient and scale invariance for adversarial attacks. In *ICLR*, 2019.

[Liu *et al.*, 2022] Yaoyao Liu, Yingying Li, Bernt Schiele, and Qianru Sun. Online hyperparameter optimization for class-incremental learning. In *AAAI*, 2022.

[Madry *et al.*, 2017] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks, 2017.

[Modas *et al.*, 2019] Apostolos Modas, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. Sparsefool: a few pixels make a big difference. In *CVPR*, pages 9087–9096, 2019.

[Moosavi-Dezfooli *et al.*, 2016] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016.

[Nie *et al.*, 2022] Weili Nie, Brandon Guo, Yujia Huang, Chaowei Xiao, Arash Vahdat, and Anima Anandkumar. Diffusion models for adversarial purification. In *ICML*, 2022.

[Pei *et al.*, 2025] Gaozheng Pei, Shaojie Lyu, Gong Chen, Ke Ma, Qianqian Xu, Yingfei Sun, and Qingming Huang. Divide and conquer: Heterogeneous noise integration for

diffusion-based adversarial purification. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 29268–29277, 2025.

[Rebuffi *et al.*, 2017] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *CVPR*, pages 2001–2010, 2017.

[Snell *et al.*, 2017] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *NeurIPS*, 30, 2017.

[Su *et al.*, 2019] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 23(5):828–841, 2019.

[Taran *et al.*, 2019] Olga Taran, Shideh Rezaeifar, Taras Holotyak, and Slava Voloshynovskiy. Defending against adversarial attacks by randomized diversification. In *CVPR*, pages 11226–11233, 2019.

[Tian *et al.*, 2020] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pages 776–794. Springer, 2020.

[Tramèr *et al.*, 2018] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. In *ICLR*, 2018.

[Verma *et al.*, 2019] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *ICML*, pages 6438–6447, 2019.

[Wang and He, 2021] Xiaosen Wang and Kun He. Enhancing the transferability of adversarial attacks through variance tuning. In *CVPR*, pages 1924–1933, 2021.

[Wang *et al.*, 2023a] Qian Wang, Yongqin Xian, Hefei Ling, Jinyuan Zhang, Xiaorui Lin, Ping Li, Jiazhong Chen, and Ning Yu. Detecting adversarial faces using only real face self-perturbations. In *IJCAI*, pages 1488–1496, 8 2023. Main Track.

[Wang *et al.*, 2023b] Zekai Wang, Tianyu Pang, Chao Du, Min Lin, Weiwei Liu, and Shuicheng Yan. Better diffusion models further improve adversarial training. In *ICML*, 2023.

[Wang *et al.*, 2024a] Qian Wang, Chen Li, Yuchen Luo, Hefei Ling, Shijuan Huang, Ruoxi Jia, and Ning Yu. Detecting adversarial data using perturbation forgery. *arXiv preprint arXiv:2405.16226*, 2024.

[Wang *et al.*, 2024b] Wenxuan Wang, Chenglei Wang, Huihui Qi, Menghao Ye, Xuelin Qian, Peng Wang, and Yanning Zhang. Sustainable self-evolution adversarial training. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 9799–9808, 2024.

[Wang *et al.*, 2025] Wenxuan Wang, Chenglei Wang, and Xuelin Qian. Dynamic dual-level defense routing for continual adversarial training. *arXiv preprint arXiv:2509.21392*, 2025.

[Wu *et al.*, 2021] Weibin Wu, Yuxin Su, Michael R. Lyu, and Irwin King. Improving the transferability of adversarial samples with adversarial transformations. In *CVPR*, pages 9020–9029, 2021.

[Yin *et al.*, 2024] Shenglin Yin, Kelu Yao, Zhen Xiao, and Jieyi Long. Embracing adaptation: An effective dynamic defense strategy against adversarial examples. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 9435–9444, 2024.

[Yoon *et al.*, 2021] Jongmin Yoon, Sung Ju Hwang, and Juho Lee. Adversarial purification with score-based generative models. In *ICML*, pages 12062–12072, 2021.

[Zhang and Sabuncu, 2018] Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *NeurIPS*, volume 31, 2018.

[Zhang *et al.*, 2019] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *ICML*, pages 7472–7482, 2019.

[Zhang *et al.*, 2020] Jingfeng Zhang, Jianing Zhu, Gang Niu, Bo Han, Masashi Sugiyama, and Mohan Kankanhalli. Geometry-aware instance-reweighted adversarial training. In *ICLR*, 2020.

[Zhang *et al.*, 2021] Chi Zhang, Nan Song, Guosheng Lin, Yun Zheng, Pan Pan, and Yinghui Xu. Few-shot incremental learning with continually evolved classifiers. In *CVPR*, pages 12455–12464, 2021.

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

[Zhou *et al.*, 2022] Da-Wei Zhou, Fu-Yun Wang, Han-Jia Ye, Liang Ma, Shiliang Pu, and De-Chuan Zhan. Forward compatible few-shot class-incremental learning. In *CVPR*, pages 9046–9056, 2022.

[Zhu *et al.*, 2021a] Fei Zhu, Zhen Cheng, Xu-yao Zhang, and Cheng-lin Liu. Class-incremental learning via dual augmentation. In *NeurIPS*, volume 34, pages 14306–14318, 2021.

[Zhu *et al.*, 2021b] Fei Zhu, Xu-Yao Zhang, Chuang Wang, Fei Yin, and Cheng-Lin Liu. Prototype augmentation and self-supervision for incremental learning. In *CVPR*, pages 5871–5880, 2021.

[Zhu *et al.*, 2022] Kai Zhu, Wei Zhai, Yang Cao, Jiebo Luo, and Zheng-Jun Zha. Self-sustaining representation expansion for non-exemplar class-incremental learning. In *CVPR*, pages 9296–9305, 2022.