Rethinking the Augmentation Module in Contrastive Learning: Learning Hierarchical Augmentation Invariance with Expanded Views Supplementary Material

A. Augmentation Details

Following SimCLR [1], the probability of color jittering is set to 0.8, with (brightness, contrast, saturation, hue) as (0.4, 0.4, 0.4, 0.1). The probability of converting to grayscale is set to 0.2. The probability of random rotating, flipping, and Gaussian blurring is set to 0.5. The rotation degree is one of $\{0°, 90°, 180°, 270°\}$.

B. Combination with Existing Contrastive Learning Methods

The proposed method can be combined with any contrastive learning methods that fit the general framework. In the main paper, we apply our method to three baseline methods: BYOL [3], SimSiam [4], and Barlow Twins [7]. Here we show some implementation details specific to each method, which are basically the same as the original paper.

BYOL [3] constructs the siamese structure with an online encoder and a target encoder, whose parameters are updated with backpropagation and average momentum, respectively. Thus, the projection heads in the target encoder side added in our method are also updated with average momentum. The decay rate of average momentum is set to 0.99. Besides, BYOL proposes a predictor layer after the online encoder. In this paper, we use the same architecture of the predictor, which has two fully connected (fc) layers and a batch normalization layer applied to its hidden fc layer. The dimension of the predictor's input and output is 2048, and the dimension of the hidden layer is 512.

SimSiam [4] utilizes two share-weights network to construct the siamese structure. It updates the parameters with backpropagation and stop-gradient operation. A predictor is also applied after the encoder, which has the same structure as BYOL. In particular, the learning rate of the predictor in SimSiam is fixed to the initial value and is not decayed in our experiments.

Barlow Twins [7] proposes a new contrastive loss computed with the similarity between the cross-correlation matrix of views and the identity matrix. It has two hyper-parameters: the scale to multiply the on-diagonal loss and the weight to balance the off-diagonal loss. These two hyper-parameters are set to $1/32$ and 3.9×10^{-3}, respectively, following the original paper.

C. Linear Evaluation, Detection, and Segmentation Details

C.1. Linear evaluation

Given the pre-trained backbone, we train a linear classifier on the frozen features from ResNet's global average pooling layer. We train the linear layer for 90 epochs for Imagenet, optimized by momentum SGD with a learning rate of 30 decreased by 0.1 at 60% and 80% of the training schedule. For IN-100, CUB-200, Flower-102, and iNat-2019, we train the linear classifier for 200 epochs with the same optimizer. For Car-196, we train the linear layer with Adam optimizer for 200 epochs with a learning rate of 0.03.

C.2. PASCAL VOC Object Detection

A Faster R-CNN [6] is used with a backbone of R50-C4, which ends with the conv_4 stage, and the box prediction head consists of the conv_5 stage (including global pooling) followed by a BN layer. The image scale is [480, 800] pixels during training and 800 during inference. We fine-tune all layers end-to-end with the mini-batch size of 16 and weight decay of 0.0001. The learning rate is 0.003 and multiplied by 0.1 at 70% and 90% of the training schedule.

C.3. COCO Object Detection and Segmentation

A Mask R-CNN [5] is used with a backbone of R50-C4. The image scale is [640, 800] pixels during training and 800 during inference. We fine-tune all layers end-to-end with the mini-batch size of 16 and weight decay of 0.0001. The learning rate is 0.02 and multiplied by 0.1 at 70% and 90% of the training schedule.

D. More Experiment Details in Discussion

In Section 5, we design two baselines, namely "Uniform" and "Hierarchical strength", and a pretext task to an-

alyze our method. Here we show the details of these experiment settings.

For "Uniform", we also apply several convolution layers to the end of each ResNet stage to extract middle layer features. These features are used to compute multiple contrastive losses as in the main paper. The only difference is that we do not use the add-one strategy to produce the augmentation modules. Specifically, we take the same augmentation pipeline as in other contrastive learning methods to produce multiple pairs of views.

For "Hierarchical strength", each augmentation module has all types of augmentations. However, the following module contains stronger augmentations than the preceding module. The augmentation strength in the last module is the same as in the classical pipeline. We artificially divide each data augmentation into four levels based on strength. Specifically, we change the probability of flipping, converting to grayscale, and blurring to control the augmentation strength. For each level, we decrease the probability by 0.05. For color jittering, we decrease the factor of brightness, contrast, saturation, and hue by 0.02 for each level. For random cropping, we increase the cropping scale by 0.1 for each level.

For the color prediction pretext task in Section 5.2, we evenly divide the color jittering augmentation into ten categories according to its strength. We take the relative distance of augmentation strength applied to the two views to form the prediction label. During training and inference of the pretext task, we take two pairs of view features and concatenate them channel-wise. Then we use a linear classifier to make the prediction.

E. Limitation and Future Works

Although we achieve impressive results on several benchmarks and downstream tasks, the accuracy improvement on the universal datasets (e.g., ImageNet & COCO) is relatively smaller than fine-grained datasets. This reflects that the problems of augmentation types and strength we identified are more severe in fine-grained datasets. A related work [2] studies the transfer ability of the learned representations, which points out that top self-supervised learners fail to preserve color information. We also find that the scope of color invariance and the color embeddings are the most vital factors in our methods. Intuitively, the loss of color information is more crucial for fine-grained datasets, which could explain the limitation mentioned above. Future works could explore the differences in the transfer ability on various downstream tasks.

We make progress in successfully adding random rotating to the augmentation pipeline, which boosts the accuracy of various datasets. However, it still leads to a negative impact on ImageNet, although it is significantly smaller than the baseline as introduced in the main paper. Future works could further fill this gap and explore the possibility of adding more types of augmentations.

Ablation study shows that combining cropping embeddings with color embeddings does not always lead to positive results. Future works could explore a better way to combine multiple augmentation embeddings.

References

[1] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. *ArXiv*, abs/2002.05709, 2020. 1

[2] Linus Ericsson, Henry Gouk, and Timothy M. Hospedales. How well do self-supervised models transfer? *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5410–5419, 2021. 2

[3] Jean-Bastien Grill, Florian Strub, Florent Altch'e, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Ávila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. *NeurIPS*, abs/2006.07733, 2020. 1

[4] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9726–9735, 2020. 1

[5] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask r-cnn. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42:386–397, 2020. 1

[6] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:1137–1149, 2015. 1

[7] Jure Zbontar, Li Jing, Ishan Misra, Yann André LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *ICML*, 2021. 1