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

Anonymous submission

Algorithm 2: Training Algorithm of Our Method

def $\mathcal{Z}$-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: $[k, O+N]$)
$\hat{Z}_t$: Teacher Output Logits (shape: $[k, O]$)
τ: Temperature Scalar
α, β: Hyperparameters

Calculate the Inter-Class Distillation Loss:
$\hat{q}_t$ = F.softmax($\mathcal{Z}$-score($\hat{Z}_t$) / τ)
q_s = F.softmax($\mathcal{Z}$-score(Z_s)[:,:O] / τ)
kld = F.kl_div(log(q_s), $\hat{q}_t$)
$\mathcal{L}_{\text{inter}}$ = (kld.sum(1, keepdim=True)) * τ^2).mean()

Calculate the Intra-Class Distillation Loss:
$\hat{q}_t$ = F.softmax($\mathcal{Z}$-score($\hat{Z}_t$.t()) / τ)
q_s = F.softmax($\mathcal{Z}$-score(Z_s.t())[:O,:] / τ)
kld = F.kl_div(log(q_s), $\hat{q}_t$)
$\mathcal{L}_{\text{intra}}$ = (kld.sum(1, keepdim=True)) * τ^2).mean()

Calculate the Total Distillation Loss:
$\mathcal{L}_{\text{KD}} = \alpha\mathcal{L}_{\text{inter}} + \beta\mathcal{L}_{\text{intra}}$

Proof of Properties of $\mathcal{Z}$-score Normalization

In this section, we prove the three key properties of $\mathcal{Z}$-score normalization—***zero mean***, ***unit standard deviation***, and ***monotonicity***—which ensure it acts as a monotonic positive linear transformation, invariant to scale and shift changes, preserving isotonic semantic information.

Proof of Zero Mean

Assume we have a dataset $\{X_1, X_2, \ldots, X_n\}$ with mean μ and standard deviation σ. After $\mathcal{Z}$-score normalization, the dataset becomes $\{Z_1, Z_2, \ldots, Z_n\}$ where:

$$Z_i = \frac{X_i - \mu}{\sigma}$$

The mean of the normalized data is:

$$\mu_Z = \frac{1}{n}\sum_{i=1}^{n} Z_i = \frac{1}{n}\sum_{i=1}^{n} \frac{X_i - \mu}{\sigma} = \frac{1}{\sigma}\cdot\frac{1}{n}\sum_{i=1}^{n}(X_i - \mu)$$

Since $\sum_{i=1}^{n} X_i = n\mu$, we have:

$$\mu_Z = \frac{1}{\sigma}\cdot\frac{1}{n}\cdot(n\mu - n\mu) = \frac{1}{\sigma}\cdot 0 = 0$$

Thus, the mean μ_Z of the $\mathcal{Z}$-score normalized data is zero.

Proof of Unit Standard Deviation

The standard deviation of the normalized data is:

$$\sigma_Z = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Z_i - \mu_Z)^2}$$

Since we know that the mean of the Z-score normalized data is 0, i.e., $\mu_Z = 0$, the formula simplifies to:

$$\sigma_Z = \sqrt{\frac{1}{n}\sum_{i=1}^{n} Z_i^2} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{X_i - \mu}{\sigma}\right)^2}$$

Expanding this, we get:

$$\sigma_Z = \frac{1}{\sigma}\sqrt{\frac{1}{n}\sum_{i=1}^{n}(X_i - \mu)^2}$$

Notice that $\sqrt{\frac{1}{n}\sum_{i=1}^{n}(X_i - \mu)^2}$ is the original standard deviation σ, so:

$$\sigma_Z = \frac{1}{\sigma}\times\sigma = 1$$

Thus, the standard deviation of the Z-score normalized data is 1.

Proof of Monotonicity

Assume for any two data points X_a and X_b, if $X_a > X_b$, then the corresponding $\mathcal{Z}$-scores are:

$$Z_a = \frac{X_a - \mu}{\sigma}, \quad Z_b = \frac{X_b - \mu}{\sigma}$$

Since $X_a > X_b$, we have:

$$X_a - \mu > X_b - \mu$$

Given that $\sigma > 0$, it follows that:

$$\frac{X_a - \mu}{\sigma} > \frac{X_b - \mu}{\sigma}$$

i.e.,

$$Z_a > Z_b$$

Thus, $\mathcal{Z}$-score normalization preserves the order of the original data, proving its monotonicity.

Entropy Analysis: Demonstrating the Smoothing Effect of $\mathcal{Z}$-Score Normalization on Overconfident Teacher Logit Distributions

In the context of knowledge distillation, the transfer of implicit knowledge—often referred to as "dark knowledge"—from the teacher model to the student model is crucial for effective learning. One of the challenges arises when the teacher model exhibits overconfidence in its predictions, which can hinder the successful transfer of nuanced relational information. After applying $\mathcal{Z}$-score normalization to the logits produced by the teacher model, which not only standardizes the distribution but also enhances the entropy of the softmax output, we smooth the probability distribution and mitigates the negative effects of overconfidence, enabling a more effective transfer of dark knowledge and further reducing forgetting. Below, we provide a formal proof of this smoothing effect by analyzing the entropy before and after normalization.

1. Entropy of the Original Logit Distribution

Given a teacher model that produces logits $\{z_1, z_2, \ldots, z_K\}$ for a classification task with K classes, the predicted probabilities after applying the softmax function are:

$$p_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

The entropy $H(p)$ of this probability distribution is defined as:

$$H(p) = -\sum_{i=1}^{K} p_i \log p_i$$

This entropy measures the uncertainty or spread of the probability distribution. Lower entropy indicates a more peaked distribution, often resulting from overconfidence in one or a few classes.

2. Impact of Overconfidence on Entropy

When the logits $\{z_1, z_2, \ldots, z_K\}$ are highly disparate, the resulting softmax probabilities tend to concentrate heavily on one class. For example, if z_2 dominates, the corresponding probability p_2 will be close to 1, with the remaining probabilities p_i for $i \neq 2$ close to 0. This leads to a low entropy scenario:

$$H(p) \approx -\left(1 \cdot \log 1 + \sum_{i \neq 2} 0 \cdot \log 0\right) = 0$$

Such low entropy indicates overconfidence, which suppresses the transfer of dark knowledge, as the model fails to convey meaningful information about the less probable classes.

3. Application of $\mathcal{Z}$-Score Normalization

To counteract this overconfidence, we apply $\mathcal{Z}$-score normalization to the logits:

$$\hat{z}_i = \frac{z_i - \mu}{\sigma}$$

where $\mu = \frac{1}{K}\sum_{i=1}^{K} z_i, \sigma = \sqrt{\frac{1}{K}\sum_{i=1}^{K}(z_i - \mu)^2}$. This transformation standardizes the logits to have a mean of 0 and a standard deviation of 1, thereby reducing the disparity among the logits.

4. Resulting Softmax Output and Increased Entropy

The softmax probabilities after normalization are given by:

$$\hat{p}_i = \frac{e^{\hat{z}_i}}{\sum_{j=1}^{K} e^{\hat{z}_j}}$$

Since the logits $\hat{z}_i$ are now less extreme, the resulting probabilities $\hat{p}_i$ are more evenly distributed. This reduction in extreme values mitigates the teacher's overconfidence.

The entropy of the new distribution $\hat{p}$ is:

$$H(\hat{p}) = -\sum_{i=1}^{K} \hat{p}_i \log \hat{p}_i$$

Given that the distribution $\hat{p}_i$ is more balanced, the entropy $H(\hat{p})$ will be higher than the original entropy $H(p)$. This increase in entropy reflects a smoother probability distribution, which more effectively captures the relational information across classes, essential for the successful transfer of dark knowledge.

5. Conclusion: Enhanced Knowledge Transfer via Increased Entropy

Through this analysis, we have demonstrated that $\mathcal{Z}$-score normalization not only standardizes the logits but also increases the entropy of the resulting probability distribution. This increased entropy reflects a smoother, less overconfident output, which is more conducive to transferring the dark knowledge embedded within the teacher model's predictions. Consequently, this leads to more effective knowledge distillation and reduces the potential for model forgetting in class-incremental learning scenarios.

Implementation Details

We evaluate our methods on three widely-used benchmarks. CIFAR-100 (Krizhevsky and Hinton 2009) comprises 100 32x32 pixel color images. It includes 50,000 images for training (500 images per class) and 10,000 for evaluation (100 images per class). ImageNet-Subset (Deng et al. 2009) is built by selecting 100 classes from the ImageNet (Deng et al. 2009) dataset. Tiny-ImageNet (Le and Yang 2015) has 200 classes, each with 500 training images, 50 validation images, and 50 test images. During the incremental training, for all tasks, the initial learning rate starts from 0.1 and decays to 1/10 of the previous learning rate after 60, 120, and 160 epochs. The total epochs are set to 200 and 100 epochs

Table 1: More Standard Knowledge Distillation (KD) techniques with and without our method on different settings of the CIFAR-100 and TinyImageNet datasets.

Method	CIFAR-100								TinyImageNet							
	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
AdaptTeacher	42.85	59.37	25.20	47.51	25.36	37.60	19.98	34.31	20.49	36.07	15.60	30.07	16.13	23.88	11.17	15.88
$w/\mathcal{L}_{\text{inter}}$	45.21	62.51	21.67	45.92	23.35	45.61	34.46	55.64	25.27	39.07	**19.98**	34.29	21.17	31.80	11.33	18.07
$w/\mathcal{L}_{\text{inter}} + \mathcal{L}_{\text{intra}}$	**50.17**	**64.19**	**25.57**	**49.62**	**34.14**	**55.54**	**37.37**	**57.34**	**27.91**	**40.14**	17.40	**33.37**	**23.59**	**34.85**	**11.71**	**20.88**
Improvement	***7.32***	***4.82***	***0.37***	***2.11***	***8.78***	***17.94***	***17.39***	***23.03***	***7.42***	***4.07***	***1.80***	***3.30***	***7.46***	***10.97***	***0.54***	***5.00***

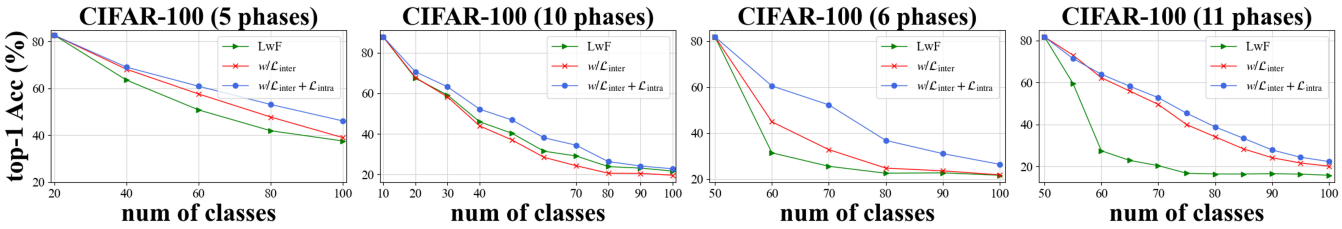

Figure 1: The average incremental accuracy curves of LwF and our methods.

in the first task and the following tasks. We set the temperature scalar γ to 2, and all KD-based methods save the old model from the last step as the teacher model. For replay-based methods, the memory size for storing samples of old classes is set to 2000, the same as conventional settings.

More Studies

Extended Results

In Table 1, we provide more results of our method on another play-and-plug KD-based method (Szatkowski et al. 2024) on CIFAR-100 and TinyImageNet, in which the teacher's batch normalization layer can be optimized during training. We implemented it on LwF (Li and Hoiem 2017). We can observe similar results that our $\mathcal{L}_{\text{inter}}$ and $\mathcal{L}_{\text{intra}}$ consistently improve the overall performance of this baseline with obvious performance gap, especially when $\mathcal{L}_{\text{intra}}$ is introduced. Additionally, we provide the average accuracy curves to show the superior performance of our method in Fig. 1.

Visualization

We present visualizations by t-SNE 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 after training the second task. In Figure 2, we randomly select five old classes for visualization and calculate the Silhouette Score (Rousseeuw 1987) to provide a quantitative evaluation of the clustering's overall quality. The larger the score, the better the clustering. We can observe that the performance is improved obviously when our $\mathcal{L}_{\text{inter}}$ and $\mathcal{L}_{\text{intra}}$ are applied to vanilla KD, as the representations on Figure 2 (c) and Figure 2 (d) are separated more apparently. The scores are also obviously higher in the two figures, demonstrating the superiority of our effective anti-forgetting.

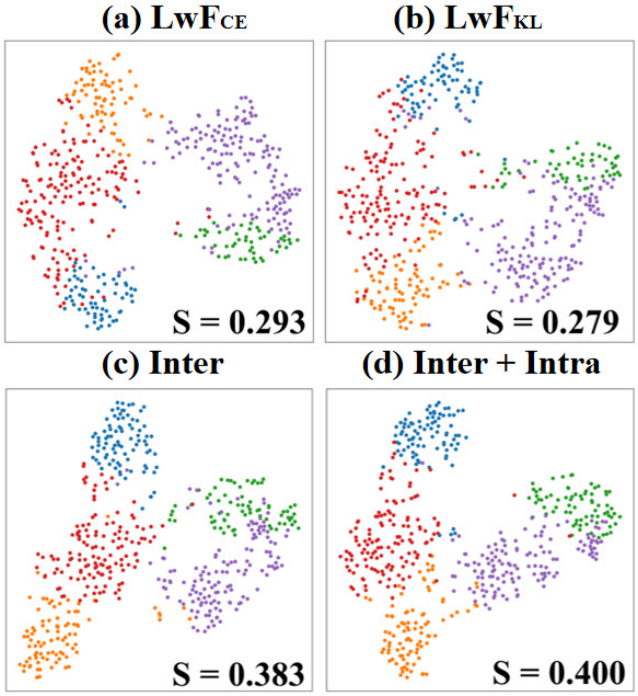

Figure 2: t-SNE visualization for different KD mechanisms.

References

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 248–255.

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

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.

Rousseeuw, P. J. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20: 53–65.

Szatkowski, F.; Pyla, M.; Przewike zlikowski, M.; Cygert, S.; Twardowski, B.; and Trzciński, T. 2024. Adapt Your Teacher: Improving Knowledge Distillation for Exemplar-Free Continual Learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 1977–1987.