Supplementary Material
Tailoring Self-Supervision for Supervised Learning

WonJun Moon, Ji-Hwan Kim, and Jae-Pil Heo*

Sungkyunkwan University

As elaborated in the main paper, our proposed LoRot is a self-supervision task tailored for supervised learning. Our motivation is that self-supervisions previously adopted in supervised learning were originally designed for unsupervised representation learning, thus significant extra computational costs for training were required to achieve insignificant gains. To maximize the benefits for supervised learning, we first introduced three desirable properties of self-supervision and how can pretext tasks can satisfy these conditions by proposing LoRot. To learn rich features, LoRot discovers subdiscriminative features within the part of the image that are not usually considered by current supervised models. Also, LoRot only rotates a part of the image which does not make much changes within the image. Lastly, LoRot is utilized in the form of multi-task learning to have high efficiency. In this supplementary report, we provide ablation and further studies of LoRot, as following outline.

Part 1: Ablation study on the hyperparameter λ
Part 2: Ablation study / analysis on the patch sizes for LoRot-I
Part 3: Ablation study on spatial pooling methods for LoRot-E
Part 4: Detailed results on OOD detection with SupCLR [7]
Part 5: Results on imbalanced classification with the baseline
Part 6: Further study of LoRot on other datasets in OOD detection
Part 7: Implementation details

Throughout this supplementary report, bolds and underlines in tables indicate the best and the second best scores, respectively. Also note that colored references, e.g. Tab., Fig. denotes table and figure in the main paper.

1 Effect of λ in Objective Function

In Tab. 1, we report the performances with varying λ which controls the loss ratio between the primary objective and our self-supervision task. We found that λ does not lead significant performance variations, while $\lambda = 0.2$ usually provides higher scores than the second-best among tested methods in all the tasks. For the classification tasks on CIFAR datasets, we ran experiments under the same setting with Tab. 1 in our main paper.

2 Effect of the Patch Sizes in LoRot-I

To investigate the performance variations of our LoRot-I with respect to the patch sizes, we perform experiments with two configurations, the fixed-sized and

* Corresponding author

Table 1: Classification accuracies (%) on CIFAR datasets and AUROC (%) scores of OOD detection with varying λ. The reported classification and OOD results are averaged over 3 and 5 runs on all the datasets, respectively. Additionally, we report the performances of a fully supervised baseline for comparison. Results demonstrate that LoRot is not very sensitive to hyperparameter that it outperforms the baseline by large margins regardless of the value of λ.

	λ	CIFAR10	CIFAR100	OOD
Baseline	-	95.01	75.07	86.07
LoRot-I	0.1	95.92	76.49	94.55
	0.2	96.16	76.60	95.20
	0.3	95.72	76.57	94.88
	0.4	95.92	75.97	94.98
	0.5	95.84	75.95	94.90
LoRot-E	0.1	95.77	75.9	94.83
	0.2	95.96	76.36	94.83
	0.3	95.75	76.4	94.63
	0.4	95.76	76.13	94.56
	0.5	95.73	76.13	94.55

Table 2: Classification accuracies (%) on CIFAR datasets and AUROC scores (%) for OOD detection with various patch configurations. When the Min and Max patch sizes differs, we randomly sample the patch size within the range, where W denotes the width of the image. For OOD detection, we follow the standard setting used in the paper and report average AUROC scores. Note that, the top row in the table is the reported results of LoRot-I in the paper. All the classification results are averaged over 3 trials, and OOD experiments are averaged over 5 trials.

	Patch Size		Settings		
	Min	Max	CIFAR10	CIFAR100	OOD
Random Size	2	W / 2	96.16	76.60	94.55
	2	W / 4	95.91	77.01	94.51
	W / 4	W / 2	95.95	75.78	95.38
Fixed Size	2	2	94.93	76.25	93.64
	W / 4	W / 4	95.68	76.39	94.16
	W / 2	W / 2	94.45	74.98	95.06

random-sized patches, and the results are reported in Tab. 2. Overall, the random-sized patches outperform the fixed-sized ones. One interesting finding from Tab. 2 is that there is a trade-off between the robustness and accuracy of the model depending on the size of the patch. Specifically, the smaller patches lead high accuracy but lower robustness, and bigger ones bring the opposite tendency. We think that these are mainly because the bigger patches are highly likely to produce

Table 3: Comparison of three different spatial feature pooling methods for LoRot-E. Reported results are the classification accuracies (%) on CIFAR datasets and AUROC scores (%) for OOD. Note that, w_f and h_f indicate the width and height of the feature map, respectively.

Spatial Pooling	Spatial Dim	CIFAR10	CIFAR100	OOD
Dense	$w_f \times h_f$	95.76	74.5	94.20
Reduced Dense	2×2	95.79	76.05	94.70
GAP	1×1	**95.96**	**76.36**	**94.90**

quizzes with regions containing objects that spread out the model's attention, while the smaller patches are more like to work as data augmentation. Indeed, spreading the model's attention is more advantageous for the model's robustness since it forces the model to consider sub-discriminative features. Among various configurations, we choose one that yields a good balance between the accuracy and robustness in the paper.

3 Spatial Pooling Methods for LoRot-E

We basically use a global average pooling (GAP) layer to spatially aggregate the final convolution layer features for the primary and pretext classifiers in LoRot. However, the task of LoRot-E explicitly includes localizing the rotated region within the image, since it requires predicting the index of the rotated quadrant. Therefore, there are more possible choices to aggregate the final convolution layer features for the pretext task to maintain the spatial information.

We investigate three approaches to figure out the proper spatial pooling method: Dense, Reduced Dense, and GAP. First, we can keep all the spatial dimensions of the features, called Dense. This way does not sacrifice any spatial information but requires a bunch of additional parameters for the pretext classifier. Second, we can reduce the spatial dimensions to 2×2, called Reduced Dense. We think 2×2 is the minimum resolution for LoRot-E since it uses a 2×2 grid layout as default. For example, when we have 4×4 feature maps, we can reduce the dimensions into 2×2 by the average pooling. Third, we can collapse the spatial dimensions to 1×1, called GAP, used in the paper. As GAP collapses all the spatial dimensions, the model may not be able to localize the quadrant with rotation. However, we claim that the self-supervision task is still solvable since the features encode the information of the absolute position thanks to the zero paddings [6]. Moreover, it is not necessary to introduce additional parameters for the pretext classifier, which avoids the computational overheads.

To validate the effects of each spatial pooling methods, we report the experimental results of three settings for image classification with CIFAR10/100 and OOD in Tab. 3. We use ResNet50 for classification on CIFAR10/100, and ResNet18 for OOD as did in the paper. All results are averaged over three or five trials for classification and OOD, respectively. Note that, the reported AUROC is the averaged score over all out-distribution datasets described in

Table 4: Full AUROC (%) results of the averaged OOD scores reported in Tab. 7 in the paper. Models are trained with CIFAR10 dataset and evaluated on each out distribution dataset listed in the table. IN denotes ImageNet.

Method	Model	SVHN	LSUN	IN	LSUN (FIX)	IN (FIX)	CIFAR-100	Avg
SupCLR [7]	ResNet50	98.6	97.1	96.2	97.3	97.1	95.6	96.98
+ Rot (MT) [4]	ResNet50	98.2	98.0	97.4	95.7	95.0	93.4	96.28
+ Rot (PT) [4]	ResNet50	98.0	98.2	97.9	96.1	96.3	94.9	96.90
+ LoRot-I	ResNet50	**99.1**	98.6	97.9	**98.0**	**97.7**	**96.4**	**97.95**
+ LoRot-E	ResNet50	**99.1**	**98.9**	**98.4**	97.8	97.3	96.0	97.92

Table 5: Imbalanced classification accuracy (%) on CIFAR10/100. Experiments are conducted with the supervised baseline. The table demonstrates the complementary benefits of LoRot in the data imbalance settings.

	Imbalanced CIFAR10			Imbalanced CIFAR100		
Imbalance Ratio	0.01	0.02	0.05	0.01	0.02	0.05
Baseline	70.36	78.06	83.42	38.32	43.80	51.00
+ Rot (DA)	64.78	70.19	77.41	35.15	38.53	50.99
+ Rot (MT)	66.01	71.75	78.18	35.76	39.08	46.24
+ Rot (PT)	71.75	76.31	83.68	38.91	43.62	50.99
+ LoRot-I	74.79	80.40	85.42	39.42	45.71	53.16
+ LoRot-E	**77.32**	**80.67**	**85.67**	**41.99**	**47.72**	**54.97**

Sec. 7. Interestingly, GAP consistently outperforms Dense and Reduced Dense. We conjecture that the learned features have information for the localization within their channels. Moreover, the additional parameters can cause the overfitting or grant too much weights on the pretext task. As a result, GAP is a proper spatial aggregation method for LoRot-E with higher performances and less computational cost.

4 OOD Detection with Contrastive Learning

In Tab. 4, we report raw individual results of the averaged OOD detection scores shown in Tab. 7 in the main paper. We observe performance gains for all datasets when LoRot used in conjunction with supervised contrastive learning [7]. On the other hand, the original rotation task hardly improves and even degrades the performance of SupCLR in either multi-tasking (MT) or parallel-task learning (PT) strategy. Particularly, although the original rotation task enhances the performance in LSUN- and ImageNet-resize datasets, it shows the slight degradations for other OOD datasets.

5 Imbalanced classification with the baseline

In Tab. 4 of the main paper, we explored complementary benefits of LoRot in the imbalanced image classification with LDAM-DRW [1] under varying imbalanced scenarios. In this supplementary report, we additionally show LoRot's compatibility to the baseline model. As shown in Tab. 5, we observe that LoRot

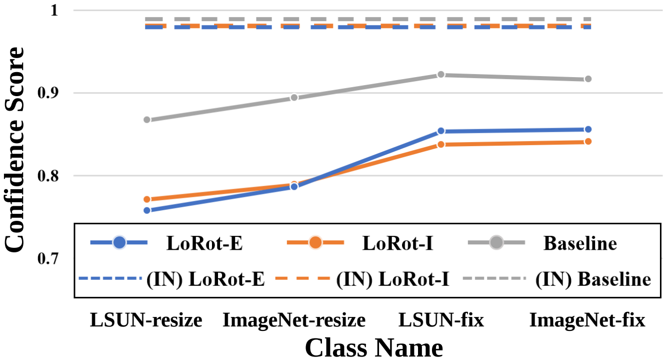

Fig. 1: Dataset-wise averaged confidence scores for in- and out-of-distribution data of the baseline and LoRot. As used in Fig. 5 in the paper, dotted lines are the averaged confidence scores of in-distribution (IN-) dataset (CIFAR10) and solid lines represent the confidence scores (y-axis) for each dataset (x-axis). These results demonstrate that the benefits of LoRot are consistent across datasets.

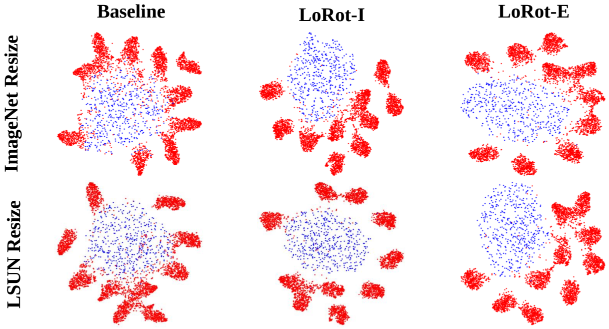

Fig. 2: t-SNE visualization for the baseline, LoRot-I, and LoRot-E on OOD detection benchmark. Dataset for blue dots (OOD dataset) is indicated on the left.

provides complementary benefits with the baseline model. Note that, LoRot-E combined with the baseline even outperforms LDAM-DRW in four out of six scenarios.

6 Further study of LoRot on other datasets in OOD detection

We further show the results on other datasets observing why LoRot is effective in detecting unknown samples. To be specific, we measured the average confidence scores for LSUN and ImageNet datasets in Fig. 1 since their class labels are not available. These results show that LoRot consistently improves the robustness of the classifier by encouraging the model to yield lower confidence scores for unknown OOD datasets. Furthermore, plotted t-SNE in Fig. 2 also implies how better separation between in- and out-distribution datasets are achieved on ImageNet and LSUN dataset. Note that red clusters represent each class in in-distribution dataset (CIFAR10).

7 Implementation Details

OOD Detection. For the OOD detection, we use the ResNet18 architecture as the backbone for a fair comparison against the reported performances in the literature. Therefore, some of the results are reproduced based on their original implementations to unify the backbone network. We deploy the Adam [8] optimizer with a batch size 64, and a learning rate of 0.001. We train the network for 100 epochs and the learning rate is decayed at the middle point of learning by the factor of 0.1. For Rotations [5] and SLA+SD [10], we set the batch size to 128 since it shows the better performances as used in their original papers. As described in the paper Sec. 4.1.1, we use CIFAR-10 as in-distribution data, while SVHN [15], the resized versions of ImageNet and LSUN [11], the fixed versions of ImageNet and LSUN [16], and CIFAR-100 [9] are treated as the out-of-distribution data.

Imbalanced Classification. For a fair comparison, we use the ResNet-32 architecture as the backbone network and follow the settings of the baseline [1]. We set the batch size to 128, and the initial learning rate to 0.1 which is dropped by 0.01 at the 160-th, and 180-th epochs. SGD is used for the optimizer with a momentum of 0.9, weight decay of $2\times\ 10^{-4}$.

Adversarial Perturbations Following previous work [5], we adopted wide ResNet 40-2 [18] architecture as the backbone network. For more details, we utilize SGD optimizer with Nesterov momentum of 0.9 and a batch size of 128. Also, we use an initial learning rate of 0.1 with cosine learning rate schedule [14] and weight decay of $5{\times}10^4$.

Standard Image Classification. For ImageNet classification, we train the model for 300 epochs with a batch size of 256 and initial learning rate of 0.1. During the training, the learning rate is decayed at every 75 epochs with the decaying factor of 0.1. The same implementation details are also applied for Tab. 7, where we describe the complementary benefits to data augmentation methods, e.g., Mixup [19], AutoAugment [2], and RandAugment [3].

For experiments regarding contrastive learning, we follow the protocols from the SupCLR [7] except for the batch size due to lack of GPU memory. Specifically, we train ResNet50 for 1000 epochs with the batch size of 512. The initial learning rate is set to 0.05 and is decayed by cosine decay scheduler. Then, with the learning rate of 5, the classifier is finetuned for evaluation.

Transfer Learning For instance segmentation, we use real-time SOLOv2 [17] model where the number of convolution layers in the prediction head is reduced to two and the input shorter side is 448. We train the model with the 3x schedule as reported in their paper. To train RetinaNet [12], we use the 1x schedule. For both experiments, we test on MS COCO 2017 dataset [13] with the ResNet50 architecture as the backbone network.

References

1. Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. 2019.
2. Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 113–123, 2019.
3. Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 702–703, 2020.
4. Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations*, 2018.
5. Dan Hendrycks, Mantas Mazeika, Saurav Kadavath, and Dawn Song. Using self-supervised learning can improve model robustness and uncertainty. *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.
6. Md Amirul Islam, Sen Jia, and Neil DB Bruce. How much position information do convolutional neural networks encode? *International Conference on Learning Representations, (ICLR)*, 2020.
7. Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. 2020.
8. Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.
9. Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.
10. Hankook Lee, Sung Ju Hwang, and Jinwoo Shin. Self-supervised label augmentation via input transformations. In *International Conference on Machine Learning*, pages 5714–5724. PMLR, 2020.
11. Shiyu Liang, Yixuan Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *International Conference on Learning Representations*, 2018.
12. Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.
13. Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*. Springer, 2014.
14. Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *International Conference on Learning Representations (ICLR)*, 2016.
15. Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.
16. Jihoon Tack, Sangwoo Mo, Jongheon Jeong, and Jinwoo Shin. Csi: Novelty detection via contrastive learning on distributionally shifted instances. In *Advances in Neural Information Processing Systems*, 2020.
17. Xinlong Wang, Rufeng Zhang, Tao Kong, Lei Li, and Chunhua Shen. Solov2: Dynamic and fast instance segmentation. 2020.

18. Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. 2016.
19. Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.