Buffer-Guided Differentiated Learning for Adversarial Fine-Tuning in Self-Supervised Pretrained Models

Jiahui Kong[1], Siya Yao[1*], Xiaofeng Zhou[1], and Xiaoyu Sean Lu[2]
[1] College of Information and Electronic Engineering, Zhejiang Gongshang University, Hangzhou, China
[2] School of Cyber Science and Engineering, Nanjing University of Science and Technology, Nanjing, China
Emails: 24020090093@pop.zjgsu.edu.cn, yaosiya@mail.zjgsu.edu.cn, 2421314810@qq.com, xiaoyu.lu@njust.edu.cn

***Abstract*—Self-supervised pretrained models are fundamental to modern visual recognition systems. However, their open-source nature makes them vulnerable to adversarial attacks, and their widespread reuse across diverse downstream tasks amplifies the impact of such attacks, particularly under universal adversarial perturbations (UAPs). Adversarial fine-tuning enhances robustness by integrating adversarial training into the downstream adaptation process and leverages the model's redundant capacity to restore standard accuracy without compromising robustness. Existing methods typically freeze most network layers and fine-tune only a few isolated ones. However, this approach overlooks inter-layer dependencies and gradient propagation, resulting in a loss of coherence in the effective pretrained representations. To address this, we propose Buffer-guided Differentiated Learning for Adversarial Fine-tuning (BDAF), which enables more fine-grained optimization to better balance robustness and accuracy. BDAF partitions network layers into robustness-sensitive layers, robustness-redundant layers, and buffer layers according to their contributions to robustness, and applies differentiated learning rates based on both functional roles and depth positions. We adopt a robustness–accuracy trade-off score to guide model selection. Extensive experiments on 7 self-supervised pretraining methods and 4 datasets, evaluated using 5 UAPs, demonstrate that BDAF consistently outperforms existing methods in both standard and robust accuracy, with strong cross-task generalization. Code is available at https://github.com/Xisi-7/BDAF.**

***Index Terms*—Pretrained model robustness, Adversarial training, Fine-tuning, Universal adversarial perturbation, Self-supervised learning**

I. Introduction

Self-supervised learning (SSL) has emerged as an effective paradigm for pretraining visual encoders, enabling the extraction of transferable representations from large-scale unlabeled data. These pretrained encoders facilitate efficient adaptation to diverse downstream tasks through lightweight fine-tuning. Several representative SSL models, such as the Simple Framework for Contrastive Learning of Visual Representations (SimCLR) [1] and Momentum Contrast (MoCo) [2], have been widely adopted and have established a mature open-source ecosystem. Despite these successes, vulnerabilities acquired during pretraining are often transferred to downstream models, as they typically inherit the original architecture and most parameters of the source encoder [3]. Due to the limited size of downstream datasets, fine-tuning often fails to correct the fragile decision boundaries inherited, making models highly sensitive to small perturbations. Furthermore, high-performance pretrained models are usually developed by a small number of organizations and publicly released, which facilitates adversarial analysis and increases the exposure of visual encoders to real-world attacks.

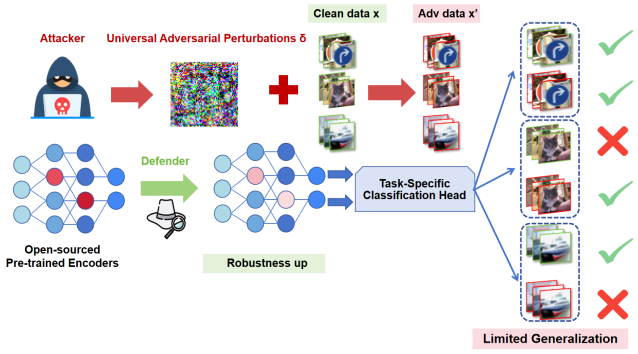

Fig. 1: Even when protected by defense mechanisms, pretrained models struggle to remain effective against stronger universal adversarial attacks across diverse downstream tasks.

Existing defense strategies for pretrained models are generally categorized into three types: parameter pruning [4] [5], model distillation [6] [7], and adversarial training (AT) [8]–[14]. Adversarial fine-tuning (AFT) [15]–[18] extends AT to pretrained models by incorporating adversarial examples during adaptation, thereby enhancing robustness while preserving efficiency and generalization. However, this balance becomes increasingly fragile due to the growing strength and transferability of universal adversarial perturbations (UAPs) [19]. Recent studies demonstrate that perturbations generated by amplifying low-level features [20] or manipulating high-frequency components [21] can transfer effectively across pretrained encoders and downstream tasks. As illustrated in Fig. 1, even defended pretrained models often struggle against stronger UAPs across diverse downstream tasks.

*Corresponding author. This work was supported by the Youth Project of the National Natural Science Foundation of China under Grant 62406282 and 62302223, Zhejiang Provincial Natural Science Foundation of China under Grant LMS26F030018, and Fundamental Research Funds for the Provincial Universities of Zhejiang under Grant 3090JYN9924001G-016.

We revisit the roles of network layers in shaping robustness during AFT and observe that most approaches fine-tune only a small subset of layers deemed redundant, while neglecting their depth-wise positions and interactions with neighboring layers. Given the heterogeneity of convolutional layers in terms of depth, gradient propagation, and sensitivity to adversarial perturbations, this selective fine-tuning disrupts cross-layer gradient flow and weakens feature collaboration.

Therefore, we propose an efficient framework called Buffer-Guided Differentiated Learning for Adversarial Fine-Tuning. BDAF explicitly models the heterogeneity of network layers along two dimensions—functional roles and network positions—during adversarial fine-tuning, effectively balancing robustness enhancement with the preservation of pretrained representations. Our contributions are summarized as follows:

- We propose a buffer-guided robustness-aware layer role assignment strategy and a differentiated learning rate fine-tuning scheme that promote smoother local gradient propagation and improved downstream generalization.
- We propose a model selection strategy based on a robustness–accuracy trade-off score, enabling principled model selection under robustness constraints.
- Extensive experiments on seven SSL methods across four datasets demonstrate that BDAF achieves strong robustness and high accuracy against state-of-the-art UAPs.

II. RELATED WORK

Adversarial training (AT) is one of the most effective defenses against adversarial attacks. Goodfellow et al. [8] first enhanced robustness by augmenting the training data with adversarial examples to improve model performance. Later, Madry et al. [9] formulated adversarial training as a robust optimization problem and introduced Projected Gradient Descent (PGD) to generate near worst-case adversarial examples, significantly boosting model robustness. Standard adversarial training often compromises natural accuracy. To address this, tradeoff-inspired adversarial defenses have been proposed [10] using surrogate-loss minimization. This approach introduces novel loss functions that combine the classification loss on clean samples with a penalty term that measures the discrepancy between the predictive distributions of clean and adversarial inputs. This method achieves a better balance between classification accuracy and adversarial robustness. Furthermore, Wang et al. [11] identified the limiting effect of naturally misclassified samples on adversarial robustness and further enhanced robustness by treating correctly classified and misclassified samples differently.

For pretrained models, recent studies exploit model redundancy during adversarial fine-tuning. Robustness-critical fine-tuning [16] exploits module-level criticality to selectively fine-tune adversarially trained models, leveraging redundant capacity to improve both generalization and robustness. GenAF [17] further advances generalization by adaptively optimizing redundant layers through an evolution-based fine-tuning strategy. Although these approaches leverage redundancy to enhance performance, they still demonstrate limited adversarial robustness on certain downstream tasks. To address this limitation, criticality-leveraged adversarial training [18] introduces a criticality index to identify layers that are particularly vulnerable to adversarial attacks with relatively low computational overhead. It then applies targeted optimization through a dynamic layer selection process, thereby improving model robustness.

Overall, research on adversarial training has progressed from solely enhancing robustness to developing optimization strategies that balance robustness with natural accuracy. Unlike previous methods that optimize models from a holistic or single-layer perspective and overlook the heterogeneity in representational capacity and vulnerability across individual layers, our approach accounts for inter-layer collaboration, representation capabilities, and functional roles in adversarial robustness, enabling more fine-grained adversarial fine-tuning.

III. METHODOLOGY

In this section, we present the overall framework of BDAF, which consists of three stages: adversarial fine-tuning, standard fine-tuning, and model selection. We start from publicly available self-supervised pretrained encoders and apply adversarial training with PGD-generated samples to obtain robust parameters $\boldsymbol{\theta}_{AT}$. This stage equips the model with resilience against adversarial perturbations, establishing a robust foundation for subsequent fine-grained optimization.

In the second stage, BDAF incorporates a role-aware, differentiated optimization mechanism during standard fine-tuning, as illustrated in Fig. 2. This approach aims to enhance adversarial robustness while preserving the representational capacity of the pretrained model and ensuring stable cross-layer feature propagation. The objective is defined as follows:

$$\mathcal{L}_{SFT} = \min_{\boldsymbol{\theta}_{AT}} \mathbb{E}_{(x,y)\in D} \mathcal{L}_{CE}(x, y; \boldsymbol{\theta}_{AT}) \tag{1}$$

where $\mathcal{L}_{CE}(\cdot)$ denotes the cross-entropy loss, and D represents the downstream task data distribution.

In the third stage, we introduce the robustness–accuracy trade-off score S_{TO} to select target models that achieve both strong robustness and generalization. Next, we provide a detailed description of the specific implementations of BDAF.

A. Buffer-guided Robustness-Aware Layer Role Assignment Strategy (BRAL)

To evaluate the impact of individual layers on robustness, we estimate each layer's robustness contribution in an adversarially trained model by perturbing one layer at a time and maximizing the model loss $\mathcal{L}$ to assess the worst-case performance degradation caused by that perturbation. The difference between the resulting worst-case loss and the original model loss defines the robustness contribution of the i-th layer, denoted as RC_i. Formally, this process can be expressed as follows:

$$RC_i = \max_{\|\Delta\boldsymbol{\theta}^{(i)}\|\leq\delta} \mathcal{L}\big(f(x, y; \boldsymbol{\theta} + \Delta\boldsymbol{\theta})\big) - \mathcal{L}\big(f(x, y; \boldsymbol{\theta})\big) \tag{2}$$

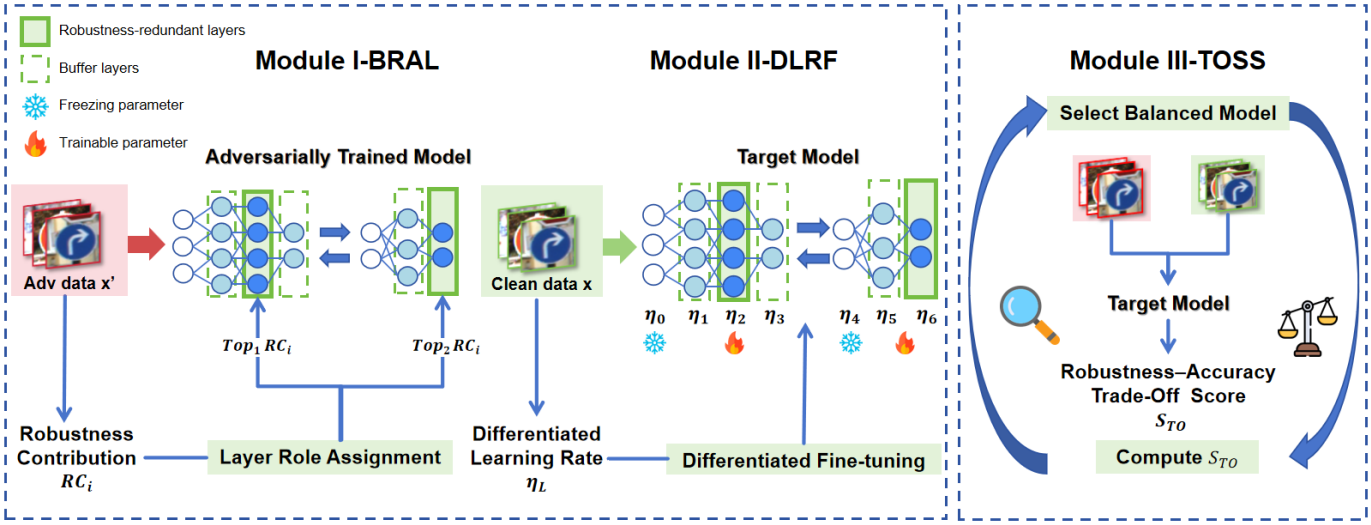

Fig. 2: The overall framework of BDAF consisting of Module I-Buffer-guided Robustness-Aware Layer Role Assignment Strategy (BRAL), Module II-Differential Learning Rate Fine-Tuning, and Module III-Robustness–Accuracy Trade-Off Score-based Model Selection (TOSS).

Where $f_i(\cdot)$ denotes the feature extraction and decision-making behavior of each layer, $\boldsymbol{\theta}$ represents the original model parameters, and the perturbation $\Delta\boldsymbol{\theta}$ is constrained within a radius δ, as denoted below:

$$\Delta\boldsymbol{\theta} = \{0, \ldots, 0, \Delta\boldsymbol{\theta}^{(i)}, 0, \ldots, 0\} \tag{3}$$

Given that CNNs exhibit strong inter-layer dependencies, feature extraction and decision-making processes are collaboratively formed across layers, with the output of each layer serving as the input to the next. Consider an n-layer DNN M with input X, whose output is denoted as $M_n(X)$, as expressed in (4):

$$M_n(X) = f_n(f_{n-1}(\ldots f_1(X))) \tag{4}$$

It is evident that if only a single layer f_{n-1} is freely fine-tuned while its adjacent layers f_n and f_{n-2} are kept frozen, the feature propagation chain may be disrupted. Motivated by this, we propose a Buffer-Guided Robustness-Aware Layer Role Assignment strategy (BRAL), which explicitly assigns functional roles to network layers based on their contributions to robustness. The bottom 20% are designated as robustness-redundant layers L_{red} and are the main fine-tuning targets because they contribute less to robustness. Layers adjacent to L_{red} are defined as buffer layers L_{buf} to smooth feature transitions and reduce drift during updates. All remaining layers are considered robustness-sensitive layers L_{sen}, which are critical for maintaining adversarial robustness and preserving feature representations. By explicitly modeling layer-wise functional roles and introducing buffer layers to maintain cross-layer consistency, BRAL provides a structured foundation for subsequent differentiated fine-tuning.

B. Differentiated Learning Rate Fine-tuning (DLRF)

Based on the layer roles assigned by BRAL, we adopt role-aware learning rate (RALR) strategies across network layers. For L_{red}, a relatively small learning rate is used to enable fine-grained parameter updates. For L_{buf}, an even smaller learning rate is applied, as these layers primarily serve to buffer gradient fluctuations and facilitate smoother feature updates. For L_{sen}, parameters are kept frozen to preserve core feature representations and maintain the model's baseline adversarial defense capability.

Although RALR provides coarse-grained control over parameter updates, it does not account for the inherent hierarchical nature of feature learning. In convolutional architectures, lower layers typically capture general and transferable features, while higher layers encode more task-specific representations. To better align the fine-tuning process with this hierarchical structure, we introduce a layer-wise learning rate decay (LLRD) mechanism alongside RALR. By integrating RALR with LLRD, DLRF enables fine-grained control over parameter updates, as formally defined in (5).

$$\eta_L = \begin{cases} \eta_r \lambda^{N-1-L}, & L \in L_{red} \\ \eta_b \lambda^{N-1-L}, & L \in L_{buf} \\ 0, & L \in L_{sen} \end{cases} \tag{5}$$

Here, η_L denotes the learning rate of the L-th layer. η_r and η_b are the base learning rates for L_{red} and L_{buf}, respectively, with $\eta_b < \eta_r$, while the learning rate for L_{sen} is set to zero. The decay factor $\lambda \in (0, 1)$ controls the strength of layer-wise decay. N denotes the total number of layers, and L is the layer index, increasing from lower to higher layers. As L decreases, the exponent $N - 1 - L$ increases, leading to a stronger decay effect and thus smaller effective learning rates for lower layers. Conversely, higher layers receive relatively

TABLE I: The Robust Testing Accuracy RA (%) results

Attack Method	Dataset	BYOL		DINO		MoCov2+		NNCLR		RESSL		SwAV		W-MSE	
		Gen-AF [17]	BDAF	Gen-AF	BDAF	Gen-AF	BDAF	Gen-AF	BDAF	Gen-AF	BDAF	Gen-AF	BDAF	Gen-AF	BDAF
UAP [22]	ANIMALS10	91.80	81.51	70.76	88.08	95.07	88.77	87.70	95.51	93.87	89.01	90.27	87.89	81.13	85.34
	CIFAR10	87.22	84.22	76.37	82.22	87.80	88.09	78.45	80.76	81.95	86.05	80.12	81.93	71.10	80.89
	GTSRB	99.90	99.98	99.73	99.98	98.09	83.20	99.86	99.99	99.86	99.98	99.83	99.99	98.85	99.95
	STL10	60.67	72.68	67.09	78.48	68.99	79.41	80.05	81.35	76.55	78.03	72.20	83.63	75.31	80.98
	AVG	**84.90**	84.60	78.49	**87.19**	**87.49**	84.87	86.52	**89.40**	88.06	**88.27**	85.61	**88.36**	81.60	**86.79**
UAP-EPGD [23]	ANIMALS10	91.89	78.56	70.79	79.88	95.31	88.08	88.08	80.80	93.86	86.72	90.90	85.20	85.77	81.32
	CIFAR10	86.90	82.23	73.08	80.19	88.24	86.98	78.47	78.16	82.12	84.90	80.53	80.83	74.48	78.48
	GTSRB	99.72	96.00	99.54	94.97	98.60	98.84	99.73	93.89	99.76	95.04	99.82	98.33	98.78	95.94
	STL10	60.46	70.34	66.72	70.94	68.71	78.28	80.19	78.08	76.99	74.59	77.61	79.35	79.04	75.34
	AVG	**84.74**	81.78	77.53	**81.50**	87.72	**88.05**	**86.62**	82.73	**88.18**	85.31	**87.22**	85.93	**84.52**	82.77
SSP [24]	ANIMALS10	92.11	79.83	41.85	74.64	94.93	88.18	50.78	70.71	94.09	78.04	86.15	83.87	70.12	75.72
	CIFAR10	87.34	81.98	62.44	79.90	87.48	86.25	78.55	77.11	82.07	84.18	62.29	80.01	26.73	76.97
	GTSRB	99.88	98.64	99.17	98.31	98.36	98.98	99.13	97.89	99.89	96.17	99.70	98.31	96.68	94.32
	STL10	60.66	70.52	39.18	67.16	68.62	78.22	74.43	78.41	70.58	75.76	37.28	72.06	67.57	74.18
	AVG	**85.00**	82.74	60.66	**80.00**	87.35	**87.91**	75.72	**81.03**	**86.66**	83.54	71.36	**83.56**	65.28	**80.30**
PAP [20]	ANIMALS10	92.12	79.75	34.39	79.77	95.06	87.75	44.36	66.14	94.07	86.47	82.48	82.58	55.42	74.47
	CIFAR10	87.65	82.66	75.05	80.97	89.30	87.09	78.82	78.61	82.16	85.13	58.33	80.69	54.70	79.69
	GTSRB	99.93	99.80	99.73	98.03	99.54	99.99	99.76	99.78	99.88	99.88	99.53	99.10	96.86	99.18
	STL10	60.45	71.32	31.34	67.94	68.69	78.47	71.97	78.24	72.68	76.07	58.46	72.33	63.02	68.55
	AVG	**85.04**	83.38	60.13	**81.68**	88.15	**88.33**	73.73	**80.69**	**87.20**	86.89	74.70	**83.68**	67.50	**80.47**
AdvEncoder [21]	ANIMALS10	91.48	80.42	65.68	75.78	94.45	87.79	83.14	82.59	91.43	83.03	89.91	82.42	71.48	67.64
	CIFAR10	86.53	82.38	34.71	79.32	64.15	84.58	75.62	78.21	78.09	83.61	78.88	78.77	78.09	75.83
	GTSRB	68.12	81.18	98.79	83.91	94.59	98.73	96.96	92.01	98.81	90.80	99.71	94.79	98.48	90.13
	STL10	60.47	69.28	58.13	64.06	66.61	77.32	73.40	73.78	63.57	69.80	63.45	70.10	70.89	70.37
	AVG	76.65	**78.32**	64.33	**75.77**	79.95	**87.11**	**82.28**	81.65	**82.98**	81.81	**82.99**	81.52	**79.74**	75.99
Overall AVG		**83.27**	82.16	68.23	**81.23**	86.13	**87.25**	80.97	**83.10**	**86.61**	85.16	80.37	**84.61**	75.73	**81.26**

larger learning rates, enabling greater adaptability to task-specific features.

C. Robustness–Accuracy Trade-Off Score-based Model Selection (TOSS)

To select checkpoints with a better balance between standard accuracy and robustness, we introduce a unified evaluation metric termed the Robustness–Accuracy Trade-Off Score S_{TO}, to guide model selection. Specifically, S_{TO} is defined as the difference between the model's test accuracy TA on clean samples and its accuracy RA under adversarial perturbations:

$$S_{TO} = TA - RA \tag{6}$$

This score quantifies the extent of performance degradation under adversarial attacks, thereby characterizing the trade-off between accuracy and robustness. A smaller S_{TO} indicates that the model maintains more consistent performance across both clean and adversarial settings while retaining competitive clean accuracy.

In addition, we retain checkpoints that simultaneously improve both TA and RA, even if their S_{TO} does not decrease, thereby ensuring that the selected models do not sacrifice one objective for the other.

IV. EXPERIMENTS

Our experimental setup follows prior work on AFT. We consider a threat model in which the attacker has direct access to the source model and can generate adversarial examples to attack the target model, without knowledge of the pre-training dataset or the downstream task. From the defender's perspective, our goal is to enhance robustness against both full upstream-knowledge attackers, who have complete access to the pretrained source model, and partial upstream-knowledge attackers, who possess limited information about the model.

A. Experimental Setup

Models and Datasets. We utilize pretrained encoders from the solo-learn framework [25], which provides high-quality implementations of state-of-the-art SSL methods. We select seven representative SSL approaches: BYOL, DINO, MoCov2+, NNCLR, RESSL, SwAV, and W-MSE. All encoders are based on the ResNet-18 backbone and pretrained on CIFAR10 [26], and are treated as victim models to evaluate BDAF. To comprehensively assess performance across different data distributions and task scenarios, we conduct experiments on four standard image classification datasets: CIFAR10, STL10 [27], GTSRB [28], and ANIMALS10 [29].We evaluate the robustness of the proposed method using five widely used UAPs, including UAP [22], UAP-EPGD [23], SSP [24], PAP [20], and AdvEncoder [21].

Benchmarks. To evaluate the performance of our proposed BDAF method, we compare it with the current state-of-the-art approach Gen-AF [17], which is the first method designed to protect pretrained encoders against UAPs.

Evaluation Metrics. To evaluate the effectiveness of the proposed method, we employ two metrics, i.e., TA and RA, to assess the model's accuracy and robustness.

Implementation Details. We set the upper limit of perturbation δ to 10/255. The L_{red} are fine-tuned with a learning

TABLE II: The Testing Accuracy TA (%) results

Fine-tuning Method	Dataset	BYOL	DINO	MoCov2+	NNCLR	RESSL	SwAV	W-MSE
Standard Fine-tuning	ANIMALS10	83.22	80.21	81.90	81.32	79.93	78.40	73.60
	CIFAR10	94.46	91.41	94.97	93.84	92.74	92.50	90.21
	GTSRB	93.87	95.76	93.63	96.18	94.14	97.64	91.68
	STL10	83.23	82.92	84.73	83.64	81.85	81.52	78.50
	AVG	88.70	87.58	88.81	88.75	87.17	87.52	83.50
BDAF	ANIMALS10	80.21	86.29	88.82	90.52	86.88	85.60	84.54
	CIFAR10	82.76	80.59	87.44	78.35	85.32	80.79	79.23
	GTSRB	99.97	99.96	100.00	99.70	99.99	99.92	99.87
	STL10	70.76	77.54	78.44	78.99	75.62	82.06	79.99
	AVG	83.43	86.10	88.68	86.89	86.95	87.09	85.91

rate of 0.001, while the L_{buf} use a smaller learning rate of 0.0003. Additionally, the decay factor is set to 0.95.

B. Experiments Results

Table I compares robust accuracy (RA) across all settings. All results, except those of BDAF, are taken from Gen-AF, as our reproduced results are not consistent with the original ones. We evaluate 7 mainstream SSL methods across 4 datasets under 5 UAPs. Although Gen-AF achieves high robustness on several datasets, particularly GTSRB, it exhibits significant performance fluctuations across different settings. Specifically, out of 140 experiments, Gen-AF yields RA below 65% in 25 cases, primarily on CIFAR10, STL10, and ANIMALS10, and these underperforming results are marked in red.

By comparison, BDAF achieves significantly more stable and consistent performance overall. It explicitly classifies network layers based on their contributions to robustness and applies differentiated learning rates guided by buffer layers. This design preserves critical robustness-sensitive features, enhances adaptation to downstream tasks, and maintains improved model coordination. Among the 140 experimental settings, BDAF has only one instance where RA is slightly below 65% (64.06%) under AdvEncoder. Moreover, in 79 out of 140 experiments, accounting for the majority of settings, BDAF achieves higher RA than Gen-AF. Under the same attack and self-supervised framework, BDAF outperforms Gen-AF in 5 out of 7 frameworks. Notably, under DINO, BDAF achieves an average RA of 81.23%, exceeding Gen-AF by 13 percentage points. Despite its overall superiority, BDAF performs relatively poorly on a few simple datasets, such as GTSRB and ANIMALS10. This is because the expanded fine-tuning scope, while effective for generally enhancing robustness, may result in excessive parameter updates on datasets that are inherently easy to classify, thereby causing a slight decrease in RA. Overall, BDAF reaches an average RA of 83.54%, which is 3.35% higher than Gen-AF (80.19%), demonstrating consistent robustness gains.

The TA results are presented in Table II, with the standard training results taken from Gen-AF. We observe that while BDAF causes moderate performance declines in some experimental setups, it achieves substantial and consistent improvements across various other configurations. For example, the W-MSE encoder trained on ANIMALS10 attains a downstream TA of 73.60% with standard training, whereas BDAF increases the TA to 84.54%. These results demonstrate that the proposed method delivers competitive overall performance and robust generalization capabilities within SSL frameworks.

C. Ablation Study

In this section, we evaluate the impact of different modules and hyperparameters. Experiments are conducted on STL10 using an NNCLR encoder pretrained on CIFAR10. In all tables, the best and second-best results are highlighted in bold and underlined, respectively.

The effect of modules. We evaluate the effectiveness of the RALR and LLRD modules through ablation studies conducted under five UAPs. Specifically, S1 denotes a baseline model without RALR and LLRD, S2 incorporates only the RALR module, and S3 represents the full DLRF framework combining both RALR and LLRD. As shown in Table 3, S2 demonstrates that employing the RALR module alone significantly improves model performance, while S3 further enhances robustness across all experimental settings by integrating RALR with LLRD. These improvements are more pronounced on complex datasets such as STL10 and CIFAR10. Overall, our method strengthens adversarial resistance and enhances generalization, particularly in challenging scenarios.

TABLE III: The effect of modules

Setting	η_b	λ	TA	RA					
				UAP	UAP-EPGD	SSP	PAP	AdvEncoder	AVG
S1	0	1	78.17	78.19	72.54	73.02	71.59	69.80	73.03
S2(RALR)	0.0003	1	**80.12**	<u>80.56</u>	**78.99**	**80.92**	<u>76.12</u>	<u>73.69</u>	<u>77.66</u>
Ours(RALR+LLRD)	0.0003	0.95	<u>78.99</u>	**81.35**	<u>78.08</u>	<u>78.41</u>	**78.24**	**73.78**	**77.97**

The effect of the buffer-layer learning rate η_b. We further investigate the effect of the buffer-layer learning rate on the performance of our method, with the results summarized in Table IV. The findings indicate that an excessively small learning rate for the buffer layer fails to effectively stabilize gradient propagation and does not yield any noticeable improvement in model generalization. In contrast, an excessively large learning rate causes excessive modification of the model, resulting in degraded robustness preservation. These observations underscore the importance of selecting an appropriate learning rate for the buffer layer to achieve a balanced trade-off.

The effect of the decay factor λ. We investigate the effect of the decay factor λ on the performance of our method. As

TABLE IV: The effect of the buffer-layer learning rate η_b under λ= 0.95

η_b	TA	RA					
		UAP	UAP-EPGD	SSP	PAP	AdvEncoder	AVG
0	77.81	79.80	76.05	77.33	76.33	72.22	76.35
0.0001	74.42	80.06	76.40	77.16	76.77	72.41	76.56
0.0005	77.17	78.78	75.88	76.15	76.08	72.64	75.91
0.0003	**78.99**	**81.35**	**78.08**	**78.41**	**78.24**	**73.78**	**77.97**

shown in Table V, setting λ to 1 disables the decay mechanism, effectively reducing the strategy to a non-decay variant. When λ is set to 0.9, the model achieves the poorest performance in both TA and RA. In contrast, $\lambda = 0.95$ achieves the optimal trade-off between robustness and accuracy.

TABLE V: The effect of the decay factor λ under η_b= 0.0003

λ	TA	RA					
		UAP	UAP-EPGD	SSP	PAP	AdvEncoder	AVG
1	**80.12**	80.56	**78.99**	**78.92**	76.12	73.69	77.66
0.9	75.24	77.17	73.90	74.83	75.07	71.66	74.52
0.95	78.99	**81.35**	78.08	78.41	**78.24**	**73.78**	**77.97**

V. CONCLUSION

In this study, we propose BDAF, a robust fine-tuning framework. BDAF maintains the stability of local cross-layer feature propagation, while differentiated learning simultaneously preserves the encoder's original feature extraction capability and enhances downstream generalization. Extensive experiments demonstrate the effectiveness of BDAF across various self-supervised learning methods and datasets under UAPs.

In future work, we will investigate the applicability of our method to ImageNet pre-training and extend it to larger models, such as ResNet-50 and ViT. This will further validate the generality and scalability of our approach across various backbones and pre-training configurations.

REFERENCES

[1] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PmLR, 2020, pp. 1597–1607.

[2] X. Chen, S. Xie, and K. He, "An empirical study of training self-supervised vision transformers," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 9640–9649.

[3] B. Wang, Y. Yao, B. Viswanath, H. Zheng, and B. Y. Zhao, "With great training comes great vulnerability: Practical attacks against transfer learning," in *27th USENIX security symposium (USENIX Security 18)*, 2018, pp. 1281–1297.

[4] K. Liu, B. Dolan-Gavitt, and S. Garg, "Fine-pruning: Defending against backdooring attacks on deep neural networks," in *International symposium on research in attacks, intrusions, and defenses*. Springer, 2018, pp. 273–294.

[5] Z. Zhang, Y. Li, J. Wang, B. Liu, D. Li, Y. Guo, X. Chen, and Y. Liu, "Remos: Reducing defect inheritance in transfer learning via relevant model slicing," in *Proceedings of the 44th international conference on software engineering*, 2022, pp. 1856–1868.

[6] Y. Li, X. Lyu, N. Koren, L. Lyu, B. Li, and X. Ma, "Neural attention distillation: Erasing backdoor triggers from deep neural networks," *arXiv preprint arXiv:2101.05930*, 2021.

[7] B. Huang, M. Chen, Y. Wang, J. Lu, M. Cheng, and W. Wang, "Boosting accuracy and robustness of student models via adaptive adversarial distillation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 24 668–24 677.

[8] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[9] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

[10] H. Zhang, Y. Yu, J. Jiao, E. Xing, L. El Ghaoui, and M. Jordan, "Theoretically principled trade-off between robustness and accuracy," in *International conference on machine learning*. PMLR, 2019, pp. 7472–7482.

[11] Y. Wang, D. Zou, J. Yi, J. Bailey, X. Ma, and Q. Gu, "Improving adversarial robustness requires revisiting misclassified examples," in *International conference on learning representations*, 2019.

[12] R. Zhang, R. Qiang, S. A. Somayajula, and P. Xie, "Autolora: Automatically tuning matrix ranks in low-rank adaptation based on meta learning," *arXiv preprint arXiv:2403.09113*, 2024.

[13] Z. Liu and V. Ojha, "Regmix: Adversarial mutual and generalization regularization for enhancing dnn robustness," *arXiv preprint arXiv:2510.05317*, 2025.

[14] B. Li and Y. Li, "Adversarial training can provably improve robustness: Theoretical analysis of feature learning process under structured data," 2025. [Online]. Available: https://arxiv.org/abs/2410.08503

[15] S. Wang, J. Zhang, Z. Yuan, and S. Shan, "Pre-trained model guided fine-tuning for zero-shot adversarial robustness," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024, pp. 24 502–24 511.

[16] K. Zhu, X. Hu, J. Wang, X. Xie, and G. Yang, "Improving generalization of adversarial training via robust critical fine-tuning," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 4424–4434.

[17] Z. Zhou, M. Li, W. Liu, S. Hu, Y. Zhang, W. Wan, L. Xue, L. Y. Zhang, D. Yao, and H. Jin, "Securely fine-tuning pre-trained encoders against adversarial examples," in *2024 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2024, pp. 3015–3033.

[18] B. Gopal, H. Yang, J. Zhang, M. Horton, and Y. Chen, "Boosting adversarial robustness with CLAT: Criticality leveraged adversarial training," in *Proceedings of the 42nd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 267. PMLR, 13–19 Jul 2025, pp. 20 142–20 161.

[19] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[20] Y. Ban and Y. Dong, "Pre-trained adversarial perturbations," *Advances in neural information processing systems*, vol. 35, pp. 1196–1209, 2022.

[21] Z. Zhou, S. Hu, R. Zhao, Q. Wang, L. Y. Zhang, J. Hou, and H. Jin, "Downstream-agnostic adversarial examples," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 4345–4355.

[22] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1765–1773.

[23] Y. Deng and L. J. Karam, "Universal adversarial attack via enhanced projected gradient descent," in *2020 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2020, pp. 1241–1245.

[24] M. Naseer, S. Khan, M. Hayat, F. S. Khan, and F. Porikli, "A self-supervised approach for adversarial robustness," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 262–271.

[25] V. G. T. da Costa, E. Fini, M. Nabi, N. Sebe, and E. Ricci, "solo-learn: A library of self-supervised methods for visual representation learning," *Journal of Machine Learning Research*, vol. 23, no. 56, pp. 1–6, 2022. [Online]. Available: http://jmlr.org/papers/v23/21-1155.html

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

[27] A. Coates, A. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2011, pp. 215–223.

[28] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition," *Neural networks*, vol. 32, pp. 323–332, 2012.

[29] "Animals-10: Animal pictures of 10 different categories," Google Images, Aug. 2020, online, accessed from Google Images.