Maintaining Fairness in Logit-based Knowledge Distillation for Class-Incremental Learning

Zijian Gao[1,2*], **Shanhao Han**[1*], **Xingxing Zhang**[3], **Kele Xu**[1,2†], **Dulan Zhou**[1,2], **Xinjun Mao**[1,2], **Yong Dou**[1], **Huaimin Wang**[1,2]

[1]College of Computer Science and Technology, National University of Defense Technology, Changsha, China.
[2]State Key Laboratory of Complex & Critical Software Environment, Changsha, China.
[3]School of Computer Science, Tsinghua University, Beijing, China.
{gaozijian19, hanshanhao5205, dulan_zhou, xukelele, xjmao, yongdou, hmwang}@nudt.edu.cn,
xxzhang1993@gmail.com

Abstract

Logit-based knowledge distillation (KD) is commonly used to mitigate catastrophic forgetting in class-incremental learning (CIL) caused by data distribution shifts. However, the strict match of logit values between student and teacher models conflicts with the cross-entropy (CE) loss objective of learning new classes, leading to significant recency bias (i.e. unfairness). To address this issue, we rethink the overlooked limitations of KD-based methods through empirical analysis. Inspired by our findings, we introduce a plug-and-play preprocess method that normalizes the logits of both the student and teacher across all classes, rather than just the old classes, before distillation. This approach allows the student to focus on both old and new classes, capturing intrinsic inter-class relations from the teacher. By doing so, our method avoids the inherent conflict between KD and CE, maintaining fairness between old and new classes. Additionally, recognizing that overconfident teacher predictions can hinder the transfer of inter-class relations (i.e., dark knowledge), we extend our method to capture intra-class relations among different instances, ensuring fairness within old classes. Our method integrates seamlessly with existing logit-based KD approaches, consistently enhancing their performance across multiple CIL benchmarks without incurring additional training costs.

Code — https://github.com/Zi-Jian-Gao/Maintaining-Fairness-in-LKD-for-CIL

Introduction

Class-incremental learning (CIL) (Masana et al. 2022) aims to enable a network to incrementally learn new classes while accurately classifying all previously encountered classes. Unlike conventional training paradigms, CIL requires deep neural networks to incorporate new data without losing historical knowledge, eliminating the need for retraining from scratch—an essential feature for practical applications. The primary challenge in CIL is overcoming catastrophic forgetting (CF) (McCloskey and Cohen 1989), where the model's performance on old classes significantly degrades after learning new ones due to data distribution shifts.

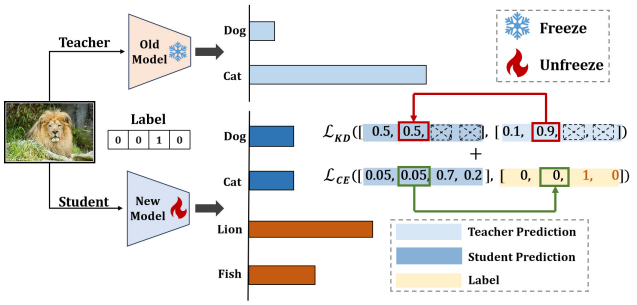

Figure 1: Vanilla KD paradigm in CIL.

Knowledge Distillation (KD) (Hinton et al. 2015) is a way to alleviate forgetting, where the old model (teacher) consolidates knowledge into the new model (student). This approach has been extensively adopted in CIL (Castro et al. 2018; Li and Hoiem 2017; Dhar et al. 2019). As depicted in Figure 1, the vanilla KD-based method (e.g., LwF (Li and Hoiem 2017)) optimizes both the KD loss and the cross-entropy (CE) loss simultaneously. However, KD enforces an exact match between the teacher's and student's logits for old classes, both in terms of value range and variance. For example, if the ground truth label for the second class is 0, but the teacher's prediction probability is 0.9, these optimization goals conflict, making reconciliation difficult. This conflict diminishes the positive impact of KD (Zhao et al. 2020; Liu et al. 2024) and results in recency bias (Masana et al. 2022; Wang et al. 2024), where the model increasingly favors classifying instances into new classes.

To investigate this limitation, we conducted empirical studies on the model's stability in vanilla KD through continual evaluation at every iteration. We observed a significant transient forgetting during task transitions, followed by a recovery phase, as shown in Figure 2 (b), also referred to as the stability gap (Lange, van de Ven, and Tuytelaars 2023). We analyzed this phenomenon in two aspects: temporary forgetting and partial recovery, using gradient-based and empirical analyses. Our findings revealed two critical

*Equal Contribution.
†Corresponding author.

issues: (1) focusing exclusively on old logits during distillation leads to severe forgetting during early transitions, and (2) the exact match requirement in KD loss hampers the recovery of forgotten knowledge in later transitions. As a result, KD-based approaches struggle to effectively learn from new tasks while mitigating forgetting.

Motivated by these insights, we developed a novel preprocess method for KD that effectively maintains fairness between old and new classes. This method replaces the exact match with a semantically invariant inter-class match, allowing the student's logits to vary in range and variance while preserving the semantic relationships of the teacher's old class logits and enabling the student to learn new tasks. Additionally, recognizing that overconfident teacher predictions can hinder the student's ability to capture intrinsic relations among old classes (i.e., dark knowledge), we introduce intra-class relation distillation, which further ensures fairness within old classes, enabling the student to learn equally from each class. In summary, the main contributions include:

- We critically reassess the overlooked sub-optimality of vanilla KD through comprehensive empirical evaluations and analyses, revealing the conflict between learning and anti-forgetting caused by the neglected interplay and rigid exact match-based KD.
- Building on these findings, we propose a plug-and-play pre-process method that employs a relaxed, semantically invariant match to capture intrinsic inter-class relations and ensure fairness between old and new classes.
- To address the issue of overconfident teacher predictions, we introduce an intra-class loss by distilling the teacher's predictions for each old class across multiple instances, ensuring fairness within old classes.
- Extensive experimental results demonstrate that the proposed method consistently enhances the performance of various KD-based CIL approaches across multiple benchmarks and settings, without incurring additional training costs.

Related Work

In this section, we review key related works in KD and CIL and discuss how our research aligns with the advancements.

Knowledge Distillation (KD) is a widely adopted technique for transferring knowledge from large, complex models (teachers) to smaller, more efficient models (students). Initially introduced by Hinton et al. (Hinton et al. 2015), KD aims to transfer the "dark knowledge" embedded in teacher models to students, often leading to superior performance compared to training models directly on the dataset (Du et al. 2020). In recent years, various KD methods have been developed to enhance distillation performance (Tung and Mori 2019; Huang et al. 2022; Chi et al. 2023; Sun et al. 2024).

In the context of CIL, KD is widely used due to its simplicity and effectiveness in mitigating forgetting (Li and Hoiem 2017; Dhar et al. 2019; Wu et al. 2019; Rebuffi et al. 2017; Zhao et al. 2020; Douillard et al. 2020; Zhu et al. 2021c,b). KD-based CIL methods can be broadly categorized into two types: feature-based and logit-based. Feature-based KD is typically used in exemplar-free CIL scenarios to preserve feature distributions and utilize prototypes for pseudo feature replay (Zhu et al. 2021b,a,c). Logit-based KD is commonly applied in regularization-based (Li and Hoiem 2017) and replay-based (Rebuffi et al. 2017; Zhao et al. 2020; Douillard et al. 2020) CIL methods, aligning the output logits of the old and new models.

One major challenge in KD-based CIL is overcoming recency bias (Masana et al. 2022; Wang et al. 2024), where the model increasingly classifies instances into new classes. To address this, the BiC (Wu et al. 2019) introduces a bias correction stage after distillation, adjusting the bias in the fully connected layer using a validation set. Alternatively, WA (Zhao et al. 2020) employs a weight aligning strategy to correct biased weights post-training without requiring additional training. Another approach by Liu et al. (Liu et al. 2024) uses placebos from an unlabeled image stream instead of new samples to distill and preserve old knowledge, thus avoiding the conflict between CE loss and KD loss.

Despite the progress in logit-based distillation, the underlying mechanism by which KD contributes to recency bias and strategies to mitigate this bias have not been fully explored. This paper addresses this gap by investigating the fundamental mechanism of KD and introducing a novel preprocess to reduce recency bias and maintain fairness between old and new classes in KD-based CIL methods.

Preliminary Analysis

Background and Notation

We start with the key notations of the KD-based baseline (i.e., LwF). In an incremental task with O old classes and N new classes, the goal is to train a model to classify across all $O+N$ classes. Let $\{(x_1, y_1), \ldots, (x_M, y_M)\}$ denotes the samples from the new classes, where M is the number of samples, and x_i and y_i are the input data and target label, respectively. During incremental training, the student model f_S learns from the new samples using classification loss L_{CE} while maintaining stability with KD loss L_{KD} from the teacher model f_T, trained in the previous tasks.

Given an input (x, y), let $\hat{Z}(x) = (\hat{z}_1, \ldots, \hat{z}_O)$ and $Z(x) = (z_1, \ldots, z_O, z_{O+1}, \ldots, z_{O+N})$ represent the output logits of the teacher and student models, respectively. The cross-entropy (CE) loss for learning is defined as:

$$\mathcal{L}_{\mathrm{CE}}(x, y) = -\sum_{i=O+1}^{O+N} \delta_{i=y} \log\left(p_i(x)\right), \tag{1}$$

where $\delta_{y=j}$ is an indicator function for the true label y_i, and $p_i(x)$ is the student prediction probability for the correct class. Meanwhile, the softmax function converts the old class logits to the probabilities $\hat{q}(x)$ and $q(x)$ for distillation, and the i-th item is calculated by

$$\hat{q}_i(x) = \frac{e^{\hat{z}_i/\tau}}{\sum_{j=1}^{O} e^{\hat{z}_j/\tau}}, \quad q_i(x) = \frac{e^{z_i/\tau}}{\sum_{j=1}^{O} e^{z_j/\tau}},$$

where τ is the temperature scalar. It is worth noting that, in CIL, cross-entropy and Kullback-Leibler (KL) Divergence

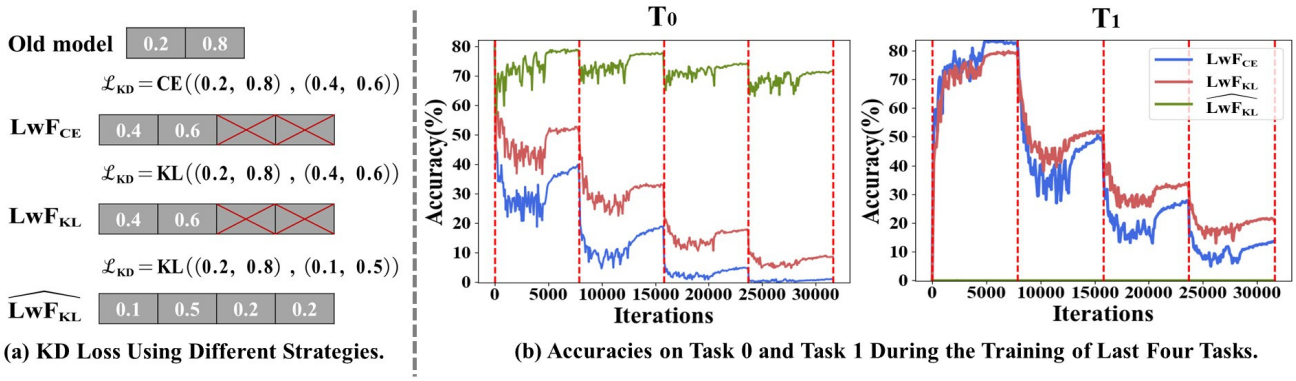

Figure 2: Detailed performance analysis with different KD mechanisms.

are both widely used for calculating the KD loss $\mathcal{L}_{\text{KD}}$. The CE-based KD loss is given by

$$\mathcal{L}_{\text{CE}}\left(\hat{q}(x), q(x)\right) = -\sum_{i=1}^{O} \hat{q}_i(x) \log\left(q_i(x)\right). \tag{2}$$

And, the KL-based KD loss is given by

$$\mathcal{L}_{\text{KL}}\left(\hat{q}(x) \| q(x)\right) = \tau^2 \sum_{i=1}^{O} \hat{q}_i(x) \log\left(\frac{\hat{q}_i(x)}{q_i(x)}\right). \tag{3}$$

It is important to note that they are nonequivalent due to the negative entropy term of teacher prediction $\hat{q}(x)$. Finally, the overall loss, combining the cross-entropy loss and the KD loss, is defined as

$$\mathcal{L} = \mathcal{L}_{\text{CE}}(x, y) + \mathcal{L}_{\text{KD}}(\hat{Z}(x), Z(x)), \tag{4}$$

KD-based Overlooked Sub-optimality

To explore the sub-optimality in KD-based CIL, we conducted comprehensive experiments on LwF using both CE-based and KL-based distillation, referred to as LwF_{CE} and LwF_{KL}, respectively. These experiments were performed under the standard incremental setting—training from scratch—using the CIFAR-100 dataset (Krizhevsky and Hinton 2009). Specifically, we equally divided the classes into five tasks and continuously evaluated the model's accuracy on tasks T_0 and T_1 at each iteration.

In Figure 2 (b), we present the accuracy curves for the model on the first two tasks T_0 and T_1across the training of the subsequent four tasks of a total of five tasks. For clarity, we have omitted the accuracy curve during the initial task T_0 training as the three baseline models exhibit identical performance. The vertical bars mark the transition between tasks. It's obvious that both LwF_{CE} (blue line) and LwF_{KL} (red line) exhibit significant *temporary forgetting* during task transitions, followed by *partial recovery*. This phenomenon, known as the stability gap (Lange, van de Ven, and Tuytelaars 2023), highlights the transient forgetting of old knowledge when learning new tasks.

Notably, LwF_{KL} outperforms LwF_{CE} on task T_0 in both aspects. This advantage is due to KL loss's ability to measure the divergence between the softened probability distributions of the teacher and student models, ensuring the transfer of fine-grained information. In contrast, CE loss focuses solely on matching class probabilities, potentially overlooking the rich information in the relative probabilities of other classes. This limitation makes CE loss less effective at capturing nuanced differences between the teacher's and student's outputs, leading to more severe forgetting and hindering knowledge recovery.

Gradient Analysis. However, the maximum forgetting in LwF_{KL} for task T_0 remains significant, with accuracy dropping from around 80 to 40 when learning task T_1. This inevitable forgetting, driven by gradients, is a common issue across parameter regularization-based CIL methods (Lange, van de Ven, and Tuytelaars 2023). From a gradient perspective, during initial updates on a new task, the parameters θ of the student model f_S (comprising the feature extractor and fully connected layer for old classes) are still closely aligned with the teacher model's parameters θ^* from the previous task, leading to $|\nabla\mathcal{L}_{\text{KD}}| \approx 0$. This motivates the development of a novel mechanism to mitigate the forgetting.

It is clear that when the expanded fully connected layer for new classes is included, the student's parameters $\hat{\theta}$ will differ significantly from the teacher's due to the additional neurons required for classifying the new classes. Specifically, as shown in Figure 2 (a), we modified the student model's probability calculation procedure to incorporate logits for the new classes:

$$q_i(x) = \frac{e^{z_i/\tau}}{\sum_{j=1}^{O+N} e^{z_j/\tau}} \tag{5}$$

This led to the development of a new distillation mechanism, referred to as $\widehat{LwF_{KL}}$ (green line). As shown in Figure 2 (b), this mechanism significantly reduces temporary forgetting, allowing almost all old knowledge from T_0 to be recovered in $\widehat{LwF_{KL}}$. Based on this finding, we can reasonably hypothesize:

Hypothesis 1. *Without the interplay between old and new classes, the KD mechanism fails to effectively mitigate forgetting and recover knowledge.*

Confidence	$> 20\%$	$> 30\%$	$> 40\%$	$> 50\%$	$> 60\%$	$> 70\%$	$> 80\%$	$> 90\%$
Proportion	$> 100\%$	$> 96.4\%$	$> 88.6\%$	$> 78.4\%$	$> 67.6\%$	$> 57.6\%$	$> 47.6\%$	$> 35.6\%$

Table 1: Overconfident teacher predictions on task T_1. The numbers indicate the proportions of new samples where the maximum classification probability exceeds the corresponding value.

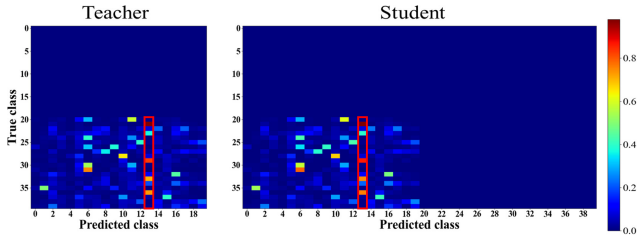

Figure 3: Highly consistent confusion matrices of the teacher and student in $\widehat{LwF_{KL}}$ after learning task T_1.

The Impact of Enforcing An Exact Match. Surprisingly, as shown in Figure 2 (b), the accuracy on the new task T_1 in $\widehat{LwF_{KL}}$ remains at 0. To explore the root causes, Figure 3 compares the confusion matrices for task T_1 using the student model after learning with that of the teacher model, revealing that the student model exhibits extremely high consistency with the teacher's classifications. Meanwhile, Table 1 shows the teacher model's maximum classification probabilities of new samples and the proportion of samples exceeding various probability thresholds.

These observations suggest that the student's prediction probabilities for old classes mirror those of the teacher (i.e., $\forall i \in [1, O], \hat{q}_i(x) = q_i(x)$), resulting in a minimized $\mathcal{L}_{\text{KD}}$. This implies that the student and teacher models share the same mean and variance of logits, reflecting that the current $\mathcal{L}_{\text{KD}}$ enforces an exact match in both value range and variance (Sun et al. 2024). This strict alignment hampers the student's ability to learn new knowledge in $\widehat{LwF_{KL}}$ and limits the recovery of old knowledge in LwF_{KL} and LwF_{CE}. Therefore, combined with Hypothesis 1, we propose the following hypothesis:

Hypothesis 2. *An ideal KD mechanism in CIL should balance learning and forgetting by considering the interplay between old and new classes and allow a more relaxed match between teacher and student predictions, thus maintaining fairness between old and new classes.*

Theoretical Foundation and Our Approach

Motivated by Hypothesis 1, we reconsidered what truly matters in the teacher's output for CIL. During practical implementation and inference, the key factor is the inter-class semantic relations between logit values, as these ultimately determine the final prediction results. Rather than enforcing an exact match of logit values, we prioritize maintaining the correct order of predictions

Traditionally, the distance metric $d(\cdot, \cdot)$ between the teacher's logits $\hat{Z}(x)$ and the student's logits $Z(x)$ is minimized to zero in vanilla KD, ensuring an exact match. However, instead of this strict alignment, we focus on preserving the relational structure of the logits to maintain semantic integrity and accurate inference. A monotonic positive linear transformation offers a straightforward yet effective mapping, remaining invariant under separate changes in scale and shift for the logits. Our goal is to identify a transformation that upholds this property. To achieve this, we introduce the widely used $\mathcal{Z}$-score normalization $\mathcal{Z}(\cdot)$ (Sahu 2015; Singh and Singh 2020; Sun et al. 2024) as a ***monotonic positive*** linear transformation function to provide isotonic mapping:

$$
\begin{aligned}
d(\mathcal{Z}(\hat{Z}(x)), \mathcal{Z}(Z(x))) &= d\left(\frac{\hat{Z}(x) - \mu_t}{\sigma_t}, \frac{Z(x) - \mu_s}{\sigma_s}\right) \\
&= d\left(\frac{1}{\sigma_t}\hat{Z}(x) - \frac{\mu_t}{\sigma_t}, \frac{1}{\sigma_s}\hat{Z}(x) - \frac{\mu_s}{\sigma_s}\right)
\end{aligned}
$$

where the scaling factors σ_t and σ_s represent the standard deviations of the data sets $\hat{Z}(x)$ and $Z(x)$, and the shifting factors μ_t and μ_s represent their respective means. $\mathcal{Z}$-score normalization intrinsically guarantees that the normalized logits have ***a mean of zero*** and ***a standard deviation of 1*** by subtracting the data set mean from each data point and dividing by the data set's standard deviation.

These properties ensure that $\mathcal{Z}$-score normalization is a monotonic positive linear transformation, invariant under separate changes in scale and shift, thus preserving isotonic semantic information.

Maintain Fairness between Old and New Classes

Based on Hypothesis 2 and the properties of $\mathcal{Z}$-score normalization, we propose a plug-and-play pre-process method that employs a relaxed, semantically invariant match to capture intrinsic inter-class relations and ensures fairness between old and new classes. The $\mathcal{Z}$-score normalization-based inter-class distillation loss is formulated as:

$$
\begin{aligned}
&\mathcal{L}_{\text{inter}} = \mathcal{L}_{\text{KL}}\left(\hat{q}(x) \| q(x)\right) = \tau^2 \sum_{i=1}^{O} \hat{q}_i(x) \log\left(\frac{\hat{q}_i(x)}{q_i(x)}\right), \\
&\hat{q}_i(x) = \frac{e^{\mathcal{Z}(\hat{Z})_i/\tau}}{\sum_{j=1}^{O} e^{\mathcal{Z}(\hat{Z})_j/\tau}}, \quad q_i(x) = \frac{e^{\mathcal{Z}(Z)_i/\tau}}{\sum_{j=1}^{O+N} e^{\mathcal{Z}(Z)_j/\tau}}.
\end{aligned} \tag{6}
$$

A Toy Example. Figure 5 presents a toy example comparing the vanilla logit-based KD mechanism in CIL with our improved KD mechanism incorporating the pre-process method. In this example, the old class logits of student S_1

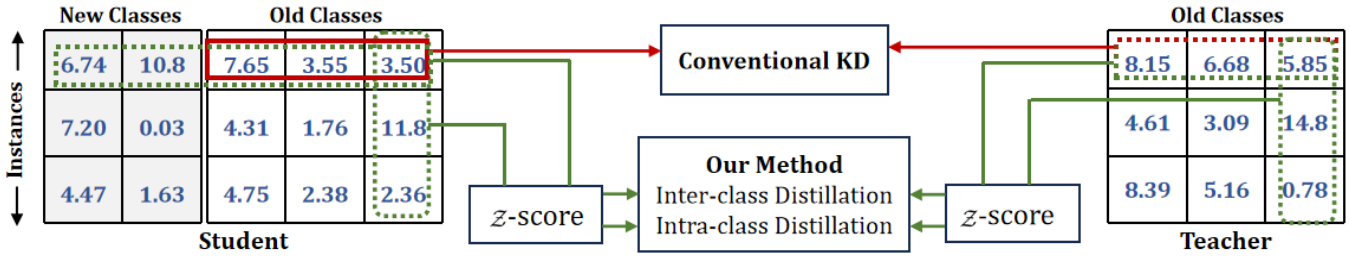

Figure 4: Schematic diagram of our method.

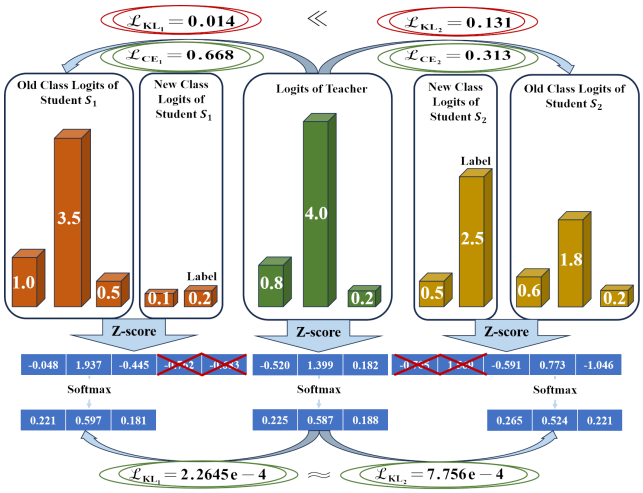

Figure 5: A Toy Example. Our $\mathcal{Z}$-score normalization pre-process method ensures that the KD loss and CE loss no longer conflict, allowing both to be optimized simultaneously for balanced learning and anti-forgetting.

are closer to the teacher's in terms of value range and variance, while student S_2 preserves the same semantic relationships as the teacher. In vanilla KD, this exact match results in student S_1 achieving a much lower KD loss (i.e., $\mathcal{L}_{\mathrm{KL}_1} = 0.014$) compared to student S_2 (i.e., $\mathcal{L}_{\mathrm{KL}_2} = 0.131$). However, the lower KD loss for S_1 leads to a higher cross-entropy loss $\mathcal{L}_{\mathrm{CE}_1}$ (0.668), indicating a limited ability to learn new classes compared to S_2 ($\mathcal{L}_{\mathrm{CE}_2} = 0.313$). This exact match enforced by vanilla KD creates a conflict between learning and anti-forgetting, as depicted in Figure 1, contributing to recency bias and leading to a diminished capacity for acquiring new knowledge in $\widehat{LwF_{KL}}$.

In contrast, our proposed $\mathcal{Z}$-score normalization pre-process rescales the logits while preserving their semantic relationships. With this semantically invariant match, student S_2 achieves a KD loss that is very close to zero and correctly predicts new samples, resolving the learning limitations observed in $\widehat{LwF_{KL}}$. Through the toy example, we vividly demonstrate how our pre-process method maintains fairness between old and new classes, ensuring balanced learning and anti-forgetting.

Algorithm 1: Pseudo code of our method in a PyTorch-like style.

```
def Z-score (logits):
    mean = logits.mean(dim=1, keepdims=True)
    stdv = logits.std(dim=1, keepdims=True)
    return (logits - mean) / (1e-7 + stdv)

# O: Number of Old Classes
# N: Number of New Classes
# k: Batch Size
# Z_s: Student Output Logits (shape: [k, O + N])
# Ẑ_t: Teacher Output Logits (shape: [k, O])
# τ: Temperature Scalar
# α, β: Hyperparameters

# Calculate the Inter-Class Distillation Loss:
q̂_t = F.softmax(Z-score(Ẑ_t) / τ)
q_s = F.softmax(Z-score(Z_s)[:,:O] / τ)
kld = F.kl_div(log(q_s), q̂_t)
L_inter = (kld.sum(1, keepdim=True)) * τ^2).mean()

# Calculate the Intra-Class Distillation Loss:
q̂_t = F.softmax(Z-score(Ẑ_t.t()) / τ)
q_s = F.softmax(Z-score(Z_s.t())[:O,:] / τ)
kld = F.kl_div(log(q_s), q̂_t)
L_intra = (kld.sum(1, keepdim=True)) * τ^2).mean()

# Calculate the Total Distillation Loss:
L_KD = αL_inter + βL_intra
```

Maintain Fairness within Old Classes

Another benefit of our pre-process method is that after applying $\mathcal{Z}$-score normalization, the implicit information within the teacher's output probabilities—known as dark knowledge—is more effectively transferred. As illustrated in Table 1, the teacher model exhibits overconfidence in new samples. For example, after applying the softmax function to the teacher's logits, it classified 35.6% of the new class samples as belonging to an old class with over 90% probability (close to the one-hot prediction).

This overconfidence can hinder the effective transfer of dark knowledge (Chi et al. 2023; Gao et al. 2024). For instance, as shown in Figure 5, the teacher's prediction probabilities after applying the softmax func-

Dataset	CIFAR-100								ImageNet-Subset							
Method	Split 5 Tasks		Split 10 Tasks		Half 6 Tasks		Half 11 Tasks		Split 5 Tasks		Split 10 Tasks		Half 6 Tasks		Half 11 Tasks	
	FAA	CAA	FAA	CAA	FAA	CAA	FAA	CAA	FAA	CAA	FAA	CAA	FAA	CAA	FAA	CAA
LwF	37.55	55.27	21.70	43.01	21.60	34.22	15.62	27.88	34.63	56.08	20.08	42.76	21.06	38.65	11.27	27.85
$w/\mathcal{L}_{\text{inter}}$	38.70	59.17	19.64	40.85	21.77	38.22	20.10	44.39	28.47	53.21	17.15	40.68	18.81	35.47	11.87	30.74
$w/\mathcal{L}_{\text{inter}} + \mathcal{L}_{\text{intra}}$	**46.09**	**62.33**	**22.81**	**46.66**	**26.36**	**48.08**	**22.80**	**47.33**	**34.96**	**59.14**	**19.17**	**44.60**	**21.46**	**42.39**	**14.58**	**38.17**
Replay	48.49	64.05	43.76	61.30	46.80	57.14	44.78	54.54	51.42	66.75	44.88	62.91	51.49	62.95	50.45	59.33
$w/\mathcal{L}_{\text{inter}}$	52.65	66.64	46.77	62.49	51.50	**62.20**	46.26	57.31	52.73	67.97	45.22	63.07	53.76	65.47	**52.21**	61.12
$w/\mathcal{L}_{\text{inter}} + \mathcal{L}_{\text{intra}}$	**53.70**	**66.94**	**47.45**	**62.51**	**51.65**	62.03	**47.59**	**57.46**	**54.57**	**68.73**	**46.14**	**63.72**	**54.88**	**65.96**	51.71	**62.01**
iCaRL	47.69	64.31	41.98	60.97	46.63	58.95	43.44	54.74	51.84	67.94	43.29	62.58	51.10	63.61	49.26	58.72
$w/\mathcal{L}_{\text{inter}}$	**52.58**	66.12	**48.38**	63.58	**52.85**	62.70	46.42	57.00	53.17	68.25	44.39	**62.76**	52.96	64.74	49.48	59.96
$w/\mathcal{L}_{\text{inter}} + \mathcal{L}_{\text{intra}}$	52.50	**66.47**	48.08	**63.76**	52.78	**63.40**	**46.77**	**57.43**	**53.86**	**68.77**	**44.39**	62.68	**53.08**	**65.35**	**50.39**	**60.62**
BiC	48.21	65.46	40.22	61.99	45.81	63.30	37.37	53.87	59.83	68.18	41.12	61.53	54.52	68.89	45.62	59.92
$w/\mathcal{L}_{\text{inter}}$	**58.59**	**67.60**	**51.48**	**64.52**	58.18	68.42	42.65	60.53	59.29	68.05	**48.92**	**63.65**	63.41	72.99	46.38	63.72
$w/\mathcal{L}_{\text{inter}} + \mathcal{L}_{\text{intra}}$	58.27	67.53	48.15	63.50	**59.56**	**68.88**	**45.29**	**61.87**	**59.37**	**68.20**	48.39	63.35	**64.42**	**73.67**	**47.89**	**65.74**
WA	53.86	64.97	48.23	62.21	55.60	65.22	51.98	62.72	51.73	**64.72**	45.67	60.05	53.87	66.73	50.38	62.78
$w/\mathcal{L}_{\text{inter}}$	**57.05**	65.40	52.55	60.40	54.85	67.15	53.73	66.15	**52.53**	63.28	45.14	58.74	56.59	68.13	49.73	63.40
$w/\mathcal{L}_{\text{inter}} + \mathcal{L}_{\text{intra}}$	56.83	**65.53**	**53.03**	**64.72**	**58.18**	**68.80**	**56.03**	**66.55**	50.23	63.12	**45.91**	**61.20**	**57.78**	**68.72**	**50.87**	**65.50**
PODNet	49.08	62.96	36.78	55.22	59.03	68.99	55.62	66.36	55.85	70.06	42.83	61.25	68.08	76.38	65.40	74.94
$w/\mathcal{L}_{\text{inter}}$	50.20	63.71	**40.44**	56.91	**61.94**	**70.57**	**57.11**	**67.60**	57.39	71.05	44.14	62.70	**70.20**	**77.75**	66.53	75.52
$w/\mathcal{L}_{\text{inter}} + \mathcal{L}_{\text{intra}}$	**50.99**	**64.18**	40.35	**57.26**	61.77	70.19	55.36	66.60	**58.59**	**71.46**	**44.31**	**63.05**	69.45	77.49	**66.64**	**75.68**
Average Improvement	*5.58*	*2.66*	*4.53*	*2.28*	*5.81*	*5.59*	*4.17*	*6.19*	*1.05*	*0.95*	*1.74*	*1.25*	*3.49*	*2.73*	*1.62*	*4.03*

Table 2: Standard KD techniques with and without our $\mathcal{L}_{\text{inter}}$ and $\mathcal{L}_{\text{intra}}$ on different settings of CIFAR-100 and ImageNet-Subset datasets. **Best results - in bold.**

tion are $[0.149, 0.740, 0.111]$. In contrast, after applying our pre-process method, these probabilities shift to $[0.225, 0.587, 0.188]$. This adjustment mitigates the negative effects of overconfidence, enabling a more effective transfer of dark knowledge and further reducing forgetting.

However, as shown in the confusion matrix in Figure 3, the teacher classified a significant portion of new class samples as a specific old class (highlighted in the red box), potentially overemphasizing this old class while neglecting others, leading to unfairness within old classes.

To address this, we propose intra-class relation distillation to further maintain fairness within old classes. This approach captures the semantic similarities of multiple instances to each old class (Huang et al. 2022). Given a specific old class o and k instances in each training batch, let $\hat{C}(o) = (\hat{c}_1, \ldots, \hat{c}_k)$ and $C(o) = (c_1, \ldots, c_k)$ represent the logits of old class o from the teacher and student models across k instances, respectively. Similar to Equation 6, the $\mathcal{Z}$-score normalization-based intra-class distillation loss can be formulated as:

$$\mathcal{L}_{\text{intra}} = \mathcal{L}_{\text{KL}}\left(\hat{q}(o) \| q(o)\right) = \tau^2 \sum_{i=1}^{k} \hat{q}_i(o) \log\left(\frac{\hat{q}_i(o)}{q_i(o)}\right), \quad \hat{q}_i(o) = \frac{e^{\mathcal{Z}(\hat{C})_i/\tau}}{\sum_{j=1}^{k} e^{\mathcal{Z}(\hat{C})_j/\tau}}, \quad q_i(o) = \frac{e^{\mathcal{Z}(C)_i/\tau}}{\sum_{j=1}^{k} e^{\mathcal{Z}(C)_j/\tau}}. \tag{7}$$

Finally, the overall loss can be defined as:

$$\mathcal{L} = \mathcal{L}_{\text{CE}} + \alpha\mathcal{L}_{\text{inter}} + \beta\mathcal{L}_{\text{intra}} \tag{8}$$

where α and β are coefficients that control the weight of inter-class and intra-class losses. Figure 4 illustrates the schematic diagram of our method and Algorithm 1 provides the pseudo-code of our method in a PyTorch-like (Paszke et al. 2019).

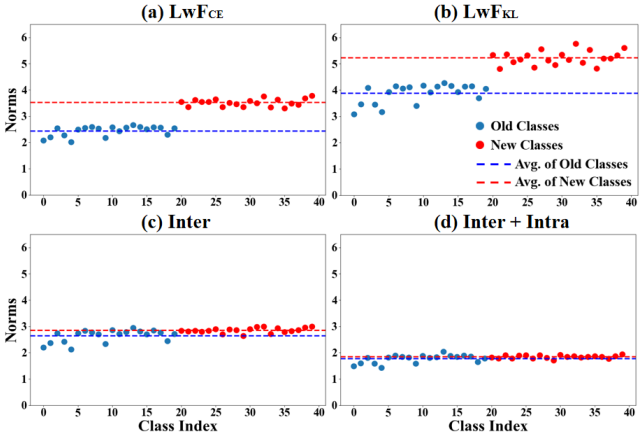

Figure 6: Norms of the classifier's weight vectors.

Visualized Analysis on Recency Bias. To visualize the recency bias of different KD mechanisms, we calculated the norms of the classifier weight vectors after learning the new task T_1 and plotted them in Figure 6. If the norms of the weight vectors for new classes are larger, the output logits for new classes may generally tend to be larger (Zhao et al. 2020). As shown in Figures 6 (a) and (b), the norms of the weight vectors for new classes are significantly larger than those for old classes, indicating severe recency bias.

After applying our inter-class loss, as shown in Figure 6

Method	Split 5 Tasks		Split 10 Tasks		Half 6 Tasks		Half 11 Tasks	
	FAA	CAA	FAA	CAA	FAA	CAA	FAA	CAA
LwF	23.05	34.83	14.87	28.04	14.64	22.14	8.51	13.85
$w/\mathcal{L}_{\text{inter}} + \mathcal{L}_{\text{intra}}$	**28.20**	**40.04**	**15.77**	**31.36**	**20.92**	**34.17**	**9.97**	**18.76**
Replay	18.80	34.93	17.53	26.60	9.95	20.79	7.39	15.57
$w/\mathcal{L}_{\text{inter}} + \mathcal{L}_{\text{intra}}$	**23.94**	**38.87**	**16.60**	**31.81**	**17.88**	**28.11**	**10.79**	**19.68**
iCaRL	29.93	38.50	21.75	34.90	25.60	39.15	15.47	30.29
$w/\mathcal{L}_{\text{inter}} + \mathcal{L}_{\text{intra}}$	**33.96**	**42.30**	20.96	**36.17**	**28.47**	**42.32**	**20.96**	**36.68**
BiC	15.72	31.32	8.17	23.95	8.65	20.26	6.24	14.71
$w/\mathcal{L}_{\text{inter}} + \mathcal{L}_{\text{intra}}$	**23.38**	**38.93**	**14.04**	**31.00**	**16.41**	**27.10**	**10.65**	**19.53**
WA	28.75	41.11	14.51	31.28	15.46	32.62	11.79	27.34
$w/\mathcal{L}_{\text{inter}} + \mathcal{L}_{\text{intra}}$	**33.78**	**42.14**	**23.28**	**36.13**	**26.60**	**38.77**	**18.15**	**33.02**
PODNet	18.34	31.38	10.64	24.31	23.98	39.37	16.77	31.16
$w/\mathcal{L}_{\text{inter}} + \mathcal{L}_{\text{intra}}$	**20.92**	**32.52**	**11.51**	**25.81**	**25.97**	**40.19**	**17.97**	**31.49**
Average Improvement	***4.93***	***3.79***	***2.45***	***3.87***	***6.33***	***6.06***	***3.72***	***4.37***

Table 3: Standard KD techniques with and without our method on different settings of the TinyImageNet dataset.

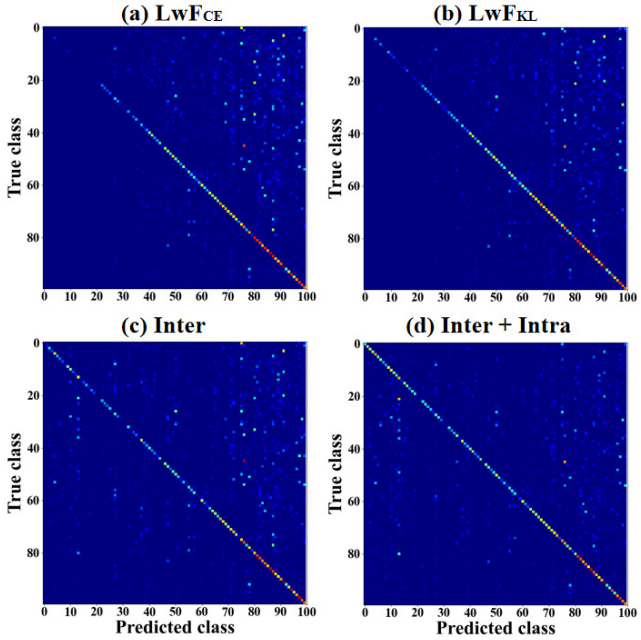

Figure 7: Confusion matrices of different KD mechanisms with five incremental phases on CIFAR-100.

(c), the recency bias is significantly reduced, with the average norm values of the old and new classes becoming much closer (indicated by the blue and red dotted lines). Furthermore, in Figure 6 (d), when both our inter-class and intra-class losses are applied, the dispersion of blue and red points decreases noticeably, leading to nearly overlapping dotted lines. These results vividly demonstrate the validity of our empirical analysis and theoretical foundations, confirming that our method effectively maintains fairness not only between old and new classes but also within old classes.

Experimental Results

Experimental Settings. To evaluate the effectiveness of our method, we conducted experiments in both 'train from scratch' and 'train from half' scenarios on three widely used benchmarks: CIFAR-100 (Krizhevsky and Hinton 2009), ImageNet-Subset (Hou et al. 2019), and TinyImageNet (Le and Yang 2015). In the 'train from scratch' scenario, the model is trained on an equal number of classes in each incremental task, while in the 'train from half' scenario, the model is trained on half the number of all classes in the first task and an equal number of classes in each subsequent task. Our implementation, based on PyTorch (Paszke et al. 2019) and PYCIL (Zhou et al. 2023), ran on an NVIDIA 4090 using ResNet-18 (He et al. 2016) as the model architecture. The models were trained with a batch size of 128 using SGD with momentum.

Following standard CIL practices, we shuffled the class order with a random seed of 1993 (Rebuffi et al. 2017; Zhou et al. 2023). For evaluation, we employed two commonly used metrics in CIL: Final Average Accuracy (FAA) and Cumulative Average Accuracy (CAA) (Wang et al. 2024). We define the average accuracy on all seen tasks as A_t after learning the task t. Upon completing all n tasks, we get the FAA = A_n, and CAA is calculated by $\frac{1}{n}\sum_{i=0}^{n-1} A_i$. FAA highlights performance gaps between incremental learning and joint learning methods, and CAA reflects overall historical performance. We applied our method on various logit distillation-based CIL methods (i.e, LwF (Li and Hoiem 2017), Replay, iCaRL (Rebuffi et al. 2017), BiC (Wu et al. 2019), WA (Zhao et al. 2020)). Especially, in PODNet (Douillard et al. 2020), we replace the feature-based distillation with logit-based distillation.

Plug-and-Play Performance. Tables 2 presents a comprehensive plug-and-play analysis of various KD-based CIL methods on the CIFAR-100 and ImageNet-Subset datasets.

For the baseline methods, the KD weight is set to 1. When using only $\mathcal{L}_{\text{inter}}$, the hyperparameters α and β are set to 1 and 0, respectively. To ensure a fair comparison without affecting learning, when both $\mathcal{L}_{\text{inter}}$ and $\mathcal{L}_{\text{intra}}$ are implemented, α and β are each set to $\frac{1}{2}$.

As shown in the table, applying our inter-class loss significantly enhances FAA and CAA across various scenarios and task settings. Notably, our pre-process for calculating the inter-class loss incurs no additional training costs. Further, when combined with the intra-class loss without altering the learning, overall performance is further enhanced, underscoring the importance of intra-class relation distillation. Average improvements demonstrate the effectiveness of our method, with increases of 5.81% in FAA and 5.59% in CAA in the Half 6 Tasks setting on the CIFAR-100 dataset. Additionally, as shown in Table 3, significant improvements were also observed when our method was tested on the TinyImageNet dataset. These consistent improvements across various methods, settings, and datasets underscore the universality and robustness of our approach.

Visual Comparison. To provide a visual comparison, we present the confusion matrices for LwF_{CE}, LwF_{KL}, LwF_{KL} with $\mathcal{L}_{\text{inter}}$, and LwF_{KL} with $\mathcal{L}_{\text{inter}} + \mathcal{L}_{\text{intra}}$ in the split 5 tasks setting on CIFAR-100. As illustrated in Figure 7, both LwF_{CE} and LwF_{KL} exhibit a clear classification bias, favoring new classes. When the inter-class loss $\mathcal{L}_{\text{inter}}$ is applied, this bias is already noticeably reduced. With the addition of the intra-class loss $\mathcal{L}_{\text{intra}}$, the confusion and bias are further minimized, demonstrating that our method effectively achieves the goal of maintaining fairness between old and new classes.

Ablation Study on Hyperparameters. In Equation 8, the hyperparameters α and β control the weights of $\mathcal{L}_{\text{inter}}$ and $\mathcal{L}_{\text{intra}}$. We conducted experiments on CIFAR-100 based on the LwF method to explore the effects of varying these hyperparameters across different incremental settings. As detailed in Table 4, we tested various values of α and β to systematically evaluate their impact on performance.

The results reveal that: 1) Without $\mathcal{L}_{\text{inter}}$ ($\alpha = 0$), performance declines sharply, approaching the level of finetuning without anti-forgetting measures, underscoring the importance of $\mathcal{L}_{\text{inter}}$. 2) When $\alpha > \beta$, overall performance improves, suggesting that prioritizing inter-class loss yields better outcomes. 3) All configurations with $\alpha \neq 0$ outperform the baseline LwF, demonstrating the robustness of our method across different hyperparameter settings. We recommend setting the cross-entropy loss weight near 1, with an inter-class to intra-class loss ratio around 3:1, generally achieving optimal or near-optimal performance.

Ablation Study on Batch Size. The training batch size k determines the dimension of the model's predictions for each old class, impacting intra-class distillation performance. In Table 5, we conducted experiments on CIFAR-100 based on the LwF method across different incremental settings, with both $\mathcal{L}_{\text{inter}}$ and $\mathcal{L}_{\text{intra}}$ set to 1. The results show that increasing batch size enhances the model's ability to mitigate forgetting, as it allows the intra-class relation to capture more instance relationships. Notably, when the batch size reaches 256, the performance improvement in the train-from-half scenario is particularly impressive, highlighting the importance of the intra-class distillation and the benefits of larger batch sizes.

Value		Split 5 Tasks		Split 10 Tasks		Half 6 Tasks		Half 11 Tasks	
α	β	FMM	CAA	FMM	CAA	FMM	CAA	FMM	CAA
0	2	17.03	39.55	9.02	28.33	9.84	26.02	6.68	21.93
2	0	**47.51**	62.27	23.34	**47.21**	27.66	47.64	**25.44**	50.05
1	1	46.09	**62.33**	22.81	46.66	26.36	48.08	22.80	47.33
2/3	4/3	43.21	60.87	21.58	45.05	25.34	46.55	20.09	43.96
4/3	2/3	46.74	61.90	23.33	46.20	27.88	49.44	23.26	48.70
1/2	3/2	38.70	59.34	20.40	43.88	24.07	44.38	19.09	41.73
3/2	1/2	47.02	62.10	**24.04**	45.96	**29.10**	**50.47**	25.31	**50.07**

Table 4: Performance across different hyperparameter ratios.

Batch Size	Split 5 Tasks		Split 10 Tasks		Half 6 Tasks		Half 11 Tasks	
	FMM	CAA	FMM	CAA	FMM	CAA	FMM	CAA
32	35.66	57.55	22.52	48.36	25.95	41.14	12.85	26.31
64	40.43	60.37	24.37	48.76	29.28	48.35	11.50	25.14
128	**46.09**	**62.33**	22.81	46.66	26.36	48.08	22.40	47.34
256	44.20	60.11	**30.69**	**49.41**	**51.71**	**64.77**	**30.18**	**54.20**

Table 5: Performance across various batch sizes.

Conclusion

In this work, we reevaluated the overlooked sub-optimality of logit-based KD in CIL, particularly its conflict with cross-entropy loss, which exacerbates recency bias. We identified the rigid exact match of logits between student and teacher models as a key factor in this issue. To address this, we introduced a novel pre-process method that normalizes logits across all classes before distillation. This approach preserves inter-class relations and mitigates the conflict between KD and cross-entropy loss, ensuring fairness between old and new classes without incurring additional training costs. Additionally, we addressed the overconfidence in teacher predictions, which hampers the transfer of dark knowledge, by incorporating intra-class relation distillation. This ensures fairness within old classes and further reduces the risk of forgetting. Extensive experiments across multiple CIL benchmarks confirm that our method consistently enhances the performance of existing KD-based approaches, demonstrating its robustness and broad applicability.

Acknowledgements

This work was supported by the National Science and Technology Major Project (2023ZD0121101), and National Natural Science Foundation of China (No.62172426, 62106123).

References

Castro, F. M.; Marín-Jiménez, M. J.; Guil, N.; Schmid, C.; and Alahari, K. 2018. End-to-end incremental learning. In

Proceedings of the European conference on computer vision, 233–248.

Chi, Z.; Zheng, T.; Li, H.; Yang, Z.; Wu, B.; Lin, B.; and Cai, D. 2023. Normkd: Normalized logits for knowledge distillation. *arXiv preprint arXiv:2308.00520*.

Dhar, P.; Singh, R. V.; Peng, K.-C.; Wu, Z.; and Chellappa, R. 2019. Learning without memorizing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 5138–5146.

Douillard, A.; Cord, M.; Ollion, C.; Robert, T.; and Valle, E. 2020. PODNet: Pooled Outputs Distillation for Small-Tasks Incremental Learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 86–102.

Du, S.; You, S.; Li, X.; Wu, J.; Wang, F.; Qian, C.; and Zhang, C. 2020. Agree to disagree: Adaptive ensemble knowledge distillation in gradient space. *advances in neural information processing systems*, 33: 12345–12355.

Gao, Z.; Xu, K.; Zhuang, H.; Liu, L.; Mao, X.; Ding, B.; Feng, D.; and Wang, H. 2024. Less confidence, less forgetting: Learning with a humbler teacher in exemplar-free Class-Incremental learning. *Neural Networks*, 179: 106513.

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.

Hinton, G.; Vinyals, O.; Dean, J.; et al. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7).

Hou, S.; Pan, X.; Loy, C. C.; Wang, Z.; and Lin, D. 2019. Learning a Unified Classifier Incrementally via Rebalancing. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 831–839.

Huang, T.; You, S.; Wang, F.; Qian, C.; and Xu, C. 2022. Knowledge distillation from a stronger teacher. *Advances in Neural Information Processing Systems*, 35: 33716–33727.

Krizhevsky, A.; and Hinton, G. 2009. Learning multiple layers of features from tiny images. *Handbook of Systemic Autoimmune Diseases*, 1(4).

Lange, M. D.; van de Ven, G. M.; and Tuytelaars, T. 2023. Continual evaluation for lifelong learning: Identifying the stability gap. In *The Eleventh International Conference on Learning Representations*.

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

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

Liu, Y.; Li, Y.; Schiele, B.; and Sun, Q. 2024. Wakening Past Concepts without Past Data: Class-Incremental Learning from Online Placebos. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2226–2235.

Masana, M.; Liu, X.; Twardowski, B.; Menta, M.; Bagdanov, A. D.; and van de Weijer, J. 2022. Class-incremental learning: survey and performance evaluation on image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

McCloskey, M.; and Cohen, N. J. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, 109–165. Elsevier.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

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 (CVPR)*, 2001–2010.

Sahu, K. K. 2015. Normalization: A preprocessing stage. *arXiv preprint arXiv:1503.06462*.

Singh, D.; and Singh, B. 2020. Investigating the impact of data normalization on classification performance. *Applied Soft Computing*, 97: 105524.

Sun, S.; Ren, W.; Li, J.; Wang, R.; and Cao, X. 2024. Logit Standardization in Knowledge Distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 15731–15740.

Tung, F.; and Mori, G. 2019. Similarity-preserving knowledge distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 1365–1374.

Wang, L.; Zhang, X.; Su, H.; and Zhu, J. 2024. A Comprehensive Survey of Continual Learning: Theory, Method and Application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(8): 5362–5383.

Wu, Y.; Chen, Y.; Wang, L.; Ye, Y.; Liu, Z.; Guo, Y.; and Fu, Y. 2019. Large scale incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 374–382.

Zhao, B.; Xiao, X.; Gan, G.; Zhang, B.; and Xia, S.-T. 2020. Maintaining discrimination and fairness in class incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 13208–13217.

Zhou, D.-W.; Wang, F.-Y.; Ye, H.-J.; and Zhan, D.-C. 2023. PyCIL: a Python toolbox for class-incremental learning. *SCIENCE CHINA Information Sciences*, 66(9): 197101–.

Zhu, F.; Cheng, Z.; Zhang, X.-y.; and Liu, C.-l. 2021a. Class-Incremental Learning via Dual Augmentation. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems*, volume 34, 14306–14318.

Zhu, F.; Zhang, X.-Y.; Wang, C.; Yin, F.; and Liu, C.-L. 2021b. Prototype augmentation and self-supervision for incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5871–5880.

Zhu, K.; Zhai, W.; Cao, Y.; Luo, J.; and Zha, Z.-J. 2021c. Self-sustaining representation expansion for non-exemplar class-incremental learning. In *Advances in Neural Information Processing Systems*, 14306–14318.