We appreciate all reviewers for their comments. Below we answer point-by-point all reviewers' questions.

1 Response to Reviewer #1CA1

Reviewer #1CA1(R1-1)

The authors only provide results on small-scale datasets, e.g., CIFAR-100. However, it is quite common to evaluate class-incremental learning on large-scale datasets. For example, iCaRl [1], BiC [2], LUCIR [3], Mnemonics [4], and PODNet-CNN [5] all provide result on ImageNet (full size, 1000 classes). Thus, it is not reasonable to ignore large-scale datasets for a top-tier conference submission.

Our Reply (R1-1)

We appreciate the reviewer's comment. We agree with the reviewer's comment, but please note that still there is much room for improvement on small-scale datasets, and many latest works also conduct experiments on small image datasets in CIL, e.g., DualAug [6], Co2L [7], etc. We conduct experiments to compare our method with DualAug and Co2L. The results are shown in Tabs.1 and 2. As shown in Tabs.1 and 2, EDBL outperforms them by large margins, about [1.5%, 8%] in class incremental learning (Class-IL, CIL) setting on CIFAR-10, CIFAR-100 and Tiny-Imagenet. EDBL also performs well in task incremental learning (Task-IL) setting, from Tabs.1, we can find that EDBL surpasses Co2L, DER and DER++ [8] by about [3%, 14%] in the Task-IL setting on CIFAR-10 and Tiny-Imagenet.

Second, due to large GPU memory required by re-sampling Mixup during training, it is hard to accomplish Re-MKD on ImageNet-1000 with the shape of (224,224) in that short time (rebuttal period). We estimate the training time of the experiment on ImageNet-1000 is more than 15 days on 4 NVIDIA 3090. In future works, we will conduct experiments on ImageNet-1000 with Re-MKD and EDBL and reports the results on the final version of this paper.

Reviewer #1CA1(R1-2)

The technical novelty of this paper is somewhat limited. This paper is based on re-sampling and mixup. Both strategies have already been widely used in many related topics, such as long-tailed recognition and few-shot learning. Thus, the technical contributions of this paper are somewhat limited.

Our Reply (R1-2)

Re-sampling and Mixup have already been widely used in many related topics, such as long-tailed recognition and few-shot learning, but they mainly applied in classification while we apply them to tackle long tail knowledge distillation (KD training with class imbalanced data, referred to as LT-KD) in CIL. Moreover, Beyer et al. [9] and Wang et al. [10] apply mixup to improve KD, but they use the data that are independent-identically-distributed (**iid**) with the original training data to make KD training, they can't verify their effectiveness of their methods in CIL setting, where the training data are long-tailed and OOD. In this work, our contribution lies on validating effectiveness of re-sampling and Mixup between old classes and new classes to tackle KD in the CIL setting. In addition, because KD with the long-tailed and OOD data is not the typical LT-KD training, so we first apply re-sampling and mixup to improve the performance of KD with OOD data in the first training stage. Then in the second training stage, which has become an typical LT-KD training, we develop IIB factor especially the KD weighting factor to re-weight high influenced samples in the balanced training to tackle long tail KD to fine-tune the new model. As we known, this is the first time to focus on KD with OOD data and the LT-KD problem in the CIL setting.

Reviewer #1CA1(R1-3)

The state-of-the-art method is compared in Table 1. For example, Der [14] archives much better performance than other baselines, but it is not compared in this paper. It seems the proposed method cannot achieve better performance than Der.

Our Reply (R1-3)

Table 1: Results of experiments conducted according to the protocol in Co2L [7]. Buffer size is 500, the incremental learning phases are 5 for CIFAR-10 and 10 for Tiny-Imagenet, * denotes using the results in [7]. (Average Acc. after completing the last phase, %)

Buffer Size	Dataset	CIFAR-10		Tiny-Imagenet	
	Scenarios	Class-IL	Task-IL	Class-IL	Task-IL
500	A-GEM* [11]	22.67	89.48	8.06	25.33
	iCaRL* [1]	47.55	88.22	9.38	31.55
	FDR* [12]	28.71	93.29	10.54	49.88
	DER* [8]	70.51	93.40	17.75	51.78
	DER++* [8]	72.70	93.88	19.381	51.91
	Co2L* [7]	74.26	95.90	20.12	53.04
	EDBL-NME(ours)	**77.01**	**96.86**	**28.06**	**67.2**

Table 2: Average incremental accuracy (%) on Base-half experiments. Models with an asterisk * denotes using the results in [6].

Base-half	CIFAR-100		Tiny-imagenet	
Phases	5	10	5	10
iCaRL [1]	59.67	56.13	48.98	39.27
BiC [2]	61.14	58.4	49.23	47.67
LUCIR [3]	63.17	60.14	49.31	47.56
SSIL-NME [13]	64.94	60.99	48.93	45.74
DualAug*[1] [13]	65.3	57.85	47.1	44.75
EDBL-CNN(ours)	66.57	62.06	**52.43**	**53.80**
EDBL-NME(ours)	**66.65**	**64.73**	50.99	49.97

We learned about Der and Der achieves the SOTA results. Der employs the technique of dynamically expanding network, where it first expands the network, and then compresses the model by pruning technology. Der focuses on reliving catastrophic forgetting and its scallability is completely based on pruning technique. However, our work focuses on tackling KD with OOD data and LT-KD to alleviate forgetting, we will conduct experiments and discuss the influences of EDBL on knowledge transferring or model compressing by using the long-tail and OOD data in future work.

2 Response to Reviewer #JLqd

Reviewer #JLqd(R2-1)

The authors repeatedly emphasize that data of newly added classes, which are OOD data from the perspective of the previously learned classes, harm the knowledge distillation process. However, the proposed framework still uses the mixed data of the newly-added data for knowledge distillation. It would be nice if the paper includes the ablation study about the 3 types of mixing classes mentioned in Line 158-159. In other words, I wonder how the overall incremental learning performance will be affected when we exclude the mixed data of the newly added classes from computing L_{kd}.

Our Reply (R2-1)

On the one hand, the stored exemplars of the old classes are consistent (iid) with the original training data of the old model, so they should be used in KD training to preserve the old knowledge. On the other hand, Beyer et al. [9] demonstrates using OOD data has some effects in KD to some extent and most KD-based methods [1, 2, 15, 13] in CIL use the data of the added classes to make transferring the old knowledge. We conduct the experiments on excluding the mixed data of the newly added classes. As shown in Tab. 3, when excluding the mixed data of the newly added classes in KD training, both the performances of baseline and EDBL-NME degrade drastically. From our experiments, using the data of the added classes improves the performance of KD in CIL. However, Beyer et al. [9] also validate empirically that the effect of using the OOD data is very limited, compared with using the original training data for KD. So based on this observation, we focus on improving the effect of using OOD data in KD training by re-sampling and Mixup.

Reviewer #JLqd(R2-2)

Why is the IIB factor only multiplied to L_{ce} Since L_{kd} also can affect the overall decision boundary and IIB factor also considers L_{kd} as in eq (10), IIB weighting factor may be helpful to L_{kd} in eq (16),

Table 3: Results of experiments on excluding the mixed data of the newly added classes in KD training.

Dataset	CIFAR-100				
phase	1	2	3	4	5
Baseline-NME	0.848	0.733	0.6567	0.5909	0.5447
Baseline-NME + exclude	0.848	0.6987	0.6290	0.5651	0.5219
EDBL-NME	0.85	0.7525	0.6805	0.6295	0.5752
EDBL-NME + exclude	0.848	0.6395	0.5838	0.5262	0.4855

Table 4: Results of Re-MKD + CBF on CIFAR-100 with 5 pahses in Base-0 protocol (Average Accuracy on each incremental phase).

Dataset	CIFAR-100				
phase	1	2	3	4	5
BiC	0.848	0.74	0.667	0.61.5	0.565
+ Re-MKD	0.848	0.71.73	0.5936	0.5759	0.5351
EEIL	0.835	0.765	0.642	0.591	0.528
+ Re-MKD	0.848	0.7185	0.6478	0.5814	0.5284
IIB-KD(One-stage)	0.8385	0.6947	0.603	0.5315	0.487
+ Re-MKD(EDBL)	0.848	0.767	0.7093	0.6573	0.6051

too. If possible, it would be nice to see how the final performance will be affected if IIB factor is also considered to L_{kd}. Or, if I missed something, please let me know.

Our Reply (R2-2)

In our experiments, when multiplying IIB factor to L_{kd}, IIB-KD methods achieves average accuracy 52.53% after completing the last training phase on CIFAR-100 with the Base0 and 5 phases protocol, compared with 53.62% without multiplying IIB factor to L_{kd}. Therefore, according to the observed phenomenon, we only multiply IIB factor to L_{ce} based on experiments.

Reviewer #JLqd(R2-3)

Conventionally, many CIL methods [15, 3, 5] utilize additional “class balanced fine-tuning (CBF).” It seems that the second phase of EDBL algorithm, which is balanced training, can be a good replacement for CBF, since CBF only utilizes small number of examples unlike the proposed framework. Thus, it would be nice if the ablation study of balanced training is added to the paper. Or, is “Re-MKD” row in Table 3 indicates the Re-MKD + CBF?

Our Reply (R2-3)

Both EEIL and BiC apply CBF technique and they have two training stages, where EEIL [15] utilizes classical image transformation to make data augmentation and then fine-tune the newly trained model with a balanced dataset while BiC [2] trains a bias-correction classification layer with a balanced dataset. We conduct experiments with Re-MKD + EEIL and Re-MKD + BiC on CIFAR-100 with Base0 and 5 phases protocol to study the effect of Re-MKD on the CBF technique. The results in Tab. 4 suggest that Re-MKD has different affects on different CBF technique. Re-MKD improves the performance of IIB-KD but it does not improve the effect of EEIL and BiC.

Reviewer #JLqd(R2-4)

Figure 3 can be improved if it contains more classes than two classes. Even though it is a minor issue for me, it seems demonstrating only a single case may weaken the claim of the paper. There are some serious formatting issues throughout the paper and Minor notation error.

Our Reply (R2-4)

Thank you for the suggestions. We will provide visualization results with more classes and more incrmental learning phases and correct the errors and improve the presentation accordingly in the new version.

3 Response to Reviewer #PDXX

Reviewer #PDXX(R3-1)

According to Algorithm 1 in the Appendix, the main difference between Phase 1 (MKD) and Phase 2 (Balancing training) is whether IIB is incorporated or not. Why not having only Phase 2? A comparison of the last 2 rows in Table 3 (if I understand correctly) seems to indicate the benefit of adding Phase 1 (but seems to be not discussed in Section 5.5.1). Further insights would be important.

Our Reply (R3-1)

EDBL has two training stages, where the first is to train a new model by Re-MKD, then fine-tunes it in the balanced training. Because the data of the added classes are OOD, so the KD training in the first stage is not a typical long tail KD training (Long tail KD training refers to distillation with long tail data, LT-KD), so we apply typical RKD method to train a new model and use Re-MKD to improve the knowledge transferring. After we obtain a new model, the second training stage has become a typical long-tail KD training and we attempt to fine-tune it by using some technique to tackle LT-KD, so we compute IIB factor especially the KD weighting factor to re-weight the high influenced samples in the second training stage. We conduct experiments with only one training stage using IIB-KD, the results are shown in Tab. 4. From Tab. 4, we can find that directly using IIB-KD to train a new model perform worse than EDBL significantly.

Reviewer #PDXX(R3-2)

Eq 11: the first term seems to be missing 2* for $f^t(x)$ and h. If so, Eq 13 and Eq 15 seem to have similar issues.

Our Reply (R3-2)

We add Eq.9 to Eq.10, then we get:

$$\frac{\partial L_{ce}(x,\Theta)}{\partial w_{kl}} + \frac{\partial L_{kd}(x,\Theta)}{\partial w_{kl}} = \begin{cases} (f_k^t(x) - y_k - f_k^{t-1}(x))h_l, & k \in [1, m] \\ (f_k^t(x) - y_k)h_l, & k \in [m+1, m+n] \end{cases} \tag{1}$$

, apply this result to Eq.8 then we get Eq.11, so Eqs.11, 13, 15 are correct.

Reviewer #PDXX(R3-3)

Lines 223, 234, and 236: balancing training -> balanced training? Phases are used in both EDBL and CIL. Using different terms could be easier to separate the two concepts; e.g. stages in EDBL and phases in CIL.

Our Reply (R3-2)

Thank you for the suggestions. We will correct the errors and improve the presentation accordingly in the new version.

4 Response to Reviewer #Fq7t

Reviewer #Fq7t(R4-1)

This paper is technically sound and the claims are supported. However, the experimental setup is flawed. The authors should discuss and compare more batch continual learning methods which use the KD mechanism (1,2,3): (1)DualAug [6], (2)DER [8], (3)Co2L [7].

Our Reply (R4-1)

We conduct experiments according to the protocol in Co2L [7] and we apply the same model (ResNet 18), the same data augmentation to conduct experiments as in Co2L. The results are shown in Tab. 1. From Tab. 1, we can find that our method significantly outperform Co2L and DER by large margins on CIFAR-10 and Tiny-Imagenet, where EDBL surpasses the best baseline by about [3%, 8%] in Class-IL and about [1%, 14%] in Task-IL on CIFAR-10 and Tiny-Imagenet. We also compare our method with DualAug[6]. From Tab. 2, our method outperforms DualAug by large margins about [1%, 7%]. Please refer to Tabs. 1 and 2 for more information.

Reviewer #Fq7t(R4-2)

The EDBL method generates more mixed data for the two types of mixup among old classes and mixup between old classes and new classes than the type of mixup among new classes. But the rate (N/2) set for the first two types is just an empirical value. Please try to investigate the performance

Table 5: Results of study on re-sampling rate.

Dataset	CIFAR-10					CIFAR-100				
phase	1	2	3	4	5	1	2	3	4	5
Baseline-NME	0.993	0.8515	0.6427	0.5839	0.6124	0.848	0.6952	0.6142	0.5552	0.5002
4			/			0.848	0.6977	0.634	0.582	0.5391
2	0.995	0.7972	0.7168	0.6948	0.669	0.848	0.6827	0.6332	0.5875	0.5447
1	0.995	0.8687	0.7533	0.7414	0.7127	0.848	0.7217	0.6505	0.5899	0.5527
0.5	0.995	0.8732	0.7648	0.7054	0.6864	0.848	0.7225	0.6495	0.5834	0.5414
0.25			/			0.848	0.729	0.641	0.5755	0.534

of EDBL at different rates and illustrate the results. Also, what does the "tail and head class" mean (Line 162)?

Our Reply (R4-2)

We conduct study on the re-sampling rate. We divide $N/2$ by the rates in the Tab. 5 to conduct experiments on CIFAR-10 and CIFAR-100 with 5 phases in Base-0 protocol. We report the results of EDBL-NME and also give the results of baseline-NME to be compared in Tab. 5 . The result in Tab. 5 suggests that the re-sampling rate too high or too low is not good. The "tail and head class" in (Line 162) denotes the old and new classes, because the stored exemplars of the old classes are limited while there are large-scale data for the new classes, so the training data are long-tailed.

Reviewer #Fq7t(R4-3)

Can the EDBL method help the KD method to alleviate the forgetting problem further? Please calculate and compare the average forgetting rate of the EDBL method and other baselines.

Table 6: Average accuracy and Average forgetting rate (%) after completing the last learning phase.

Base-0	CIFAR-10		CIFAR-100	
/	ACC (↑)	FGT (↓)	ACC (↑)	FGT (↓)
iCaRL [1]	66.06	20.98	54.2	11.15
BiC [2]	65.97	19.58	56.5	10.77
SSIL-NME [13]	66.17	8.55	53.54	8.28
EDBL-CNN(ours)	66.18	**6.45**	**60.51**	**5.14**
EDBL-NME(ours)	**68.77**	12.90	57.52	12.92

Our Reply (R4-3)

We calculate the average forgetting rate (FGT) of EDBL and other baselines, e.g. SSIL, iCaRL, BiC according to the FGT formulation given in [16]. The comparisons of FGT and average accuracy between them are shown in Tab. 6. As shown in Tab.6, the EDBL method especially EDBL-CNN has much lower average forgetting rate.

Reviewer #Fq7t(R4-4)

The paper says the classification weighting factor is for learning new tasks. But it also calculates the classification weighting factor of the mixed data generated by the samples from old classes. Can the author explain the role of the classification weighting factor in this case?

Our Reply (R4-4)

When learning new tasks, EDBL trains the new model via minimizing the classification cross-entropy loss on all the classes including the old and new classes. Because the training data are class imbalanced, EDBL computed the classification weighting factor to re-weight all the high influenced samples to tackle the long-tail classification learning.

Reviewer #Fq7t(R4-5)

That is just a minor question: does the EDBL method also work well in the few-shot continual learning setting? In this setting, the model is over-fitting to the new data severely. And the IIB loss may have a significant impact in this setting.

Our Reply (R4-5)

Thank you for the suggestions. The few-shot continual learning is the most cases in reality. We will apply EDBL to the few-shot continual learning to study the effect of EDBL on the few-shot learning.

References

[1] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017.

[2] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 374–382, 2019.

[3] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 831–839, 2019.

[4] Yaoyao Liu, Yuting Su, An-An Liu, Bernt Schiele, and Qianru Sun. Mnemonics training: Multi-class incremental learning without forgetting. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 12245–12254, 2020.

[5] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16*, pages 86–102. Springer, 2020.

[6] Fei Zhu, Zhen Cheng, Xu-yao Zhang, and Cheng-lin Liu. Class-incremental learning via dual augmentation. *Advances in Neural Information Processing Systems*, 34, 2021.

[7] Hyuntak Cha, Jaeho Lee, and Jinwoo Shin. Co2l: Contrastive continual learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9516–9525, 2021.

[8] Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. *Advances in neural information processing systems*, 33:15920–15930, 2020.

[9] Lucas Beyer, Xiaohua Zhai, Amélie Royer, Larisa Markeeva, Rohan Anil, and Alexander Kolesnikov. Knowledge distillation: A good teacher is patient and consistent. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10925–10934, 2022.

[10] Dongdong Wang, Yandong Li, Liqiang Wang, and Boqing Gong. Neural networks are more productive teachers than human raters: Active mixup for data-efficient knowledge distillation from a blackbox model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1498–1507, 2020.

[11] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. In *International Conference on Learning Representations*, 2018.

[12] Ari Benjamin, David Rolnick, and Konrad Kording. Measuring and regularizing networks in function space. In *International Conference on Learning Representations*, 2018.

[13] Hongjoon Ahn, Jihwan Kwak, Subin Lim, Hyeonsu Bang, Hyojun Kim, and Taesup Moon. Ss-il: Separated softmax for incremental learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 844–853, 2021.

[14] Shipeng Yan, Jiangwei Xie, and Xuming He. Der: Dynamically expandable representation for class incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3014–3023, 2021.

[15] Francisco M Castro, Manuel J Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. In *Proceedings of the European conference on computer vision (ECCV)*, pages 233–248, 2018.

[16] Kibok Lee, Kimin Lee, Jinwoo Shin, and Honglak Lee. Overcoming catastrophic forgetting with unlabeled data in the wild. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 312–321, 2019.