UNDER REVIEW

RESCIENCE C

Replication / ML Reproducibility Challenge 2022

[Re] Pure Noise to the Rescue of Insufficient Data

Anonymous[1, ID]

[1]Anonymous Institution

Edited by
(Editor)

Reviewed by
(Reviewer 1)
(Reviewer 2)

Received
03 February 2023

Published
–

DOI
–

Reproducibility Summary

Scope of Reproducibility – We examine the main claims of the original paper [1], which states that in an image classification task with imbalanced training data, (i) using pure noise to augment minority-class images encourages generalization by improving minority-class accuracy. This method is paired with (ii) a new batch normalization layer that normalizes noise images using affine parameters learned from natural images, which improves the model's performance. Moreover, (iii) this improvement is robust to varying levels of data augmentation. Finally, the authors propose that (iv) adding pure noise images can improve classification even on balanced training data.

Methodology – We implemented the training pipeline from the description of the paper using PyTorch and integrated authors' code snippets for sampling pure noise images and batch normalizing noise and natural images separately. All of our experiments were run on a machine from a cloud computing service with one NVIDIA RTX A5000 Graphics Card and had a total computational time of approximately 432 GPU hours.

Results – We reproduced the main claims that (i) oversampling with pure noise improves generalization by improving the minority-class accuracy, (ii) the proposed batch normalization (BN) method outperforms baselines, (iii) and this improvement is robust across data augmentations. Our results also support that (iv) adding pure noise images can improve classification on balanced training data. However, additional experiments suggest that the performance improvement from OPeN may be more orthogonal to the improvement caused by a bigger network or more complex data augmentation.

What was easy – The code snippet in the original paper was thoroughly documented and was easy to use. The authors also clearly documented most of the hyperparameters that were used in the main experiments.

What was difficult – The repo linked in the original paper was not populated yet. As a result, we had to retrieve the CIFAR-10-LT dataset from previous works [2, 3], re-implement WideResNet [4], and the overall training pipeline.

Communication with original authors – We contacted the authors for clarifications on the implementation details of the algorithm. Prior works had many important implementation details such as linear learning rate warmup or deferred oversampling, so we confirmed with the authors on whether these methods were used.

Copyright © 2023 Anonymous, released under a Creative Commons Attribution 4.0 International license.
Correspondence should be addressed to ()
The authors have declared that no competing interests exists.
Code is available at https://github.com/rescience-c/template.

1 Introduction

Real-world datasets often have long-tailed label distributions, for example because some classes are more rare in the real world, the acquisition source is innately biased towards a few labels, or because some classes are easier to label than others. Deep neural networks often perform poorly on less-represented classes as the model is easily biased towards majority classes and results in poor generalization for minority classes.
Well-known approaches to mitigating the class imbalance problem are re-weighting the loss and re-sampling during training. However, both approaches can encourage the model to overfit to the minority class [1, 3, 5].
Zada, Benou, and Irani [1] proposed **O**versampling with **P**ur**e** **N**oise Images (OPeN), a new re-sampling technique of replacing some oversampled images with pure noise images. During training, OPeN replaces some images in a mini-batch with pure noise images generated at the beginning of each epoch. The probability of replacing an image x of class i with a noise image x_{noise} is proportional to the rate of oversampling δ:

$$\mathbb{P}(\text{Replace } x \text{ with } x_{\text{noise}} \mid \text{Class} = i) = \left(1 - \frac{n_i}{\max_j n_j}\right) \cdot \delta \tag{1}$$

where n_i is the number of samples for each class i.
OPeN creates mini-batches containing images from two different distributions: the CIFAR-10 distribution and the pure noise distribution. Since batch normalization (BN) [6] intrinsically assumes that the input comes from a single distribution, Zada, Benou, and Irani [1] also propose Distribution Aware Routing Batch Normalization (DAR-BN) that replaces the BN layers. DAR-BN separates the pure noise images to normalize the activation maps separately from the natural images.

2 Scope of reproducibility

We investigate the following claims from Zada, Benou, and Irani [1]. We list in parentheses the figures in the original paper that correspond to each claim.

1. OPeN improves model performance on CIFAR-10/100-LT by improving accuracies on classes with lower frequencies. (Table 1, Figure 7)
2. DAR-BN improves the performance of OPeN on CIFAR-10/100-LT compared to baseline Batch Normalization methods. (Table 4)
3. The performance improvement of OPeN is robust under various data augmentation methods. (Figure 3)
4. OPeN improves performance on the full CIFAR-10/100 dataset. (Section 5)

3 Methodology

3.1 Model descriptions

For CIFAR-10-LT and CIFAR-100-LT datasets, Zada, Benou, and Irani [1] use the WideResNet-28-10 [4] architecture. Because the author's code was not public, we modified the implementation by Matsubara [7] by replacing batch normalization layers with DAR-BN.

3.2 Datasets

Zada, Benou, and Irani [1] used 5 datasets: CIFAR-10-LT, CIFAR-100-LT, CelebA-5, ImageNet-LT, and Places-LT. As the authors only reported results from CIFAR-10-LT and CIFAR-100-LT in their ablation studies, we also focus on these two datasets.
CIFAR-10-LT and CIFAR-100-LT are long-tailed variants of the CIFAR-10 and CIFAR-100 datasets respectively, proposed by Cui et al. [8]. These long-tailed training datasets are created by reducing the number of training samples following an exponential function $n_i \cdot \mathrm{IR}^{i/(C-1)}$, where C is the number of classes in the dataset and i is a class index from 0 to $C-1$. IR denotes the imbalance ratio of the dataset, defined as the ratio of frequencies of the largest and smallest classes.

Dataset	Imbalance Ratio (IR)	Number of training examples
CIFAR-10-LT	100	12406
CIFAR-10-LT	50	13996
CIFAR-100-LT	100	10847
CIFAR-100-LT	50	12608

Table 1. Different long-tail variants of the CIFAR-10/100 datasets. A higher imbalance ratio signifies that the dataset is more imbalanced.

For evaluation, we compute the accuracy using the original CIFAR-10/100 validation dataset of 10000 images. This allows for evaluation on a balanced set of examples, penalizing models that focus on majority classes during training.
For normalizing the input images, we used the per-channel mean of $(0.4914, 0.4822, 0.4465)$ and standard deviation of $(0.2023, 0.1994, 0.2010)$ for both datasets, following Zhong et al. [9] and Cao et al. [2]. However, we found them to differ from the values we computed, so we conduct additional experiments in Section A.2.1 of the Appendix.

3.3 Hyperparameters

To provide a complete overview of the experiments, we use this section to list all the hyperparameters. For all experiments in the paper, unless specified, the experiment settings match that of Table 2.

Hyperparameters	Values	Hyperparameters	Values
Model	WideResNet-28-10	Initial learning rate (lr)	0.1
Dropout rate	0.3	lr decay epochs	$160, 180$
Batch size*	128	lr decay gamma	0.01
Optimizer	SGD	Linear warmup epochs*	5
Momentum	0.9	OPeN noise image ratio (δ)	$1/3$
Weight decay	2×10^{-4}	OPeN start epoch	160

Table 2. Default hyperparameters used for experiments. * denote hyperparameters not described in Zada, Benou, and Irani [1] but confirmed through email. Check Section 5.2 for more details.

3.4 Experimental setup and code

As the authors have not released the code yet, we re-implemented most of the code from the description of the paper while using open-source code from prior works. We imported long-tailed dataset generation from Cao et al. [2], and the base WideResNet model from Matsubara [7], which we modified to use DAR-BN. We used the code snippets from the original paper for noise image generation and parts of DAR-BN.
We used Weights and Biases [10] for tracking experiments, and OmegaConf, a subset of Hydra [11], for configuring hyperparameters. All the code used to run experiments in

this paper has been anonymized and submitted with the paper as supplementary material and available at https://anonymous.4open.science/r/pure-noise-4166/. It will be released on GitHub once the Reproducibility Challenge is finished.

3.5 Computational requirements

All experiments were performed on a cloud computing service using virtual machines with 12 vCPU, 62 GB RAM, and one NVIDIA RTX A5000 graphics card with 24 GB VRAM. Using the default experiment setting specified in Table 2, Empirical Risk Minimization (training the model without oversampling or OPeN) took approximately 1 hour and 50 minutes. Using the checkpoints saved after 160 epochs, training with deferred oversampling took 22 minutes, and training with OPeN took 45 minutes.

4 Results

4.1 Results reproducing original paper

OPeN encourages generalization by improving minority-class accuracy — To verify Claim 1, we trained the model using four different oversampling schemes: (i) Empirical Risk Minimization (ERM): training without oversampling (ii) Resampling (RS): sampling by weights inverse of class frequency (iii) Deferred Resampling (DRS): deferring RS to last phase of training (iv) OPeN: oversampling with pure noise during the same last phase of training. For CIFAR-10-LT (IR=100) dataset, we reproduced the mean validation accuracy of the DRS baseline and OPeN to within 0.6% of the reported value, which supports Claim 1. For other datasets and IR ratios, the performance of DRS was not reported in the original paper. We measured the performance of DRS for those datasets because DRS is the fair baseline for OPeN as both methods use the same deferred resampling schedule. OPeN outperformed the baselines across all datasets.

Source	Reported [1]	Ours
ERM	79.6	81.18
RS	75.1	74.82
DRS	83.0	83.22
OPeN	84.6	85.04

Table 3. Comparing accuracy of resampling schemes on CIFAR-10-LT (IR=100) dataset. Reported accuracy are from Table 1 in Zada, Benou, and Irani [1].

We also compute the per-class accuracies to understand if the improvement is from minority classes. Indeed, we confirm that compared to DRS, OPeN improves the performance of the two least frequent classes by 8.2% while sacrificing only 0.9% accuracy for the two most frequent classes. For a complete comparison, we ask the readers to look at Figure 5 and Figure 6 in the Appendix.

DAR-BN outperforms other batch normalization layers when used with OPeN — To verify Claim 2, we trained three models with different batch normalization: (i) Standard BN [6]: normalizing pure noise and natural activation maps together using one BN layer (ii) Auxiliary BN [12]: normalizing pure noise and natural activation maps separately using two BN layers (iii) Distribution-Aware Routing BN (DAR-BN) [1]: using the affine parameters learned from natural activation maps to normalize noise activation maps. For CIFAR-10-LT (IR=100) and CIFAR-100-LT (IR=100) datasets, DAR-BN outperformed Standard BN and Auxiliary BN in terms of mean validation accuracy (Table 4). This supports the claim

and shows that DAR-BN is essential to the success of OPeN, as without DAR-BN, the accuracy is lower than the accuracy of DRS (83.22). In Table 8, we also perform the same experiment on ResNet and come to the same conclusion, further validating the claim.

Dataset	CIFAR-10-LT		CIFAR-100-LT	
Source	Reported [1]	Ours	Reported [1]	Ours
Standard BN [6]	81.45	81.81	49.18	49.26
Auxiliary BN [12]	83.38	82.23	50.13	51.27
DAR-BN [1]	84.64	85.04	51.50	52.12

Table 4. Ablation experiment: comparing DAR-BN with other Batch Normalization layers (IR=100). Reported scores are from Table 4 in Zada, Benou, and Irani [1].

OPeN is robust to various data augmentation methods — To verify Claim 3, we compared ERM, DRS, and OPeN on CIFAR-10-LT (IR=100) dataset using three data augmentations of increasing strength: (i) random horizontal flip and random 32x32 pixel crop with padding of 4 (ii) add Cutout, [13] which zeros out one 16x16 pixel patch (iii) add SimCLR, [14] which randomly applies color jitter, grayscale, and Gaussian blur. OPeN outperformed DRS and ERM across all augmentations, which supports the claim. We forgo AutoAugment [15] for this ablation study because AutoAugment was optimized using the full balanced dataset and is an unfair augmentation strategy for the imbalanced sub-dataset [1, 16].

	Flip and Crop		Add Cutout		Add SimCLR	
Source	Reported	Ours	Reported	Ours	Reported	Ours
ERM	74.3	74.6	77.7	78.7	79.6	80.7
DRS [2]	76.5	75.4	80.3	79.5	83.0	83.2
OPeN [1]	80.3	79.9	83.1	83.9	84.6	84.3

Table 5. Data augmentation ablation experiment. Reported accuracy are from Figure 3 in Zada, Benou, and Irani [1].

Adding pure noise improves performance on balanced datasets — In Claim 4, the authors propose that using pure noise is useful as a general data augmentation method beyond imbalanced datasets. That is, given a balanced dataset, we can simply add a fixed number of pure noise images to each class and train with DAR-BN. Since this approach does not modify natural images, it can complement any existing data augmentations. The authors experiment by adding random noise images with a fixed noise-to-natural ratio of $1:4$ in each batch and report percentage improvement over training without random noise images. The authors used different hyperparameters, such as Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and AutoAugment [15] for data augmentation. Furthermore, we communicated with the authors to find that a fixed learning rate of 0.001 was used without linear warmup [17]. Our results showed that adding pure noise images improve the performance on the balanced dataset.

Source	Improvement	Baseline Accuracy	Pure Noise Accuracy
Reported	+0.9%	-	-
Ours	+1.6%	87.16	88.57

Table 6. Performance improvement of OPeN on the full balanced CIFAR-10 dataset. The original paper reported the percentage improvement but not the baseline and pure noise accuracy.

4.2 Results beyond original paper

ResNet architecture — Zada, Benou, and Irani [1] used WideResNet-28-10 for their experiments with CIFAR-10 and CIFAR-100. However, prior works [2, 3, 9] used a smaller ResNet-32 network. To compare performance with results originally reported by prior works, we train OPeN on the ResNet architecture. For these experiments, we used the ResNet-32 implementation by Idelbayev [18] and replaced batch normalization layers with DAR-BN.
In Table 7, we compare OPeN with the performance reported by prior works. We find that OPeN still shows improvement over ERM, RS, and DRS. However, the comparative advantage of OPeN compared to LDAM-DRW or M2m is less apparent in ResNet-32 with Flip and Crop augmentation, compared to the authors' original result with WideResNet-28-10 with SimCLR augmentation. This suggests that the performance improvement from OPeN may be more orthogonal to the improvement caused by a bigger network or more complex data augmentation.

Source	Method	Accuracy
LDAM-DRW [2]	ERM	70.36
	LDAM-DRW	77.03
M2m [3]	ERM	68.7 ± 1.43
	RS	70.4 ± 1.15
	DRS	75.2 ± 0.26
	M2m	78.3 ± 0.16

Source	Method	Accuracy
Ours	ERM	71.70
	RS	70.09
	DRS	75.78
	OPeN	77.52

Table 7. Performance of ResNet-32 models for CIFAR-10-LT (IR=100). Note that for M2m [3], a different dataset variant has been used. Check Section 4.2.2 for more information.

We also perform ablation studies on the effect of DAR-BN on the ResNet architecture and find that DAR-BN improves performance, supporting the central claim by the authors (Table 4).

BN Layer	Accuracy
Standard BN [6]	74.37
Auxiliary BN [12]	75.08
DAR-BN [1]	77.52

Table 8. Batch normalization ablation experiment for OPeN. Same experiment setting as Table 4, but with ResNet-32 and Flip and Crop augmentation.

Finally, we experimented with the ResNet-32 network on a full, balanced CIFAR-10 dataset. Unlike when using WideResNet (Table 6), adding pure noise showed slightly lower performance.

Accuracy without pure noise	Accuracy with pure noise	Change
86.51	86.19	-0.37%

Table 9. Performance of adding pure noise to the full balanced CIFAR-10 dataset when using ResNet-32 model.

Random seed for long-tailed dataset generation — The CIFAR-10-LT dataset is a subset of the CIFAR-10 dataset, so different random seed creates a different dataset. This can be problematic as different papers use different training data, resulting in an unfair comparison of methods.
Cui et al. [8] did not set a random seed but saved their datasets in tfrecords format. Later works implemented their own version of the long-tailed dataset and set a random seed.

We compared the downloaded images from Cui et al. [8] and ran the dataset generation code from Cao et al. [2] and Kim, Jeong, and Shin [3] and discovered that the resulting CIFAR-10-LT datasets have a considerable amount of different training images. On average, each training dataset has around 25% unique images.[1]
To analyze the effect of this discrepancy in the training dataset, we trained the model on each variant. We find that the long-tail subset used to train the model results in a noticeable change in performance. For our work, we use the subset by Cao et al. [2], which gave accuracy scores closest to that reported by Zada, Benou, and Irani [1]. We ask future researchers to specify the long-tail subset they used for reproducibility, and we list the indices of images used for each variant in our code.

Source	ERM	DRS	OPeN
Cui et al. [8]	79.26	80.87	84.19
Kim, Jeong, and Shin [3]	78.37	81.64	87.11
Cao et al. [2]	81.18	83.22	85.04
Reported by Zada, Benou, and Irani [1]	79.6	83.0	84.6

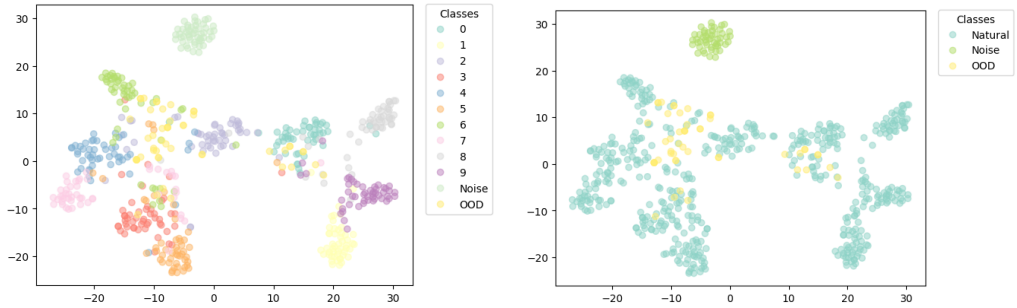
Table 10. Performance of models trained on different CIFAR-10-LT (IR=100) datasets from various sources.

Analysis of model priors — Zada, Benou, and Irani [1] hypothesized that the enhanced performance of OPeN may be due to the shift in model priors. We perform experiments to understand to which degree OPeN influences the model prior. We test three hypotheses:

1. OPeN encourages the model to encode noise and out-of-distribution images similar to minority images.
2. OPeN results in noise and out-of-distribution images being classified as minority images.
3. Model trained with OPeN predicts any image as a minority class more often.

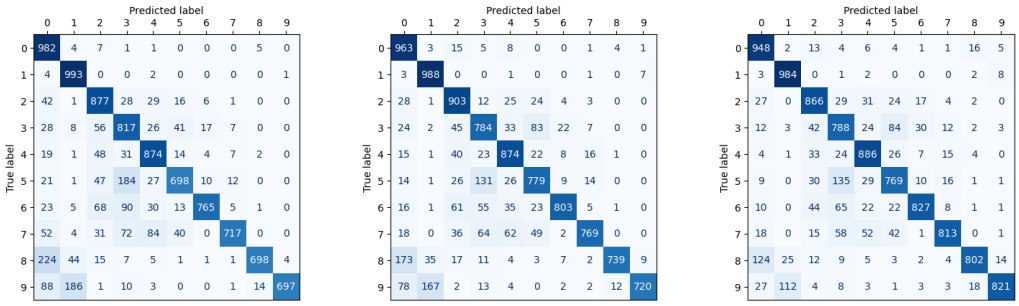
Following Kim, Jeong, and Shin [3], we use t-SNE [19] to visualize the embeddings generated by the network. Embeddings are computed from 50 randomly chosen samples from the validation set of CIFAR-10 using the features from the penultimate layer of the WideResNet network. To represent out-of-distribution (OOD) images, we sample 50 images from one class of the CIFAR-100 validation dataset, as its classes are mutually exclusive to CIFAR-10 [20]. For noise images, we generate 50 new pure noise images. t-SNE is used on these embeddings and is visualized in Figure 1. For both noise and OOD images, we do not see any noticeable proximity to any of the minority classes.
For the second hypothesis, we generate 1000 pure noise images and sample 1000 out-of-distribution images across all 100 classes from CIFAR-100 and pass them through a trained model. We compare the predictions of the ERM, DRS, and the OPeN model. The model trained with OPeN is less likely to predict OOD images as a majority class and more likely to be predicted as a minority class, confirming our hypothesis (Figures 8 and 9 in Appendix).
We note that all pure noise images are classified as classes 2, 4, or 6, whereas the OOD images are more dispersed. As seen in Figure 1, pure noise images are more clustered together, resulting in predictions gathered in a few classes, whereas for out-of-distribution images not seen during training, the predictions are more evenly distributed.
To verify the final hypothesis, we plot the confusion matrix of models trained with ERM, DRS, and OPeN in Figure 2. We find that models trained in OPeN are less likely to have images in the minority classes predicted as one of the majority classes.

[1] We refer the readers to the Appendix for a visualization the intersection of datasets (Figure 3) and for an example of unique images for each dataset (Figure 4).



(a) Each class colored differently **(b)** Only noise and OOD images colored differently

Figure 1. t-SNE of 50 examples from each class in the CIFAR-10 validation dataset, 50 out-of-distribution images from the CIFAR-100 validation dataset, and 50 random pure noise. Images are embedded with a model trained with OPeN. Noise images are gathered into one cluster that is separable from any other class clusters, whereas OOD images are more dispersed across multiple classes. This phenomenon is not unique to OPeN, as it also appears with DRS (Figure 7).



(a) ERM **(b)** Deferred oversampling **(c)** OPeN

Figure 2. Confusion matrices of models trained on CIFAR-10-LT (IR=100).

To conclude, we find that training the model with OPeN makes the model more likely to classify any input image as a minority class. However, this is not done by embedding the noise images to be similar to the minority classes.

5 Discussion

Our experiments support the four claims by Zada, Benou, and Irani [1]. First, our results showed that OPeN improves the mean test accuracy over DRS and other baseline resampling methods across CIFAR-10-LT (IR=100, 50) and CIFAR-100-LT (IR=100, 50) datasets. We confirmed that this improvement is driven by a significant improvement in the accuracy of minority classes. Also, our ablation study supports the claim that using the affine parameters learned from natural activation to normalize the noise activations (DAR-BN) is crucial to the performance of OPeN. Moreover, our experiments showed that OPeN is robust to various data augmentation methods, as OPeN outperforms baseline resampling methods across data augmentations of varying strengths. Finally, our results showed that adding pure noise can be used as an additional data augmentation method to improve the performance on a full, balanced CIFAR-10 dataset.
Then, we ran experiments using a smaller ResNet-32 network and Flip and Crop augmentations to compare with the performance reported by prior works. OPeN still showed improvement over ERM, RS, and DRS, and DAR-BN showed improvement over Standard and Auxiliary BN. However, the comparative advantage of OPeN to preceding papers was less apparent when using a smaller model and simpler data augmentations, which

suggests that performance improvement from OPeN may be more orthogonal to the improvement caused by a bigger network or more complex data augmentation. Also, we noticed that adding pure noise to the balanced CIFAR-10 dataset slightly lowered the performance when using ResNet-32.
Beyond the original paper, we proposed three hypotheses to understand if the enhanced performance of OPeN is due to a shift in model priors. Our analysis shows that OPeN makes the model more likely to classify pure noise, OOD, and misclassified test images as a minority class. However, our visualization suggests that this is not done by encoding the pure noise or OOD images to be similar to the images from minority classes.
Furthermore, our investigation into the preceding papers in imbalanced classification suggests directions to improve the reproducibility of future work in this domain. We found that the images in two instances of the CIFAR-10-LT dataset can vary significantly depending on the random seed used for sampling from the full, balanced CIFAR-10 dataset. Also, prior work used varying mean and standard deviation, which are sometimes computed from the full balanced dataset, for input normalization (Tables 12 and 13). Hence, more detailed documentation for generating the long-tailed dataset and computing the dataset statistics for input normalization may help improve reproducibility and fair comparison across papers.

5.1 What was easy and what was difficult

The authors provided two functions that (i) given a batch, samples noise indices and replaces corresponding natural images with pure noise (ii) given a batch, noise indices, and a BN layer, applies DAR-BN. These functions were clearly documented with docstrings and were easy to use. The authors also clearly documented the key hyperparameters that were used in the main experiments.
Aside from the core functions, the authors' code was not available, so we had to fully implement it based on the description of the paper. Hence, reproducing the reported performance on the first dataset took more time than we initially anticipated, as we had to study available code from previous related papers, including Kim, Jeong, and Shin [3] and Cao et al. [2]. For example, the paper described clipping the sampled noise images to $[0, 1]$, but we did not find a corresponding operation in the provided functions. We found that the InputNormalize module from Kim, Jeong, and Shin [3] had clipping already implemented, so we imported the module and fit it into our training workflow. Following Cao et al. [2], the authors used two different resampling baselines: one baseline started oversampling from the first epoch (Table 1 of [1]), and another baseline started oversampling from the 160th epoch (Figure 3 of [1]). The difference between these two baselines was unclear and needed clarification from the authors. Also, we verified a few available implementations of WideResNet-28-10 [4] and ResNet-32 [21] for correctness.
Yet, comparing prior related works revealed interesting discrepancies as well, such as the differences in the generated long-tailed dataset and input normalization.

5.2 Communication with original authors

Overall, we found the paper to be reproducible, as we were able to validate the effectiveness of OPeN before contacting the authors for clarification. However, we struggled to match the performance of baseline methods. We were able to contact the authors through email to confirm the following details:

- A batch size of 128 was used during training.
- Learning rate warm-up [17] was used for the first 5 epochs.
- Effective number of samples [8] was not used for calculating oversampling weights.
- Oversampling did not increase the number of examples seen per epoch.
- A fixed learning rate of 0.001 was used for training on the full CIFAR-10 dataset.
- A dropout rate of 0.3 was used for WideResNet.

References

1. S. Zada, I. Benou, and M. Irani. "Pure Noise to the Rescue of Insufficient Data: Improving Imbalanced Classification by Training on Random Noise Images." In: **CoRR** abs/2112.08810 (2021). arXiv: 2112.08810.
2. K. Cao, C. Wei, A. Gaidon, N. Aréchiga, and T. Ma. "Learning Imbalanced Datasets with Label-Distribution-Aware Margin Loss." In: **CoRR** abs/1906.07413 (2019). arXiv: 1906.07413.
3. J. Kim, J. Jeong, and J. Shin. "M2m: Imbalanced Classification via Major-to-minor Translation." In: **CoRR** abs/2004.00431 (2020). arXiv: 2004.00431.
4. S. Zagoruyko and N. Komodakis. "Wide Residual Networks." In: **CoRR** abs/1605.07146 (2016). arXiv: 1605.07146.
5. M. Buda, A. Maki, and M. A. Mazurowski. "A systematic study of the class imbalance problem in convolutional neural networks." In: **CoRR** abs/1710.05381 (2017). arXiv: 1710.05381.
6. S. Ioffe and C. Szegedy. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift." In: **CoRR** abs/1502.03167 (2015). arXiv: 1502.03167.
7. Y. Matsubara. "torchdistill: A Modular, Configuration-Driven Framework for Knowledge Distillation." In: **International Workshop on Reproducible Research in Pattern Recognition**. Springer. 2021, pp. 24–44.
8. Y. Cui, M. Jia, T. Lin, Y. Song, and S. J. Belongie. "Class-Balanced Loss Based on Effective Number of Samples." In: **CoRR** abs/1901.05555 (2019). arXiv: 1901.05555.
9. Z. Zhong, J. Cui, S. Liu, and J. Jia. "Improving Calibration for Long-Tailed Recognition." In: **CoRR** abs/2104.00466 (2021). arXiv: 2104.00466.
10. L. Biewald. **Experiment Tracking with Weights and Biases**. Software available from wandb.com. 2020.
11. O. Yadan. **Hydra - A framework for elegantly configuring complex applications**. Github. 2019.
12. C. Xie, M. Tan, B. Gong, J. Wang, A. L. Yuille, and Q. V. Le. "Adversarial Examples Improve Image Recognition." In: **CoRR** abs/1911.09665 (2019). arXiv: 1911.09665.
13. T. Devries and G. W. Taylor. "Improved Regularization of Convolutional Neural Networks with Cutout." In: **CoRR** abs/1708.04552 (2017). arXiv: 1708.04552.
14. T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton. "A Simple Framework for Contrastive Learning of Visual Representations." In: **CoRR** abs/2002.05709 (2020). arXiv: 2002.05709.
15. E. D. Cubuk, B. Zoph, D. Mané, V. Vasudevan, and Q. V. Le. "AutoAugment: Learning Augmentation Policies from Data." In: **CoRR** abs/1805.09501 (2018). arXiv: 1805.09501.
16. I. Azuri and D. Weinshall. "Learning from Small Data Through Sampling an Implicit Conditional Generative Latent Optimization Model." In: **CoRR** abs/2003.14297 (2020). arXiv: 2003.14297.
17. P. Goyal, P. Dollár, R. B. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He. "Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour." In: **CoRR** abs/1706.02677 (2017). arXiv: 1706.02677.
18. Y. Idelbayev. **Proper ResNet Implementation for CIFAR10/CIFAR100 in PyTorch**. https://github.com/akamaster/pytorch_resnet_cifar10. Accessed: 2023-01-16.
19. L. Van der Maaten and G. Hinton. "Visualizing data using t-SNE." In: **Journal of machine learning research** 9.11 (2008).
20. A. Krizhevsky, G. Hinton, et al. "Learning multiple layers of features from tiny images." In: (2009).
21. K. He, X. Zhang, S. Ren, and J. Sun. "Deep Residual Learning for Image Recognition." In: **CoRR** abs/1512.03385 (2015). arXiv: 1512.03385.