Enhancing Robustness in Incremental Learning with Adversarial Training

Seungju Cho*, **Hongsin Lee***, **Changick Kim**

Korea Advanced Institute of Science and Technology (KAIST)
{joyga, hongsin04, changick}@kaist.ac.kr

Abstract

Adversarial training is one of the most effective approaches against adversarial attacks. However, adversarial training has primarily been studied in scenarios where data for all classes is provided, with limited research conducted in the context of incremental learning where knowledge is introduced sequentially. In this study, we investigate Adversarially Robust Class Incremental Learning (ARCIL), which deals with adversarial robustness in incremental learning. We first explore a series of baselines that integrate incremental learning with existing adversarial training methods, finding that they lead to conflicts between acquiring new knowledge and retaining past knowledge. Furthermore, we discover that training new knowledge causes the disappearance of a key characteristic in robust models: a flat loss landscape in input space. To address such issues, we propose a novel and robust baseline for ARCIL, named **FL**atness-preserving **A**dversarial **I**ncremental learning for **R**obustness (FLAIR). Experimental results demonstrate that FLAIR significantly outperforms other baselines. To the best of our knowledge, we are the first to comprehensively investigate the baselines, challenges, and solutions for ARCIL, which we believe represents a significant advance toward achieving real-world robustness.

Introduction

Given the susceptibility of deep neural networks (DNNs) to adversarial attacks, adversarial training is recognized as the most effective defense method (Athalye, Carlini, and Wagner 2018; Wu, Xia, and Wang 2020; Wang et al. 2021; Wu et al. 2021). Furthermore, adversarial training has garnered significant attention not only for its role in improving adversarial robustness but also for enhancing feature representation and interpretation (Engstrom et al. 2019; Salman et al. 2020; Santurkar, Tsipras, and Madry 2020; Bai et al. 2021; Deng et al. 2021; Allen-Zhu and Li 2022; Kireev, Andriushchenko, and Flammarion 2022). Despite its effectiveness, adversarial training has not been sufficiently explored in the context of real-world scenarios with continuously evolving data. This leads us to address a crucial problem: ensuring robustness in incremental learning environments. We term this challenge Adversarially Robust Class Incremental Learning (ARCIL), highlighting the need for solutions that combine the strengths of adversarial training with the dynamic requirements of incremental learning.

In incremental learning, a central challenge is acquiring new knowledge without forgetting what has already been learned. To address this issue, incremental learning employs various methods, such as knowledge distillation (Li and Hoiem 2017; Rebuffi et al. 2017) and rehearsal techniques (Riemer et al. 2018; Buzzega et al. 2020). However, these methods do not take adversarial robustness into account, so we construct new baselines for ARCIL by applying adversarial training on those incremental learning in Table 1. This includes not only standard adversarial training methods (Madry et al. 2017; Zhang et al. 2019; Wang et al. 2020) but also adversarial distillation techniques (Goldblum et al. 2020; Zi et al. 2021; Huang et al. 2023) in place of the traditional knowledge distillation used in incremental learning, to ensure a fair comparison within the framework. Nonetheless, existing baselines are insufficient to solve ARCIL.

We identify several reasons why the baseline methods struggle to perform well. Firstly, the straightforward application of adversarial distillation leads to conflicts between learning new tasks and preserving knowledge from previous tasks. This issue is exacerbated in adversarial training, due to its adversarial input. Secondly, we figure out that incorporating new tasks causes the disappearance of one of the main characteristics of adversarial training, a flat loss landscape (Qin et al. 2019; Moosavi-Dezfooli et al. 2019). Given the abundance of research on adversarial robustness and flatness, we believe preserving flatness is important to maintain robustness. However, we observe in Table 1 that the baselines tend to forget the flatness of the loss for past tasks as new tasks are introduced. Lastly, ARCIL inherently suffers from a lack of training data, which is a significant issue since adversarial training generally requires more data than natural training (Rebuffi et al. 2021; Li and Spratling 2023; Yue et al. 2024).

To resolve these issues, we propose a **FL**atness-preserving **A**dversarial **I**ncremental learning for **R**obustness (**FLAIR**). The proposed method employs separated-logits (Rebuffi et al. 2017; Ahn et al. 2021) to ensure that the logits corresponding to past tasks remain unaffected when learning new tasks. This approach mitigates conflicts between retaining existing knowledge and acquiring new information while leveraging the benefits of adversarial distillation. Addition-

*These authors contributed equally.

Type	Methods	Formulation
AT	PGD-AT	$\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{S}_t}\left[l_{CE}(f_t(\boldsymbol{x}_{adv}),y)\right]$
	TRADES	$\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{S}_t}\left[l_{CE}(f_t(\boldsymbol{x}),y)+\alpha\cdot l_{KL}(f_t(\boldsymbol{x}_{adv})\|f_t(\boldsymbol{x}))\right]$
	MART	$\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{S}_t}\left[l_{BCE}(f_t(\boldsymbol{x}_{adv}),\mathbf{1}_y)+\alpha\cdot(1-\Pr(f_t(x)=y))\cdot l_{KL}(f_t(\boldsymbol{x}_{adv})\|f_t(\boldsymbol{x}))\right]$
I-AD	I-ARD	$\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{S}_t}\left[l_{CE}(f_t(\boldsymbol{x}_{adv}),y)+\beta\cdot l_{KL}([f_t(\boldsymbol{x}_{adv})]_0^{t-1}\|f_{t-1}(\boldsymbol{x}))\right]$
	I-RSLAD	$\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{S}_t}\left[l_{CE}(f_t(\boldsymbol{x}_{adv}),y)+\beta\cdot\left(\alpha\cdot l_{KL}([f_t(\boldsymbol{x}_{adv})]_0^{t-1}\|f_{t-1}(\boldsymbol{x}))+(1-\alpha)\cdot l_{KL}([f_t(\boldsymbol{x})]_0^{t-1}\|f_{t-1}(\boldsymbol{x}))\right)\right]$
	I-AdaAD	$\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{S}_t}\left[l_{CE}(f_t(\boldsymbol{x}_{adv}),y)+\beta\cdot\left(\alpha\cdot l_{KL}([f_t(\boldsymbol{x}_{adv})]_0^{t-1}\|f_{t-1}(\boldsymbol{x}_{adv}))+(1-\alpha)\cdot l_{KL}([f_t(\boldsymbol{x})]_0^{t-1}\|f_{t-1}(\boldsymbol{x}))\right)\right]$
Non-Rehearsal R-CIL	R-EWC-on	$\mathbb{E}_{(\boldsymbol{x},y)\sim D_t}\left[l_{CE}(f_t(\boldsymbol{x}_{adv}),y)+l_{EWC}(\theta_t,\theta_{t-1})\right]$
	R-LwF	$\mathbb{E}_{(\boldsymbol{x},y)\sim D_t}\left[l_{CE}(f_t(\boldsymbol{x}_{adv}),y)+\alpha\cdot l_{KL}([f_t(\boldsymbol{x})]_0^{t-1}\|f_{t-1}(\boldsymbol{x}))\right]$
	R-LwF-MC	$\mathbb{E}_{(\boldsymbol{x},y)\sim D_t}\left[l_{BCE}([f_t(\boldsymbol{x}_{adv})]_{t-1}^{t},\mathbf{1}_y)+l_{BCE}([f_t(\boldsymbol{x})]_0^{t-1},f_{t-1}(\boldsymbol{x}))\right]$
	R-SI	$\mathbb{E}_{(\boldsymbol{x},y)\sim D_t}\left[l_{CE}(f_t(\boldsymbol{x}_{adv}),y)+l_{SI}(\theta_t,\theta_{t-1})\right]$
Rehearsal R-CIL	R-ER	$\mathbb{E}_{(\boldsymbol{x},y)\sim D_t}\left[l_{CE}(f_t(\boldsymbol{x}_{adv}),y)\right]+\mathbb{E}_{(\boldsymbol{x},y)\sim M}\left[l_{CE}(f_t(\boldsymbol{x}_{adv}),y)\right]$
	R-ER-ACE	$\mathbb{E}_{(\boldsymbol{x},y)\sim D_t}\left[l_{ACE}(f_t(\boldsymbol{x}_{adv}),C_{curr})\right]+\mathbb{E}_{(\boldsymbol{x},y)\sim M}\left[l_{CE}(f_t(\boldsymbol{x}_{adv}),y)\right]$
	R-DER	$\mathbb{E}_{(\boldsymbol{x},y)\sim D_t}\left[l_{CE}(f_t(\boldsymbol{x}_{adv}),y)\right]+\mathbb{E}_{(\boldsymbol{x},\mathbf{z})\sim M}\left[\alpha\cdot l_{MSE}(f_t(\boldsymbol{x}_{adv}),\mathbf{z})\right]$
	R-DER++	$\mathbb{E}_{(\boldsymbol{x},y)\sim D_t}\left[l_{CE}(f_t(\boldsymbol{x}_{adv}),y)\right]+\mathbb{E}_{(\boldsymbol{x},\mathbf{z},y)\sim M}\left[\alpha\cdot l_{MSE}(f_t(\boldsymbol{x}_{adv}),\mathbf{z})+\beta\cdot l_{CE}(f_t(\boldsymbol{x}_{adv}),y)\right]$
	R-iCaRL	$\mathbb{E}_{(\boldsymbol{x},y)\sim D_t\cup B_{t-1}}\left[l_{BCE}([f_t(\boldsymbol{x}_{adv})]_{t-1}^{t},\mathbf{1}_y)+l_{BCE}([f_t(\boldsymbol{x})]_0^{t-1},f_{t-1}(\boldsymbol{x}))\right]$
	R-LUCIR	$\mathbb{E}_{(\boldsymbol{x},y)\sim D_t\cup B_{t-1}}\left[l_{CE}(f_t(\boldsymbol{x}_{adv}),y)+\alpha\cdot l_{dis}^{G}(\boldsymbol{x}_{adv})\right]+\mathbb{E}_{(\boldsymbol{x},y)\sim B_{t-1}}\left[\beta\cdot l_{mr}(\boldsymbol{x}_{adv})\right]$
ARCIL	TABA	$\mathbb{E}_{(\boldsymbol{x},y)\sim D_t\cup B_{t-1}\cup A_{\text{TABA}}}\left[l_{BCE}([f_t(\boldsymbol{x}_{adv})]_{t-1}^{t},\mathbf{1}_y)+l_{BCE}([f_t(\boldsymbol{x}_{adv})]_0^{t-1},f_{t-1}(\boldsymbol{x}))\right]$
	FLAIR	$\mathbb{E}_{(\boldsymbol{x},y)\sim S_t}\left[l_{BCE}([f_t(\boldsymbol{x}_{adv})]_{t-1}^{t},\mathbf{1}_y)+\alpha\cdot l_{BCE}([f_t(\boldsymbol{x}_{adv})]_0^{t-1},f_{t-1}(\boldsymbol{x}_{adv}))+\beta\cdot l_{FPD}(\boldsymbol{x},\boldsymbol{x}_{adv};f_t,f_{t-1})\right]$

Table 1: Different methods to fit in ARCIL setting. Type AT stands for naive Adversarial Training on ARCIL, while I-AD is revised adversarial distillation methods by considering the previous task model as a teacher model. R-CIL consists of a set of revised CIL methods, mainly by changing the input $\boldsymbol{x}$ of the learning incremented task to $\boldsymbol{x}_{adv}$. R-CIL is further categorized into two subtypes: Non-Rehearsal and Rehearsal types. Details of specific notations can be found in the appendix.

ally, FLAIR provides a straightforward method for preserving loss flatness. By employing Taylor approximation, we observe that the output difference between clean and adversarial example, encapsulates gradient and hessian information. Preserving this output difference ensures that gradient and hessian details are retained, thereby maintaining the flatness. Furthermore, we employ data augmentation techniques to address the issue of insufficient training data.

In summary, our contributions are as follows:

- We address the less well-studied problem of ARCIL and propose baselines that incorporate both incremental learning and adversarial training.
- We systematically analyze three key issues with existing baselines for ARCIL and propose FLAIR, which provides effective solutions to address these challenges.
- We evaluate the performance in the ARCIL setting, and FLAIR surpasses baseline results, achieving superior performance.

Related Work

In this section, we review adversarial training and incremental learning approaches in the context of image classification.

Adversarial Training

Adversarial training improves model robustness by addressing the following min-max problem.

$$\begin{aligned}&\arg\min_{\theta}\mathbb{E}_{(\boldsymbol{x},y)\sim D}\left[l_{min}\left(f_\theta(\boldsymbol{x}_{adv}),y\right)\right],\\&\text{where }\boldsymbol{x}_{adv}=\arg\max_{\|\boldsymbol{x}_{adv}-\boldsymbol{x}\|_p\leq\epsilon}l_{max}\left(f_\theta(\boldsymbol{x}_{adv}),y\right),\end{aligned}\tag{1}$$

where D is the data distribution, θ is parameters of the model f, ϵ is the maximum perturbation limit, and l_{min} and l_{max} denote loss functions for min-max problem. Most adversarial training uses a multi-step based Projected Gradient Attack (PGD) (Madry et al. 2017) for the inner maximization problem, and several regularization losses have been proposed to solve the outer minimization problem of (1). Representatively, TRADES (Zhang et al. 2019) incorporates the KL divergence loss between the output of clean and adversarial images, and MART (Wang et al. 2020) introduces per-sample weights through the confidence of each sample.

To enhance the effectiveness of adversarial training, various techniques have been proposed, including data augmentation (Rebuffi et al. 2021; Li and Spratling 2023) and knowledge distillation (Goldblum et al. 2020; Zi et al. 2021; Zhu et al. 2021; Maroto, Ortiz-Jiménez, and Frossard 2022; Zhao et al. 2022; Huang et al. 2023). Among them, meth-

ods such as ARD (Goldblum et al. 2020), RSLAD (Zi et al. 2021), and AdaAD (Huang et al. 2023) use knowledge distillation to transfer the robustness of large teacher models. These adversarial distillation approaches significantly improve the performance of small models when a robust teacher model is available.

A flat loss landscape is closely related to generalization, and therefore, it impacts adversarial robustness performance (Izmailov et al. 2018; Chen et al. 2020). Although several studies have proposed methods to achieve a flatter loss landscape (Moosavi-Dezfooli et al. 2019; Qin et al. 2019), most research on adversarial training has focused on scenarios where all data is provided upfront. In incremental settings, where data is received sequentially and past data may be lost, maintaining a flatter loss function becomes challenging as new data is introduced and old data is not available. Our research investigates methodologies for achieving adversarial robustness in such incremental settings while addressing the preservation of a flatter loss function.

Class Incremental Learning

Class incremental learning (CIL) assumes data from new, non-overlapping classes is provided sequentially. The distribution shift leads to a phenomenon known as *catastrophic forgetting* (Goodfellow et al. 2013), where the model forgets previously learned information as it learns new data. The central challenge in CIL is to integrate new data effectively while preserving knowledge from earlier tasks.

Various methods have been proposed to tackle this challenge, including knowledge distillation (Li and Hoiem 2017; Rebuffi et al. 2017; Ahn et al. 2021), weight regularization (Kirkpatrick et al. 2017; Li and Hoiem 2017; Zenke, Poole, and Ganguli 2017; Schwarz et al. 2018), or memory buffer (Rebuffi et al. 2017; Riemer et al. 2018; Hou et al. 2019; Buzzega et al. 2020; Caccia et al. 2021). Among these, memory buffer-based approaches, particularly rehearsal methods, have become strong baselines due to their effectiveness and compatibility with other techniques (Rebuffi et al. 2017; Buzzega et al. 2020; Riemer et al. 2018; Hou et al. 2019; Caccia et al. 2021). In contrast, non-rehearsal methods have been extensively studied because they are more applicable to real-world scenarios (Boschini et al. 2022; Aljundi et al. 2019; Farquhar and Gal 2018).

While many CIL methods aim to mitigate catastrophic forgetting, there is a notable research gap concerning adversarially perturbed data. Recent studies have explored adversarial robustness in incremental learning but have primarily relied on simple data augmentation techniques and may not have fully addressed the complexities of the problem (Bai et al. 2023). Our work addresses this gap by offering a more robust and comprehensive examination of the impact of adversarial training.

Proposed Method

We first analyze the underlying issues of ARCIL and then introduce our proposed method.

Problem Formulation

A model f is sequentially trained on tasks $1, 2, \ldots, T$, with f_t denoting the model trained on task t. In each individual task t, inputs $\boldsymbol{x} \in \mathbb{R}^d$ and labels $y \in \mathbb{R}$ are independently and identically drawn from the task-specific data distribution $D_t = \{X_t, Y_t\}$, where $D_i \cap D_j = \emptyset$ for $i \neq j$. Here, X_t and Y_t are the set of inputs and true labels of task t. The goal is to maintain robustness against adversarial attacks in new tasks while preserving the robustness learned in previous tasks.

We investigate two incremental learning settings. The first setting involves the absence of any data from previous tasks (Non-Rehearsal), while the second setting deals with having a small amount of data from previous tasks (Rehearsal). For the setting with a rehearsal buffer, we utilized a limited and constant memory size for storing past data, as proposed in previous works such as (Rebuffi et al. 2017; Riemer et al. 2018; Hou et al. 2019; Buzzega et al. 2020; Bai et al. 2023). In our method, the rehearsal buffer B_{t-1} is selected with the herding algorithm after training on each task as in (Rebuffi et al. 2017), and concatenate the buffer at the beginning of each incremental task on the current dataset, denoted as $D_t \cup B_{t-1}$. Regardless of using the rehearsal buffer, we denote the current training dataset as S_t, instead of D_t or $D_t \cup B_{t-1}$ if there is no misunderstanding.

Adversarial Distillation with Separated Logit

Knowledge distillation is an effective technique for achieving high performance in adversarial training by imparting robustness from a strong teacher model (Goldblum et al. 2020; Huang et al. 2023). Similarly, knowledge distillation can be utilized in incremental learning by leveraging a model trained on past tasks as a teacher (Li and Hoiem 2017; Rebuffi et al. 2017). Thus, intuitively incorporating these two methods to address ARCIL can provide a reasonable solution.

$$l(\boldsymbol{x}, y) = l_{CE}(f_t(\boldsymbol{x}_{adv}), y) + \alpha \cdot l_{AD}(\boldsymbol{x}, \boldsymbol{x}_{adv}; f_t, f_{t-1})), \tag{2}$$

where α is a hyperparameter that controls the strength of knowledge distillation from the past task model to the current model, and l_{AD} represents the adversarial distillation loss function, such as ARD (Goldblum et al. 2020), RSLAD (Zi et al. 2021), or AdaAD (Huang et al. 2023). However, naively applying adversarial distillation to incremental learning decreases overall performance as in Table 2 and Table 3.

We attribute the insufficient performance to the conflicting objectives of learning new knowledge and maintaining previous robustness. This represents a common trade-off in incremental learning known as the stability-plasticity dilemma (Mermillod, Bugaiska, and Bonin 2013), but is more exacerbated in ARCIL because of adversarial input, $\boldsymbol{x}_{adv}$. In Equation (1), the adversarial input maximizes the inner loss function, leading to new task data being adversarially crafted, often increasing the logit of one of the previous task classes. In such cases, reducing the attacked logit of the adversarial input is necessary to learn the new task, but this can result in even more severe forgetting than in natural incremental learning. In contrast, distillation aims to preserve

the logit of previous tasks. This conflict between the two loss functions explains why naively applying adversarial distillation does not work well in ARCIL.

To prevent this conflict, we design the loss function to avoid affecting the logits of past classes when learning new tasks, using binary cross-entropy loss only on the new class indices. This simple technique not only prevents conflicts but also allows for greater flexibility in the distillation process. In summary, in the incremental setting, we apply adversarial distillation with substitution of cross entropy $l_{CE}(\boldsymbol{x}_{adv}, y)$ into binary cross-entropy loss with current task logit $l_{BCE}([f_t(\boldsymbol{x}_{adv})]_{t-1}^{t}, \mathbf{1}_y)$. Here, $[\cdot]_{t-1}^{t}$ represents a slicing operation that returns the outputs immediately after the $(t-1)$-th task up to and including the t-th task output, and $\mathbf{1}_y$ is one-hot vector for label y. To preserve past knowledge through distillation, methods such as ARD, RSLAD, or AdaAD can be used for $l_{AD}(\boldsymbol{x}, \boldsymbol{x}_{adv}; f_t, f_{t-1})$. We experimentally select AdaAD for our method as follows: $l_{BCE}([f_t(\boldsymbol{x}_{adv})]_{0}^{t-1}, f_{t-1}(\boldsymbol{x}_{adv}))$.

Overall, we define the adversarial distillation with separated logit (ADSL) as follows:

$$\begin{aligned} l_{ADSL}(\boldsymbol{x}, y) = l_{BCE}([f_t(\boldsymbol{x}_{adv})]_{t-1}^{t}, y) \\ + \alpha \cdot l_{BCE}([f_t(\boldsymbol{x}_{adv})]_{0}^{t-1}, f_{t-1}(\boldsymbol{x}_{adv})), \end{aligned} \tag{3}$$

where α is a hyperparameter that controls the weight of preserving the previous model's knowledge.

Flatness Preserving Distillation

To further improve the capability of retaining knowledge, we investigate whether adversarially robust models maintain their distinctive characteristics while learning new tasks. We focus on the flatness of the loss landscape, a characteristic of robust models (Moosavi-Dezfooli et al. 2019; Qin et al. 2019), and assess it using gradient and Hessian calculations. We define Gradient Forgetting (GF) and Hessian Forgetting (HF) to evaluate the preservation of loss flatness.

$$\begin{aligned} \text{GF} &= \frac{1}{T-1} \sum_{i=1}^{T-1} \mathbb{E}_{(\boldsymbol{x},y)\sim D_i} \left[\left\| \nabla f_T(\boldsymbol{x}) - \nabla f_i(\boldsymbol{x}) \right\|_2 \right], \\ \text{HF} &= \frac{1}{T-1} \sum_{i=1}^{T-1} \mathbb{E}_{(\boldsymbol{x},y)\sim D_i} \left[\left\| \mathbf{H}_T(\boldsymbol{x}) - \mathbf{H}_i(\boldsymbol{x}) \right\|_F \right], \end{aligned} \tag{4}$$

where D_i is i-th task test dataset, and $\mathbf{H}_i$ denotes hessian matrix of i-th task learned model. We find that all baselines in Table 1 exhibit high GF and HF, indicating a disappearance of flatness, as shown in Table 4. This suggests that the loss landscape learned through adversarial training for each task is altered, which we hypothesize is linked to robustness forgetting. Since the flatness of the loss landscape disappears after learning new tasks, we term this phenomenon the *flatness forgetting* problem.

To address this problem, we consider the gradient and hessian indirectly, as direct calculation of the gradient and hessian is computationally expensive. For a clean input $\boldsymbol{x}$ from the data of the current training dataset S_t, the adversarial perturbed input is given by:

$$\boldsymbol{x}_{adv} = \boldsymbol{x} + \boldsymbol{\delta}. \tag{5}$$

For a small perturbation $\boldsymbol{\delta}$, the output of $\boldsymbol{x}_{adv}$ given the t-th task model f_t can be approximated using Taylor expansion through the input space.

$$f_t(\boldsymbol{x}_{adv}) = f_t(\boldsymbol{x}) + \nabla f_t(\boldsymbol{x})^T \boldsymbol{\delta} + \frac{1}{2} \boldsymbol{\delta}^T \mathbf{H}_t(\boldsymbol{x}) \boldsymbol{\delta}, \tag{6}$$

where $\nabla f_t(\boldsymbol{x})$ and $\mathbf{H}_t$ are the gradient and hessian matrix of the t-th task model, and we neglect higher order terms. Similarly, the output of the $(t-1)$-th task model can also be approximated using Taylor expansion given the t-th task input $\boldsymbol{x}_{adv}$.

Maintaining the first-order and second-order terms in (6) between t and $(t-1)$-th task model is necessary to preserve the flatness. Therefore, we exploit the subtraction between clean and adversarial outputs as follows:

$$\Delta f_t = \nabla f_t(\boldsymbol{x})^T \boldsymbol{\delta} + \frac{1}{2} \boldsymbol{\delta}^T \mathbf{H}_t(\boldsymbol{x}) \boldsymbol{\delta}, \tag{7}$$

where $\Delta f_t = f_t(\boldsymbol{x}_{adv}) - f_t(\boldsymbol{x})$. Here, we neglect the high-order term due to the small magnitude of the perturbation $\boldsymbol{\delta}$. As a result, we can efficiently distill both the gradient and hessian information of the $(t-1)$-th task model into t-th task model by using the difference between clean and adversarial inputs:

$$l_{FPD}(\boldsymbol{x}, \boldsymbol{x}_{adv}; f_t, f_{t-1}) = D\left(\Delta f_t \, , \Delta f_{t-1}\right), \tag{8}$$

where D is a difference metric such as the KL divergence loss.

Augmentation Training Data

Lastly, we highlight another challenge for ARCIL: the lack of training data in each task. It is well known that AT requires a large amount of training data to capture richer representations for robustness sufficiently (Rebuffi et al. 2021; Li and Spratling 2023; Yue et al. 2024). However, in ARCIL, the number of data points from previous and future tasks is minimal or nonexistent, unlike in standard AT. A widely used and effective technique for enhancing features in AT is data augmentation. Although TABA (Bai et al. 2023) adopts Mixup augmentation on the selected buffer samples, we find that simple yet effective augmentation is sufficient to enhance robustness in ARCIL.

Yue et al. asserts that utilizing a broader range of augmentations beyond the commonly used horizontal flip would be more effective in long-tailed distributions, which often lack sufficient training data for specific classes. The rationale behind this is that while traditional AT benefits from a wealth of data for each class, making simple augmentations sufficient, the lack of training data in the long-tailed distributions is insufficient for robust learning. The rationale behind this is that while traditional AT benefits from a wealth of data for each class, making simple augmentations sufficient, the lack of training data in the long-tailed distributions is insufficient for robust learning. Similarly, ARCIL scenarios have partial access to past data, thus requiring more diverse augmentations would be effective. To enhance performance, we follow the methods proposed by Yue et al. by introducing RandAugment (RA) (Cubuk et al. 2020) or AutoAugment (AuA) (Cubuk et al. 2019).

Type	Method	S-CIFAR10				S-CIFAR100				S-SVHN			
		Clean↑	PGD↑	AA↑	R-BWT↑	Clean↑	PGD↑	AA↑	R-BWT↑	Clean↑	PGD↑	AA↑	R-BWT↑
AT	PGD-AT	18.98	16.30	16.27	-72.36	8.14	4.82	4.73	-39.54	19.10	10.72	6.45	-73.80
	TRADES	18.26	15.83	15.81	-70.45	8.27	5.78	5.61	-48.32	19.07	11.02	6.78	-76.96
	MART	18.78	16.37	16.32	-73.53	8.19	5.60	5.35	-47.36	18.47	11.50	6.90	-76.22
I-AD	I-ARD	19.03	16.33	16.32	-72.81	8.28	5.28	5.14	-44.06	20.81	10.07	9.10	-71.73
	I-RSLAD	18.84	16.10	16.10	-71.80	8.38	5.29	5.15	-44.50	19.07	10.42	6.31	-71.95
	I-AdaAD	18.89	16.19	16.17	-71.90	8.46	5.39	5.12	-43.80	18.99	10.70	6.49	-72.15
R-CIL	R-LwF	18.98	16.31	16.28	-72.66	8.28	5.08	5.03	-42.56	19.16	10.31	6.22	-72.32
	R-LwF-MC	40.24	15.85	14.77	-34.05	27.16	8.16	6.54	-19.68	58.02	5.01	3.20	-42.98
	R-EWC-on	18.08	8.47	8.16	-58.98	6.62	2.78	2.68	-19.74	11.12	7.02	4.65	-63.86
	R-SI	18.40	15.54	15.50	-67.78	8.45	5.40	5.26	-45.52	18.50	11.15	6.65	-73.53
ARCIL	**FLAIR**	43.64	18.42	17.02	**-22.09**	28.72	11.02	8.59	-15.06	65.16	16.25	15.96	**-16.86**
	FLAIR+	**44.22**	**22.42**	**20.46**	-24.66	**29.47**	**13.60**	**10.20**	**-14.26**	**66.26**	**30.04**	**27.21**	-17.53

Table 2: Clean, 20-step PGD, AutoAttack (AA) accuracy (%), and Robust Backward Transfer (R-BWT) measured on ResNet-18 for S-CIFAR10, S-CIFAR100, and S-SVHN without a memory buffer.

Flatness-preserving Adversarial Incremental learning for Robustness

We propose a novel method for ARCIL, naming our technique ***FL**atness-preserving **A**dversarial **I**ncremental **L**earning for **R**obustness* (**FLAIR**). Our training loss function is as follows.

$$\begin{aligned} l_{FLAIR}(\boldsymbol{x}, y) = & \, l_{BCE}([f_t(\boldsymbol{x}_{adv})]_{t-1}^{t}, y) \\ & + \alpha \cdot l_{BCE}([f_t(\boldsymbol{x}_{adv})]_{0}^{t-1}, f_{t-1}(\boldsymbol{x}_{adv})) \\ & + \beta \cdot l_{FPD}(\boldsymbol{x}, \boldsymbol{x}_{adv}; f_t, f_{t-1}) \end{aligned}$$

where α and β are hyper-parameter. Additionally, when we train our method with augmentations such as RA or AuA, we denote it as **FLAIR+**.

Experiments

Experimental Settings

Datasets We conducted experiments with and without a memory buffer on the following datasets. *Split CIFAR-10 (S-CIFAR10)* divides CIFAR-10 (Krizhevsky 2009) into five tasks, each consisting of two classes. *Split CIFAR-100 (S-CIFAR100)* divides CIFAR-100 (Krizhevsky 2009) into ten tasks, with ten classes per task. *Split SVHN (S-SVHN)* divides the SVHN (Netzer et al. 2011) into five tasks, with two classes per task. Additionally, *Split TinyImageNet (S-TinyImageNet)*, which divides Tiny ImageNet (Le and Yang 2015) into ten tasks with 20 classes per task using a memory buffer, can be found in the appendix.

Training For adversarial training, we used ten steps of PGD attack with a random start, and the maximum perturbation is limited to $\epsilon_\infty = 8/255$, while each step is taken with a step size of $2/255$. In all experiments, we utilized the ResNet-18 architecture (He et al. 2016), with additional results on MobileNetV2 (Sandler et al. 2018) provided in the appendix. We conducted a grid search over the hyperparameters $\{0,0.5,1,2,4\}$ for our method and all baseline methods, reporting the best results. The code can be accessed at *https://github.com/HongsinLee/FLAIR*.

Baseline We devise four type baselines for solving ARCIL to discuss our proposed methods fairly: AT, AD, R-CIL, and ARCIL in Table 1. AT represents naive adversarial training methods including PGD-AT (Madry et al. 2017), TRADES (Zhang et al. 2019), and MART (Wang et al. 2020) on ARCIL without considering any techniques for preventing forgetting. I-AD is revised adversarial distillation methods with learning new tasks for ARCIL by considering the previous task model as a teacher model, including I-ARD (Goldblum et al. 2020), I-RSLAD (Zi et al. 2021), and I-AdaAD (Huang et al. 2023), where the prefix "I-" indicates modifications for the incremental setting derived from the original methods. R-CIL consists of adversarially trained CIL baselines using both Non-Rehearsal and Rehearsal methods. For Non-Rehearsal R-CIL, the methods include R-EWC-on (Kirkpatrick et al. 2017), R-LwF (Li and Hoiem 2017), R-LwF-MC (Rebuffi et al. 2017), and R-SI (Zenke, Poole, and Ganguli 2017), while for Rehearsal R-CIL, the methods include R-ER (Riemer et al. 2018), R-ER-ACE (Caccia et al. 2021), R-DER/DER++ (Buzzega et al. 2020), R-iCaRL (Rebuffi et al. 2017), and R-LUCIR (Hou et al. 2019), where the prefix "R-" indicates modifications to enhance robustness in CIL methods. We implement TABA[1] for type ARCIL. Detailed settings are in the appendix.

Evaluation We measured clean, 20 steps of PGD, and AutoAttack (AA) (Croce and Hein 2020) accuracy after learning all incremental tasks with $\epsilon_\infty = 8/255$. To measure forgetting on robustness, we consider the robust backward transfer (R-BWT) metric as follows.

$$\text{R-BWT} = \frac{1}{T-1} \sum_{t=1}^{T-1} (RA_{T,t} - RA_{t,t}), \tag{9}$$

where $RA_{i,j}$ denotes PGD accuracy of task j after learning task i. Accuracy of past task usually drops as new tasks are learned, so a higher R-BWT means less forgetfulness.

[1] No open-source code is available for TABA. As a result, our results may differ from those reported for the original TABA. For further comparison with TABA results, please refer to the appendix.

Type	Method	S-CIFAR10				S-CIFAR100				S-SVHN			
		Clean↑	PGD↑	AA↑	R-BWT↑	Clean↑	PGD↑	AA↑	R-BWT↑	Clean↑	PGD↑	AA↑	R-BWT↑
AT	PGD-AT	52.73	24.34	23.86	-59.35	19.32	7.33	7.20	-29.24	62.76	22.54	18.62	-56.38
	TRADES	40.29	22.99	22.03	-63.16	17.76	9.03	8.68	-38.90	49.83	20.53	16.70	-63.91
	MART	51.08	28.26	26.09	-59.40	19.67	8.44	8.18	-34.99	60.47	23.79	19.89	-52.93
I-AD	I-ARD	50.52	26.99	26.12	-57.00	18.84	8.14	7.94	-36.69	56.06	20.53	15.79	-57.95
	I-RSLAD	49.89	27.29	26.17	-56.60	18.72	7.96	7.64	-36.02	53.39	19.17	14.19	-58.85
	I-AdaAD	52.06	27.98	27.10	-56.16	20.19	8.07	7.66	-35.03	56.25	20.89	15.78	-57.13
R-CIL	R-ER	52.03	24.23	23.85	-58.04	18.09	7.00	6.82	-30.21	57.76	21.00	17.51	-57.92
	R-ER-ACE	62.93	20.17	19.57	-23.05	31.87	7.00	6.74	-15.14	78.36	27.26	26.68	-25.29
	R-DER	24.30	16.21	16.10	-61.24	16.13	7.13	6.06	-38.82	33.17	11.66	8.08	-66.56
	R-DER++	27.75	16.27	16.02	-70.53	19.78	7.39	6.95	-31.43	32.27	11.29	7.50	-67.11
	R-iCaRL	57.78	27.29	25.87	-13.51	36.47	9.73	8.20	-17.92	76.47	18.02	17.13	-38.59
	R-LUCIR	62.90	29.23	26.12	-30.71	23.88	8.80	7.74	-24.91	61.04	25.38	20.28	-51.45
ARCIL	TABA	59.94	25.31	24.18	-18.75	28.44	8.02	7.16	-22.06	60.30	16.81	12.03	-35.12
	FLAIR	**63.81**	30.28	27.65	**-12.08**	**38.66**	17.04	13.45	**-7.87**	80.83	31.11	29.77	**-1.58**
	FLAIR+	61.29	**33.35**	**30.06**	-12.49	38.04	**18.03**	**14.43**	-8.43	**83.83**	**44.33**	**40.10**	-2.74

Table 3: Clean, 20-step PGD, AutoAttack (AA) accuracy (%), and Robust Backward Transfer (R-BWT) measured on ResNet-18 for S-CIFAR10, S-CIFAR100, and S-SVHN with 2000 size of memory buffer.

Main Results

In Table 2 and Table 3, we present a summary of the experimental results for all methods. The low performance of type AT indicates that standard AT is not appropriate for ARCIL. The insufficient performance of type I-AD indicates that distillation methods without considering separated logits result in high R-BWT, failing to resolve the issue of forgetting. Moreover, most methods that perform well in standard CIL show significantly lower performance, indicating that simply applying AT to CIL is insufficient for addressing ARCIL. Our methods achieve the highest clean and robust accuracy across all datasets, demonstrating the effectiveness of our approach. Specifically, R-BWT, which measures the degree of forgetting, also achieved the highest score, indicating that our method effectively mitigates forgetting.

Method	S-CIFAR10		S-CIFAR100	
	GF↓	HF↓	GF↓	HF↓
PGD-AT	1.641	2.184	1.832	2.475
TRADES	1.648	2.167	1.751	2.340
MART	1.603	2.409	1.732	2.421
I-ARD	1.802	2.190	1.925	2.110
I-RSLAD	1.808	2.195	1.888	2.051
I-AdaAD	1.827	2.114	1.842	1.950
R-LwF	1.553	2.003	1.731	2.256
R-LwF-mc	0.994	1.204	0.895	1.052
R-EWC-on	1.834	2.841	1.745	2.629
R-SI	1.578	2.119	1.773	2.342
FLAIR w/o FPD	0.964	1.139	0.561	0.616
FLAIR	0.582	0.665	0.492	0.539
FLAIR+	**0.557**	**0.657**	**0.423**	**0.463**

Table 4: Gradient Forgetting (GF), and Hessian Forgetting (HF). 'w/o FPD' indicates without the FPD method.

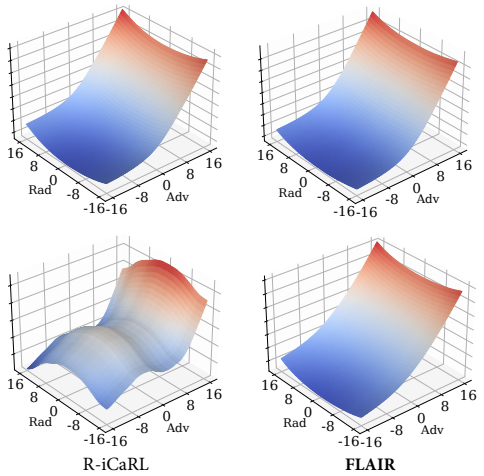

Figure 1: Loss landscape for each method at the beginning task t (top row) and after learning task t (bottom row).

We measure gradient and Hessian forgetting, as defined in Equation (4) to validate that FPD effectively preserves flatness. In the Table 4, note that all baselines exhibit the flatness forgetting problem, even including FLAIR without FPD. This indicates that adversarial distillation alone is insufficient to address the flatness forgetting problem, resulting in a lack of robustness. On the other hand, applying FPD achieves the smallest GF and HF, demonstrating its effectiveness in preserving flatness and leading to strong robustness. In Figure 1, we assess the loss landscape for flatness using the visualization approach (Park and Lee 2021). We plot cross-entropy loss projected in two directions: adversarial and random direction. Comparing our method with the reasonably performing baseline R-iCaRL, we find that our approach effectively preserves flatness.

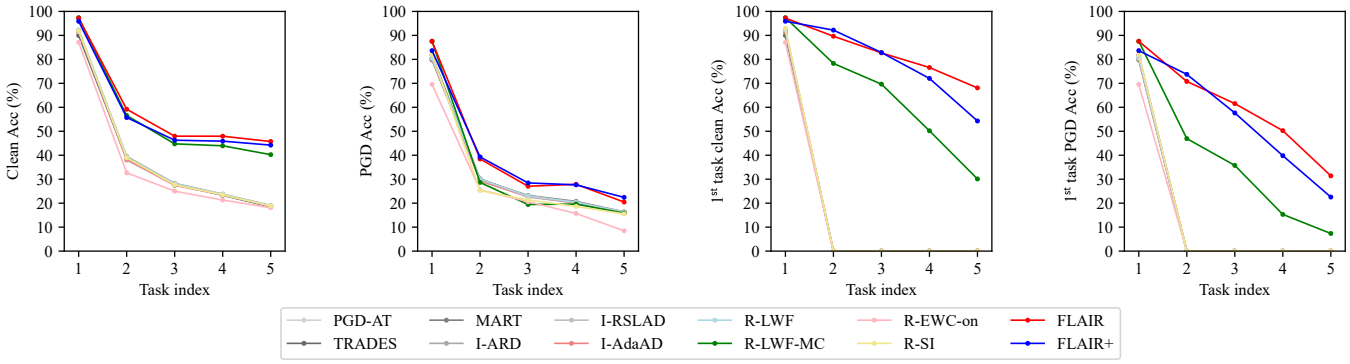

Figure 2: Clean *(Left)* and 20-step PGD *(Middle left)* accuracy on all test datasets from the first to the current task, and clean *(Middle right)* and 20-step PGD *(Right)* accuracy on the 1st task's test dataset. Results are from various baselines and our methods on S-CIFAR100 across incremental task steps.

Ablation Studies

More detailed experimental settings and additional results are available in the appendix.

Experimental results after each task In Figure 2, we illustrate the clean and PGD accuracy across incremental task steps for both the overall test dataset and the 1st task's test dataset. Our methods show strong performance with minimal forgetting. Notably, except R-EWC-on and our methods, all other baselines exhibit a complete drop in clean and PGD accuracy on the 1st task's test dataset after the 1st task, indicating full forgetting of previous knowledge. This suggests that most baselines fail to retain knowledge from previous tasks when learning new ones, which leads to nearly ideal classification performance only on the most recent task's training dataset. Consequently, their clean and robust accuracy approach $100/T(\%)$, as shown in Table 2.

Sensitivity of hyperparameters In Figure 3, we measured hyperparameter sensitivity using the sum of clean and AutoAttack accuracy. Generally, we see that performance improves as α and β increase, with high performance within the range of $\alpha \in [0.5, 1]$ and $\beta \in [1, 2]$. we can see that the sensitivity does not appear to be significant. Therefore, we selected the hyperparameters through a grid search, and the corresponding values are provided in the appendix.

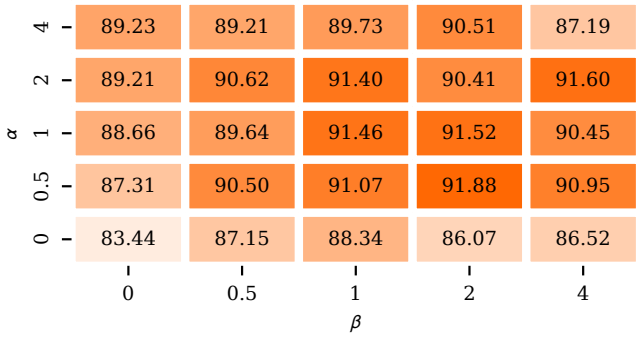

Figure 3: Sensitivity of hyperparameter on S-CIFAR10 with 2000 size of memory buffer.

Effects of each component of FLAIR In Table 5, we investigated the effects of three core techniques in FLAIR, namely Adversarial Distillation with Separated Logit (ADSL), Flatness Preserving Distillation (FPD), and Augmentation (AUG). As shown in the Table 5, each component contributes to performance improvement compared to when it is omitted. Both ADSL and FPD reduce forgetting, while augmentation boosts the robustness. Overall, the best results are achieved when all techniques are employed together.

ADSL	FPD	AUG	Clean↑	PGD↑	AA ↑	R-BWT↑
△			22.55	11.07	9.23	-53.34
✓			40.24	16.60	15.84	-28.41
	✓		41.16	17.57	16.24	-31.21
		✓	19.02	16.42	16.38	-72.00
✓		✓	40.23	20.10	18.07	-33.79
	✓	✓	43.92	21.94	19.66	-25.21
✓	✓		43.64	18.42	17.02	**-22.09**
✓	✓	✓	**44.22**	**22.42**	**20.46**	-24.66

Table 5: Impact of each component of FLAIR on S-CIFAR10 without a memory buffer. The symbol △ indicates ADSL with $\alpha = 0$.

Conclusion

We have introduced FLAIR, a novel method designed to tackle the ARCIL problem. Our analysis identifies three major challenges within ARCIL: (1) the conflict between learning new tasks and retaining knowledge from previous tasks through distillation, (2) the disappearance of loss flatness, and (3) insufficient training data. To address these issues, FLAIR employs separated logits to reduce conflicts between new and old task knowledge, distills output differences from previous models to maintain loss flatness, and incorporates data augmentation to counteract the lack of training data. This comprehensive strategy proves highly effective, establishing FLAIR as a strong baseline in this field.

References

Ahn, H.; Kwak, J.; Lim, S.; Bang, H.; Kim, H.; and Moon, T. 2021. Ss-il: Separated softmax for incremental learning. In *Proceedings of the IEEE/CVF International conference on computer vision*, 844–853.

Aljundi, R.; Lin, M.; Goujaud, B.; and Bengio, Y. 2019. Gradient based sample selection for online continual learning. *Advances in neural information processing systems*, 32.

Allen-Zhu, Z.; and Li, Y. 2022. Feature purification: How adversarial training performs robust deep learning. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, 977–988. IEEE.

Athalye, A.; Carlini, N.; and Wagner, D. 2018. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International conference on machine learning*, 274–283. PMLR.

Bai, T.; Chen, C.; Lyu, L.; Zhao, J.; and Wen, B. 2023. Towards Adversarially Robust Continual Learning. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5. IEEE.

Bai, Y.; Yan, X.; Jiang, Y.; Xia, S.-T.; and Wang, Y. 2021. Clustering effect of adversarial robust models. *Advances in Neural Information Processing Systems*, 34: 29590–29601.

Boschini, M.; Bonicelli, L.; Buzzega, P.; Porrello, A.; and Calderara, S. 2022. Class-incremental continual learning into the extended der-verse. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5): 5497–5512.

Buzzega, P.; Boschini, M.; Porrello, A.; Abati, D.; and CALDERARA, S. 2020. Dark Experience for General Continual Learning: a Strong, Simple Baseline. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 15920–15930. Curran Associates, Inc.

Caccia, L.; Aljundi, R.; Asadi, N.; Tuytelaars, T.; Pineau, J.; and Belilovsky, E. 2021. New insights on reducing abrupt representation change in online continual learning. *arXiv preprint arXiv:2104.05025*.

Chen, T.; Zhang, Z.; Liu, S.; Chang, S.; and Wang, Z. 2020. Robust overfitting may be mitigated by properly learned smoothening. In *International Conference on Learning Representations*.

Croce, F.; and Hein, M. 2020. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*, 2206–2216. PMLR.

Cubuk, E. D.; Zoph, B.; Mane, D.; Vasudevan, V.; and Le, Q. V. 2019. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 113–123.

Cubuk, E. D.; Zoph, B.; Shlens, J.; and Le, Q. V. 2020. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 702–703.

Deng, Z.; Zhang, L.; Vodrahalli, K.; Kawaguchi, K.; and Zou, J. Y. 2021. Adversarial training helps transfer learning via better representations. *Advances in Neural Information Processing Systems*, 34: 25179–25191.

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

Farquhar, S.; and Gal, Y. 2018. Towards robust evaluations of continual learning. *arXiv preprint arXiv:1805.09733*.

Goldblum, M.; Fowl, L.; Feizi, S.; and Goldstein, T. 2020. Adversarially Robust Distillation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04): 3996–4003.

Goodfellow, I. J.; Mirza, M.; Xiao, D.; Courville, A.; and Bengio, Y. 2013. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hou, S.; Pan, X.; Loy, C. C.; Wang, Z.; and Lin, D. 2019. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 831–839.

Huang, B.; Chen, M.; Wang, Y.; Lu, J.; Cheng, M.; and Wang, W. 2023. Boosting Accuracy and Robustness of Student Models via Adaptive Adversarial Distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 24668–24677.

Izmailov, P.; Podoprikhin, D.; Garipov, T.; Vetrov, D.; and Wilson, A. G. 2018. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*.

Kireev, K.; Andriushchenko, M.; and Flammarion, N. 2022. On the effectiveness of adversarial training against common corruptions. In *Uncertainty in Artificial Intelligence*, 1012–1021. PMLR.

Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A. A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13): 3521–3526.

Krizhevsky, A. 2009. Learning Multiple Layers of Features from Tiny Images. 32–33.

Le, Y.; and Yang, X. 2015. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7): 3.

Li, L.; and Spratling, M. 2023. Data augmentation alone can improve adversarial training. *arXiv preprint arXiv:2301.09879*.

Li, Z.; and Hoiem, D. 2017. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12): 2935–2947.

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

Maroto, J.; Ortiz-Jiménez, G.; and Frossard, P. 2022. On the benefits of knowledge distillation for adversarial robustness. *CoRR*, abs/2203.07159.

Mermillod, M.; Bugaiska, A.; and Bonin, P. 2013. The stability-plasticity dilemma: Investigating the continuum from catastrophic forgetting to age-limited learning effects.

Moosavi-Dezfooli, S.-M.; Fawzi, A.; Uesato, J.; and Frossard, P. 2019. Robustness via curvature regularization, and vice versa. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9078–9086.

Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; Ng, A. Y.; et al. 2011. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, 4. Granada.

Park, G. Y.; and Lee, S. W. 2021. Reliably fast adversarial training via latent adversarial perturbation. In *Proceedings of the IEEE/CVF international conference on computer vision*, 7758–7767.

Qin, C.; Martens, J.; Gowal, S.; Krishnan, D.; Dvijotham, K.; Fawzi, A.; De, S.; Stanforth, R.; and Kohli, P. 2019. Adversarial robustness through local linearization. *Advances in Neural Information Processing Systems*, 32.

Rebuffi, S.-A.; Gowal, S.; Calian, D. A.; Stimberg, F.; Wiles, O.; and Mann, T. 2021. Fixing data augmentation to improve adversarial robustness. *arXiv preprint arXiv:2103.01946*.

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

Riemer, M.; Cases, I.; Ajemian, R.; Liu, M.; Rish, I.; Tu, Y.; and Tesauro, G. 2018. Learning to learn without forgetting by maximizing transfer and minimizing interference. *arXiv preprint arXiv:1810.11910*.

Salman, H.; Ilyas, A.; Engstrom, L.; Kapoor, A.; and Madry, A. 2020. Do adversarially robust imagenet models transfer better? *Advances in Neural Information Processing Systems*, 33: 3533–3545.

Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; and Chen, L.-C. 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4510–4520.

Santurkar, S.; Tsipras, D.; and Madry, A. 2020. Breeds: Benchmarks for subpopulation shift. *arXiv preprint arXiv:2008.04859*.

Schwarz, J.; Czarnecki, W.; Luketina, J.; Grabska-Barwinska, A.; Teh, Y. W.; Pascanu, R.; and Hadsell, R. 2018. Progress & compress: A scalable framework for continual learning. In *International conference on machine learning*, 4528–4537. PMLR.

Wang, Y.; Ma, X.; Bailey, J.; Yi, J.; Zhou, B.; and Gu, Q. 2021. On the convergence and robustness of adversarial training. *arXiv preprint arXiv:2112.08304*.

Wang, Y.; Zou, D.; Yi, J.; Bailey, J.; Ma, X.; and Gu, Q. 2020. Improving adversarial robustness requires revisiting misclassified examples. In *International Conference on Learning Representations*.

Wu, B.; Chen, J.; Cai, D.; He, X.; and Gu, Q. 2021. Do wider neural networks really help adversarial robustness? *Advances in Neural Information Processing Systems*, 34: 7054–7067.

Wu, D.; Xia, S.-T.; and Wang, Y. 2020. Adversarial weight perturbation helps robust generalization. *Advances in Neural Information Processing Systems*, 33: 2958–2969.

Yue, X.; Mou, N.; Wang, Q.; and Zhao, L. 2024. Revisiting Adversarial Training under Long-Tailed Distributions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 24492–24501.

Zenke, F.; Poole, B.; and Ganguli, S. 2017. Continual learning through synaptic intelligence. In *International conference on machine learning*, 3987–3995. PMLR.

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

Zhao, S.; Yu, J.; Sun, Z.; Zhang, B.; and Wei, X. 2022. Enhanced accuracy and robustness via multi-teacher adversarial distillation. In *European Conference on Computer Vision*, 585–602. Springer.

Zhu, J.; Yao, J.; Han, B.; Zhang, J.; Liu, T.; Niu, G.; Zhou, J.; Xu, J.; and Yang, H. 2021. Reliable adversarial distillation with unreliable teachers. *arXiv preprint arXiv:2106.04928*.

Zi, B.; Zhao, S.; Ma, X.; and Jiang, Y.-G. 2021. Revisiting adversarial robustness distillation: Robust soft labels make student better. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 16443–16452.