Rethinking Rotation in Self-Supervised Contrastive Learning: Adaptive Positive or Negative Data Augmentation

Atsuyuki Miyai[1] Qing Yu[1] Daiki Ikami[2] Go Irie[2] Kiyoharu Aizawa[1]
[1]The University of Tokyo [2]NTT Corporation, Japan
{miyai,yu,aizawa}@hal.t.u-tokyo.ac.jp daiki-ikami@go.tuat.ac.jp goirie@ieee.org

Abstract

Rotation is frequently listed as a candidate for data augmentation in contrastive learning but seldom provides satisfactory improvements. We argue that this is because the rotated image is always treated as either positive or negative. The semantics of an image can be rotation-invariant or rotation-variant, so whether the rotated image is treated as positive or negative should be determined based on the content of the image. Therefore, we propose a novel augmentation strategy, adaptive Positive or Negative Data Augmentation (PNDA), in which an original and its rotated image are a positive pair if they are semantically close and a negative pair if they are semantically different. To achieve PNDA, we first determine whether rotation is positive or negative on an image-by-image basis in an unsupervised way. Then, we apply PNDA to contrastive learning frameworks. Our experiments showed that PNDA improves the performance of contrastive learning. The code is available at `https://github.com/AtsuMiyai/rethinking_rotation`.

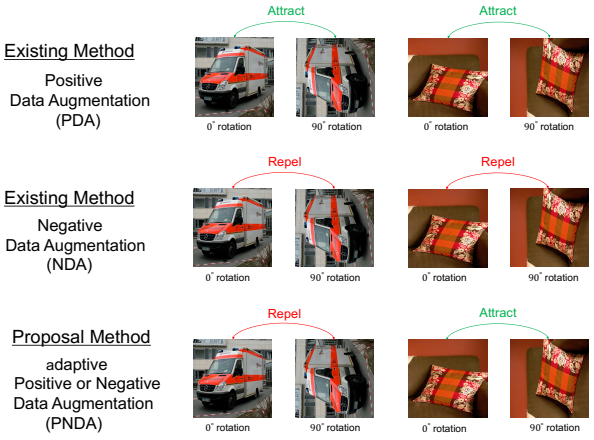


Figure 1: **Comparison of previous and the proposed augmentation strategy. Upper:** PDA treats all rotated images as positives and encourages them to be pulled closer. **middle:** NDA treats all rotated images as negatives and encourages them to be pushed away. **Lower:** Our proposed PNDA considers the semantics of the images, and treats rotation as either positive or negative for each image.

1. Introduction

Recently, self-supervised learning [24, 13, 16, 4, 15] has shown remarkable results in representation learning. The gap between self-supervised and supervised learning has been bridged by contrastive learning [16, 4, 14, 6, 1, 2]. For self-supervised contrastive learning, data augmentation is one of the most important techniques [29]. A common approach for contrastive learning creates positives with some augmentations and encourages them to be pulled closer. Since this augmentation strategy creates positive samples, we refer to it as positive data augmentation (PDA). In addition, some methods [27, 28, 12] use augmentation to create negatives and encourage them to be pushed away. This augmentation strategy is called negative data augmentation (NDA).

Rotation has been attempted to be used for these augmentations, but few improvements have been made. Although rotation is useful in various fields, Chen *et al.* [4] reported that rotation PDA degrades the representation ability in self-supervised contrastive learning because rotation largely affects image semantics. Since then, rotation has been treated as harmful for self-supervised contrastive learning. We consider that this is because previous approaches tried treating rotation as either positive or negative without considering the semantics of each image.

To solve this problem and make full use of rotation, it is important to consider whether the rotation affects the semantics of each image. Natural images are divided into two classes: rotation-agnostic image (RAI) with an ambiguous orientation and non-rotation-agnostic image (non-RAI) with a clear orientation. In RAI, an object can have various orientations. By applying rotation PDA to RAI and

encouraging them to be pulled closer, the image will obtain embedding features robust to rotation. On the other hand, in non-RAI, the orientation of an object is limited. By applying rotation PDA to non-RAI and encouraging them to be pulled closer, the images will lose orientation information and might get undesirable features. For non-RAI, it is preferable to treat rotation as negative to maintain orientation information.

Based on this observation, in this study, we introduce a novel augmentation strategy called adaptive Positive or Negative Data Augmentation (PNDA). In Fig.1, we show an overview of PDA, NDA, and PNDA. While PDA and NDA do not consider the semantics of each image, our proposed PNDA considers the semantics of each image, and treats rotation as positive if the original and rotated images have the same semantics and negative if their semantics are different. To achieve PNDA, we extract RAI for which rotation is treated as positive. However, there is no method to determine whether an image is RAI or non-RAI. Thus, we also tackle a novel task for sampling RAI and propose an entropy-based method. This sampling method focuses on the difference in the difficulty of the rotation prediction between RAI and non-RAI and can extract RAI based on the entropy of the rotation predictor's output.

We evaluate rotation PNDA with contrastive learning frameworks such as MoCo v2 and SimCLR. As a result of several experiments, we showed that the proposed rotation PNDA improves the performance of contrastive learning, while rotation PDA and NDA might decrease it.

The contributions of our paper are summarized as follows:

- We propose a novel augmentation strategy called PNDA that considers the semantics of the images and treats rotation as the better one of either positive or negative for each image.
- We propose a new task of sampling rotation-agnostic images for which rotation is treated as positive.
- We apply rotation PNDA with contrastive learning frameworks, and found that rotation PNDA improves the performance of contrastive learning.

2. Related work

2.1. Contrastive Learning

Contrastive learning has become one of the most successful methods in self-supervised learning [16, 4, 14, 6, 2]. One popular approach for contrastive learning, such as MoCo [16] and SimCLR [4], is to create two views of the same image and attract them while repulsing different images. Many studies have explored the positives or negatives of MoCo and SimCLR [9, 35, 19]. Some methods, such as BYOL [14] or SimSiam [6], use only positives, but recent studies [12, 30] have shown that better representation can be learned by incorporating negatives into these methods. For contrast learning, the use of positives and negatives is important to learn better representations.

2.2. Data Augmentation for Contrastive Learning

There are two main types of augmentation strategies for contrastive learning: positive data augmentation (PDA) and negative data augmentation (NDA).

2.2.1 Positive Data Augmentation (PDA)

Contrastive learning methods create positives with augmentations and get them closer. For example, Chen *et al.* [4] proposed composition of data augmentations *e.g.* Grayscale, Random Resized Cropping, Color Jittering, and Gaussian Blur to make the model robust to these augmentations. On the other hand, they reported that adding rotation to these augmentations degrades the performance. However, they used rotation PDA without considering the difference in the semantic content between RAI and non-RAI.

2.2.2 Negative Data Augmentation (NDA)

Several methods have been proposed to create negative samples by applying specific transformations to images [3, 28, 27]. Sinha *et al.* [27] investigated whether several augmentations, including Cutmix [33] and Mixup [34], which are typically used as positive in supervised learning, can be used as NDA for representation learning. However, they did not argue that rotation NDA is effective. Tack *et al.* [28] stated rotation NDA is effective for unsupervised out-of-distribution detection, but they also did not state that rotation NDA is effective for representation learning. These methods [3, 28, 27] treat the transformed images as negatives without considering the semantics of each image.

2.3. Rotation Invariance

Rotation invariance is one of many good and well-studied properties of visual representation, and many existing methods incorporate rotational invariant features into feature learning frameworks. For supervised learning, G-CNNs [7] and Warped Convolutions [18] showed excellent results in learning rotational invariant features. For self-supervised learning, Feng *et al.* [11] worked on rotation feature learning, which learns a representation that decouples rotation related and unrelated parts. However, previous works separated the rotation related and unrelated parts implicitly as internal information of the network and did not explicitly extract RAI. Here, we tackle a novel task for sampling RAI.

In this paper, we propose a novel augmentation strategy called PNDA that considers the semantics of the image and treats rotation as positive for RAI and negative for non-RAI. To achieve PNDA, we also tackle a novel task for sampling RAI. We demonstrated the effectiveness of rotation PNDA with contrastive learning frameworks with sampled RAI and non-RAI.

3. Rotation-agnostic Image Sampling

To achieve PNDA, we first need to extract RAI for which rotation is treated as positive. In Fig. 2, we show the setting of extracting RAI. We have data of RAI and non-RAI, and our goal here is to extract RAI in an unsupervised way. In this section, we present our novel entropy-based method for sampling RAI. First, we illustrate the overall concept of the method in Section 3.1. Then, in Section 3.2, we detail the training procedure, and in Section 3.3 we explain the inference procedure. Finally, we explain the criterion of tuning the hyperparameters in Section 3.4.

3.1. Overall Concept

This sampling method focuses on the difference in the difficulty of the rotation prediction between RAI and non-RAI. For RAI, the feature distributions of the original and rotated images are similar, so the model can hardly predict which rotation is applied. Thus, the entropy of the rotation predictor's outputs should be large for RAI. On the other hand, for non-RAI, the feature distributions of the original and rotated images are different, so the model can easily predict which rotation is applied. Hence, the entropy of the rotation predictor's outputs should be small for non-RAI. Therefore we can separate RAI and non-RAI by the entropy of the rotation predictor.

We show an overview of our approach in Fig.3. G is a feature generator network, and F is a rotation predictor network. Our idea is to train the rotation predictor F to learn the boundary between RAI and non-RAI. The key is to update the rotation predictor F using only non-RAI and to create a rotation predictor that can correctly predict the rotation of only non-RAI.

3.2. Training Procedure

From the previous discussion in Section 3.1, we propose a training procedure consisting of the following two steps, as shown in Fig. 3.

3.2.1 Step1.

At the first step, we train an initial model with all samples before overfitting. We define the set of transformations $\mathcal{S}$ as all the image rotations by multiples of 90 degrees, i.e., image rotations by 0, 90, 180, and 270 degrees. Namely, we denote $\mathcal{S} := \{S_0, S_{90}, S_{180}, S_{270}\}$. We apply $\mathcal{S}$ to a set of all images. We train the model to predict which transformation $S \in \mathcal{S}$ is applied. As preprocessing, for a given batch of samples $\mathcal{B} = \{x_i\}_{i=1}^{B}$, we apply $\mathcal{S}$ to $\mathcal{B}$. The objective function in this step is as follows.

$$\mathcal{L}_{\mathrm{crs}} = \frac{1}{B} \sum_{S \in \mathcal{S}} \sum_{x_S \in \mathcal{B}_S} -S \log \left(p\left(x_S\right) \right), \ \mathcal{B}_S = \{S(x_i)\}_{i=1}^{B}. \tag{1}$$

$p\left(x_S\right)$ denotes the $|\mathcal{S}|$-dimensional softmax class probabilities for input x_S. We train the model at this step for β_1 epochs.

3.2.2 Step2.

We propose a separation loss to separate RAI from non-RAI. Specifically, we first define the following two losses.

$$\begin{aligned} \mathcal{L}_{\mathrm{es}} &= \frac{1}{B} \sum_{S \in \mathcal{S}} \sum_{x_S \in \mathcal{B}_S} \mathcal{L}_{\mathrm{es}} \left(p\left(x_S\right) \right), \\ \mathcal{L}_{\mathrm{es}}\left(p\right) &= \begin{cases} -|H(p) - \rho| & (|H(p) - \rho| > m), \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \tag{2}$$

$$\begin{aligned} \tilde{\mathcal{L}}_{\mathrm{crs}} &= \frac{1}{B} \sum_{S \in \mathcal{S}} \sum_{x_S \in \mathcal{B}_S} \tilde{\mathcal{L}}_{\mathrm{crs}} \left(p\left(x_S\right) \right), \\ \tilde{\mathcal{L}}_{\mathrm{crs}}\left(p\right) &= \begin{cases} \mathcal{L}_{\mathrm{crs}}\left(p\left(x_S\right)\right) & (H(p) - \rho < -m), \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \tag{3}$$

Eq.(2) is the entropy separation loss proposed by [26]. $H(p)$ is the entropy of p. ρ is set to $\frac{\log(|\mathcal{S}|)}{2}$, since $\log(|\mathcal{S}|)$ is the maximum value of $H(p)$. m is the margin for separation. This loss enables the entropy of RAI to be larger and urges the rotation predictor to misclassify the rotation of RAI, and enables the entropy of non-RAI to be smaller, and urges the rotation predictor to predict the rotation of non-RAI more confidently. The loss in Eq.(3) enables the model to learn using only non-RAI. With a hyperparameter λ, the final objective is as follows.

$$\tilde{\mathcal{L}}_{\mathrm{crs}} + \lambda \mathcal{L}_{\mathrm{es}}. \tag{4}$$

We train the model at this step for β_2 epochs. λ in Eq.(4) is proportional to the epoch number: $\lambda = \lambda' \frac{\mathrm{epoch}}{\beta_2}$, where λ' is a constant number.

3.3. Inference

At inference, we take images with four different rotations as input and calculate the average entropy of the outputs as a score that represents the difficulty of rotation prediction. We treat the images whose score is larger than $\rho + m$ as RAI and the other images as non-RAI. We treat rotation as positive for RAI and as negative for non-RAI.

3.4. Criterion for tuning hyperparameters

The way of tuning the hyperparameters λ' and m focuses on the rotation classification accuracy after Step1 and

Figure 2: **The setting of RAI sampling**. We have the set of RAI and non-RAI images. Our goal is to extract RAI in an unsupervised way.

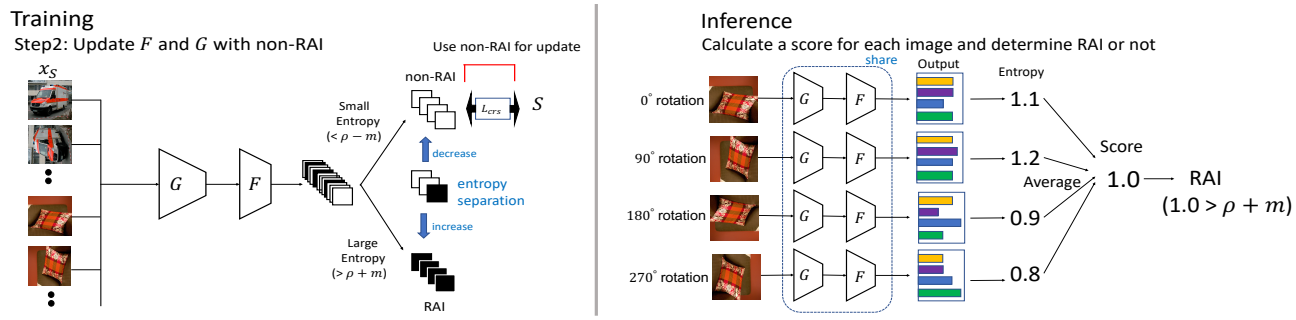


Figure 3: **Overview of training steps and inference step in the proposed sampling method.** G is a feature extractor and F is a rotation predictor. During training, at Step1, we initialize the network with all samples before overfitting. At Step2, we update the network using only non-RAI and make a boundary between non-RAI and RAI. At inference, we calculate the score by averaging the entropy of F's outputs of 4 rotated images, and we determine RAI or not.

Step2. At Step1, we train the rotation predictor before overfitting. At Step2, we train the rotation predictor with non-RAI and separate the entropy of RAI and non-RAI largely and extract RAI. The rotation prediction accuracy of the rotation predictor after Step2 should be almost the same as after Step1 because the number of non-RAI and RAI does not change between Step1 and Step2. We tuned the hyperparameters λ' and m so that the accuracy of the rotation prediction after Step 2 is the same as that after Step 1.

4. PNDA for contrastive learning

In this section, we explain how to apply PNDA to contrastive learning frameworks [4, 5]. We first describe contrastive learning (i.e. the InfoNCE loss) in the context of instance discrimination. Next, we introduce our approach to applying PNDA to contrastive learning.

4.1. Contrastive Learning

InfoNCE loss (i.e. contrastive loss) is commonly used in instance discrimination problems [4, 16]. Given an encoder network f and an image x, we denote the output of the network as $z = f(x)$. We use z_i as the embedding of a sample x_i and use z_p as the embedding of its positive sample x_p. We use $z_n \in N_i$ as embeddings of negative samples. The InfoNCE loss is defined as follows:

$$\mathcal{L}_i^{\text{InfoNCE}} = -\log \frac{\exp({z_i}^\top z_p/\tau)}{\exp({z_i}^\top z_p/\tau) + \sum_{z_n \in N_i} \exp({z_i}^\top z_n/\tau)}, \quad (5)$$

where τ is a temperature parameter.

SimCLR [4] and MoCo v2 [5] create two views of the same image $\hat{x}_i$, ${\hat{x}_i}^+$ with random augmentation aug($\cdot$). Formally, $\hat{x}_i = \text{aug}(x_i)$ and ${\hat{x}_i}^+ = \text{aug}(x_i)$. These two views are fed through the encoder f to obtain embeddings $z_i = f(\hat{x}_i)$ and ${z_i}^+ = f({\hat{x}_i}^+)$. They encourage z_i and ${z_i}^+$ to be pulled closer. That is, when $\hat{x}_i$ is an anchor image, the positive sample is ${\hat{x}_i}^+$. For negative samples, MoCo v2 uses a large dictionary as a queue of negative samples. SimCLR randomly samples a mini-batch of M examples and makes pairs of augmented examples $\hat{X}$ and $\hat{X}^+$. Formally, $\hat{X} = \{\hat{x}_i\}_{i=1}^M$ and $\hat{X}^+ = \{{\hat{x}_i}^+\}_{i=1}^M$. The mini-batch size results in $2M$. For negative samples, SimCLR uses the other $2(M-1)$ augmented examples other than positives within the mini-batch.

4.2. Contrastive Learning with PNDA

Rotation PNDA treats rotated images as positives for RAI and negatives for non-RAI. We define the set of positive samples ${P_i}^r$ and negative samples ${N_i}^r$ which contain rotated images of an anchor x_i. To deal with multiple positive pairs, we use supervised contrastive loss [20]. The ex-

tended InfoNCE loss for PNDA is defined as follows:

$$\mathcal{L}_i^{\mathrm{PNDA}} = -\frac{1}{|P_i{}^r|}\sum_{z_p \in P_i{}^r} \log \frac{\exp(z_i{}^\top z_p/\tau)}{\sum_{z' \in P_i{}^r \cup N_i{}^r} \exp(z_i{}^\top z'/\tau)}. \quad (6)$$

To give a more detailed explanation, we define $\mathrm{Rot}(x, \theta)$ is an image that rotates image x by θ degrees. The detail explanations about $P_i{}^r$ and $N_i{}^r$ for MoCo v2 and SimCLR are provided below:

4.2.1 PNDA for MoCo v2.

For MoCo v2, we refer to [12], which incorporates patch-based NDA into MoCo v2. We extended [12] for rotation PNDA. The set of anchor images for PNDA is the same as vanilla MoCo v2. We set $P_i{}^r$ to $\hat{x}_i{}^+$, $\mathrm{Rot}(\hat{x}_i{}^+, 90)$, $\mathrm{Rot}(\hat{x}_i{}^+, 180)$ and $\mathrm{Rot}(\hat{x}_i{}^+, 270)$ for RAI and $\hat{x}_i{}^+$ for non-RAI. We set $N_i{}^r$ to vanilla MoCo v2's negative samples for RAI and vanilla MoCo v2's negative samples, $\mathrm{Rot}(\hat{x}_i{}^+, 90)$, $\mathrm{Rot}(\hat{x}_i{}^+, 180)$ and $\mathrm{Rot}(\hat{x}_i{}^+, 270)$ for non-RAI.

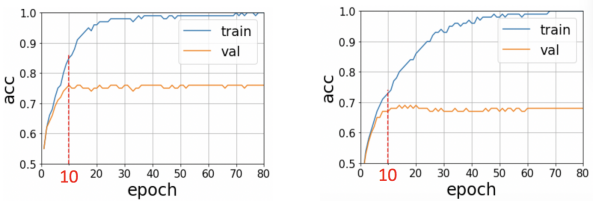
4.2.2 PNDA for SimCLR.

To the best of our knowledge, there is no method that applies NDA to SimCLR for representation learning. We prioritize batch-wise processing, which is the essential mechanism of SimCLR, and the ease of implementation. Like vanilla SimCLR, we create $\hat{X}^+$ and $\hat{X}^+$. In addition, with θ_1 and θ_2 which are different degrees chosen randomly from {90, 180, 270}, we create two sets of rotated images $\hat{X}_{\theta_1}$ and $\hat{X}_{\theta_2}$. Formally, $\hat{X}_{\theta_1} = \{\mathrm{Rot}(\hat{x}_i, \theta_1)\}_{i=1}^M$, $\hat{X}_{\theta_2} = \{\mathrm{Rot}(\hat{x}_i{}^+, \theta_2)\}_{i=1}^M$. The mini-batch size results in $4M$. Like vanilla SimCLR, we use $\hat{X}$ and $\hat{X}^+$ as anchor images. We set $P_i{}^r$ to $\hat{x}_i{}^+$, $\mathrm{Rot}(\hat{x}_i, \theta_1)$ and $\mathrm{Rot}(\hat{x}_i{}^+, \theta_2)$ for RAI and $\hat{x}_i{}^+$ for non-RAI. We set $N_i{}^r$ to the other $4(M-1)$ augmented examples within the mini-batch for RAI and the other $4M-2$ augmented examples, including rotated images of an anchor x_i, within the mini-batch for non-RAI. Note that, although the mini-batch size increases, the diversity of the images in the mini-batch does not change since we increase the data by rotation.

5. Experiments

5.1. Datasets

We use CIFAR-100 [21] and Tiny ImageNet [22], which are used in self-supervised setting [36, 10, 31, 25]. CIFAR-100 contains 50,000 training images and 10,000 test images scaled down to 32×32 in 100 different classes. Tiny ImageNet contains 100,000 training images and 10,000 test images scaled down to 64×64 in 200 different classes, which are drawn from the original 1,000 classes of ImageNet [8].



(a) CIFAR-100 (b) Tiny ImageNet

Figure 4: The train and validation accuracy curves for rotation classification with split-training and split-validation data. We use the epoch just before overfitting to β_1.

5.2. Rotation-agnostic Image Sampling

5.2.1 Implementation Details.

We used ResNet-18 [17] as a feature encoder. Especially, for CIFAR-100, we use extended ResNet used by [4]. They replaced the first convolution layer with the convolutional layer with 64 output channels, the stride size of 1, the kernel size of 3, and the padding size of 3. They removed the first max-pooling from the encoder and added a non-linear projection head to the end of the encoder. In this study, we refer it to ResNet* to distinguish it from ResNet.

To set β_1, we need to know the epoch before overfitting with all training data because self-supervised learning methods use all training data. However, to know the accurate epoch just before overfitting with all training data is impossible due to the lack of validation data. In order to know the approximate epoch just before overfitting, we treat 80% of all training data as split-training data and treat the rest 20% of all training data as split-validation data, and investigate the epoch just before overfitting with split-training data. We set β_1 to the epoch just before overfitting with split-training data, which is close to the epoch with all training data. In Fig. 4, we show the train and validation accuracy curves with split-train and split-validation data for rotation classification. We set β_1 to 10 for both datasets.

We set β_2 to 200 for CIFAR-100 and 150 for Tiny ImageNet. According to Section 3.4, we set λ' to 0.20 for CIFAR-100 and 0.10 for Tiny ImageNet and m to 0.20 for both datasets. We use the Adam optimizer with a learning rate of 0.001 for CIFAR-100 and the Stochastic Gradient Descent (SGD) with a momentum of 0.9 and a learning rate of 0.1 for Tiny ImageNet. We use cosine decay schedule [23]. We train with a batch size of 64 in all experiments. We conducted 3 runs and chose the model that best matched the criterion in Section 3.4. We conducted the training on a single Nvidia V100 GPU.

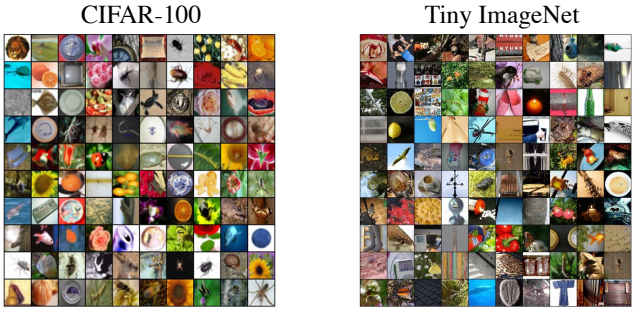
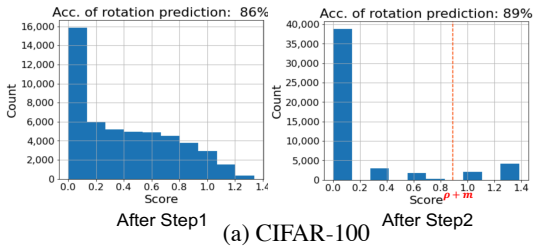


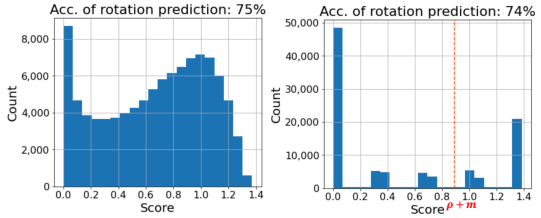
Figure 5: Examples of RAI extracted by our sampling framework on CIFAR-100 and Tiny ImageNet.

Table 1: Number of RAI extracted by our sampling framework on CIFAR-100 and Tiny ImageNet

Dataset	#RAI	#images	ratio of RAI (%)
CIFAR-100	**6,229**	50,000	**12.4**
Tiny ImageNet	**30,711**	100,000	**30.7**



(a) CIFAR-100



(b) Tiny ImageNet

Figure 6: The histograms of the scores obtained with the model after Step1 and Step2 on CIFAR-100 and Tiny ImageNet. As Section 3.3, a score denotes the difficulty of predicting the rotation of an image. These results show that the model after Step2 ensures separation between non-RAI and RAI, whereas the model after Step1 confuses non-RAI and RAI.

5.2.2 Results on Rotation-agnostic Image Sampling.

In Fig. 5, we show some examples of RAI extracted by our sampling framework on CIFAR-100 and Tiny ImageNet. This result shows that our sampling framework can extract RAI approximately correctly. Table 1 shows the number of RAI on CIFAR-100 and Tiny ImageNet. About 12% for CIFAR-100 and about 31% for Tiny ImageNet of all the images are extracted as RAI. Fig.6 shows the histogram of the score of the outputs with the model after Step1 and Step2 on CIFAR-100 and Tiny ImageNet. As described in Section 3.3, a score denotes how difficult the rotation prediction is. Although the accuracies of rotation prediction with both models are almost the same, the distributions of scores are quite different. The model after Step2 makes the difference in the scores between RAI and non-RAI larger and enhances the separation, while the model after Step1 confuses non-RAI and RAI. Note that the number in Table 1 is the number of RAI extracted by our sampling framework and not the actual number of RAI in the datasets. There are no ground truths of RAI and non-RAI, so the exact number of RAI is unknown.

5.3. PNDA for contrastive learning

5.3.1 Compared Methods.

We mainly use MoCo v2 [5] and SimCLR [4] as contrastive learning frameworks. In addition to these baselines, we apply rotation PDA and rotation NDA to these frameworks. Rotation PDA regards all samples as RAI and treats rotated images as positives. Rotation NDA regards all samples as non-RAI and treats rotated images as negatives.

5.3.2 Evaluation Protocols.

Following the previous works [16, 4], we verify our methods by linear classification on frozen features, following a common protocol. After unsupervised pretraining, we freeze the features and train a supervised linear classifier (a fully-connected layer followed by softmax). We train this classifier on the global average pooling features of a ResNet. We report top-1 classification accuracy.

5.3.3 Implementation Details.

We use ResNet-18 and ResNet-50 [17] as our encoder to be consistent with the existing literature [16, 4]. Especially, for CIFAR-100, we use ResNet*. We train models for 300 epochs on CIFAR-100 and for 200 epochs on Tiny ImageNet. We conducted the training on a single Nvidia V100 GPU. A more detailed explanation can be found in the supplementary materials.

5.3.4 Results on PNDA for Contrastive Learning.

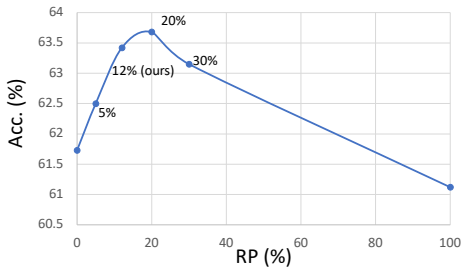
Table 2, 3 show the results of rotation PDA, NDA and PNDA for MoCo v2 and SimCLR. We found that rotation PDA degrades the performance in all experiments. Rotation NDA outperforms the baselines of MoCo v2 and SimCLR

Table 2: Top-1 linear classification accuracies of rotation PDA, NDA, PNDA for MoCo v2 and SimCLR on CIFAR-100. The scores are averaged over 3 trials. RP denotes the ratio of positive rotated images. These results show that PDA and NDA might degrade the performance, but rotation PNDA boosts the performance of contrastive learning.

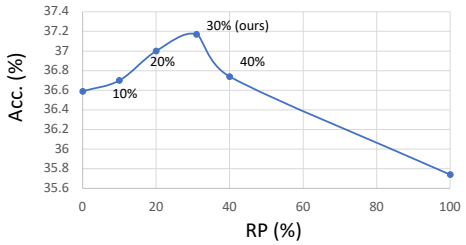
		None	+ PDA	+NDA	+ PNDA (ours)
	RP (%)	-	100	0	12
MoCo v2 [5]	ResNet-18*	62.74±0.37	57.18±0.27↓5.56	62.75±0.29	**63.18±0.22**↑0.44
	ResNet-50*	67.51±0.08	63.36±0.12↓4.15	67.28±0.32	**68.20±0.23**↑0.69
SimCLR [4]	ResNet-18*	62.71±0.38	61.12±0.18↓1.59	61.73±0.23	**63.42±0.04**↑0.71
	ResNet-50*	65.90±0.17	64.46±0.09↓1.44	64.67±0.01	**66.55±0.12**↑0.65

Table 3: Top-1 linear classification accuracies of rotation PDA, NDA, PNDA for MoCo v2 and SimCLR on Tiny ImageNet. The scores are averaged over 3 trials. RP denotes the ratio of positive rotated images. These results show that PDA and NDA might degrade the performance, but rotation PNDA boosts the performance of contrastive learning.

		None	+ PDA	+NDA	+ PNDA (ours)
	RP (%)	-	100	0	31
MoCo v2 [5]	ResNet-18	34.33±0.23	30.76±0.08↓3.57	34.60±0.16	**35.78±0.30**↑1.45
	ResNet-50	38.88±0.40	35.06±0.61↓3.82	38.94±0.51	**39.93±0.47**↑1.05
	ResNet-18*	45.06±0.28	41.42±0.20↓3.64	45.29±0.20	**46.35±0.10**↑1.29
SimCLR [4]	ResNet-18	35.91±0.22	35.74±0.18↓0.17	36.59±0.14	**37.17±0.15**↑1.26
	ResNet-50	40.10±0.30	40.00±0.20↓0.10	41.07±0.13	**41.48±0.24**↑1.38



(a) CIFAR-100



(b) Tiny ImageNet

Figure 7: The effect of the ratio of positive rotated images on CIFAR-100 and Tiny ImageNet. The results show that our sampling method can extract approximately the correct number of RAI images.

in some settings, but the differences between them are not large. However, our proposal PNDA outperforms all comparison methods in all experiments, although PNDA only treats rotation as positive for a few images (12% for CIFAR-100 and 31% for Tiny ImageNet) and negative for the other images.

5.4. Ablation Studies

The ratio of positive rotated images. We examined the effect of the ratio of positive rotated images. 0, 5, 20, 30, and 100% of the images on CIFAR-100 and 0, 10, 20, 40, and 100% of the images on Tiny ImageNet in the descending order of the score are treated as positive rotated images. Then, we use RAI extracted by our sampling framework (12% for CIFAR-100 and 31% for Tiny ImageNet) and compare the accuracies. Fig. 7 shows the results of our experiments with SimCLR using ResNet18*. The experimental results show that the number of RAI extracted by our sampling framework is close to optimal. This result also demonstrates the validity of tuning the hyperparameters of our sampling method in Section 3.4.

The effect of each element of PNDA. We investigated the effectiveness of each element of PNDA. Table 4 shows the comparison results for MoCo v2 with ResNet18* on CIFAR-100. The results show that both the processes of treating RAI's rotated images as positives and non-RAI's rotated images as negatives contribute to the high performance of PNDA. This result indicates the necessity of processing each image separately for RAI and non-RAI.

Table 4: Ablation of each element of PNDA. We use MoCo v2 with ResNet18* on CIFAR-10

Methods	positive for RAI	negative for non-RAI	acc (%)
MoCo v2	-	-	62.74
+ positive	✓	-	<u>62.92</u>
+ positive or negative (PNDA)	✓	✓	**63.18**

Table 5: Top-1 linear classification accuracies of rotation PDA, NDA, PNDA for BYOL. The scores are averaged over 3 trials on CIFAR-100.

		None	+ PDA	+NDA	+ PNDA (ours)
	RP (%)	-	100	0	12
BYOL [14]	ResNet-18*	60.81±0.16	57.11±0.23↓3.70	60.51±0.47	**61.68±0.47** ↑0.87

5.5. PNDA for BYOL

Our PNDA can be applied to contrastive learning frameworks without negatives such as BYOL [14]. Methods, such as BYOL [14], do not rely on negatives. BYOL minimizes their negative cosine similarity between positives. With the embedding feature z_i and z_p in Section 4.1, the loss for BYOL is defined as follows:

$$\mathcal{L}_i^{\mathrm{BYOL}} = \|z_i - z_p\|. \tag{7}$$

For BYOL, we refer to [12], which incorporates patch-based NDA into BYOL. We extended [12] for rotation PNDA. We define the set of rotated positive samples $P_i{}^{r'}$ and rotated negative samples $N_i{}^{r'}$ which are rotated images of an anchor x_i. The extended BYOL loss for PNDA is defined as follows:

$$\mathcal{L}_i^{\mathrm{PNDA}} = \|z_i - z_p\| + \frac{1}{|P_i{}^{r'}|}\sum_{z_{p'}\in P_i{}^{r'}}\|z_i - z_{p'}\| - \frac{\alpha}{|N_i{}^{r'}|}\sum_{z_n\in N_i{}^{r'}}\|z_i - z_n\|, \tag{8}$$

where α is the parameter that controls the penalty on the similarity between the representations of the anchor image and the negative rotated images. We set α to 0.05. We set $P_i{}^{r'}$ as Rot($\hat{x_i}^+$, 90), Rot($\hat{x_i}^+$, 180) and Rot($\hat{x_i}^+$, 270) for RAI and ϕ, which denotes no images, for non-RAI. We set $N_i{}^{r'}$ as ϕ for RAI and Rot($\hat{x_i}^+$, 90), Rot($\hat{x_i}^+$, 180) and Rot($\hat{x_i}^+$, 270) for non-RAI.

Table 5 shows the results for BYOL. We found that our proposal PNDA improves the performance.

6. Discussion

6.1. Limitations

The performance of our proposal PNDA depends on the RAI sampling results. In the previous section, we showed that PNDA boosts the performance of contrastive learning. However, the sampling results could be improved. We extracted RAI by focusing on the difficulty of predicting image rotation, but we cannot consider some issues, such as the background dependencies [32] or the case of multiple objects in an image [8]. Large-scale datasets, such as ImageNet [8], have these issues and require more accurate sampling methods. By developing a more accurate sampling method, the performance of PNDA can still be improved. Addressing such issues is a future challenge.

6.2. Extentions

To the best of our knowledge, this work is the first attempt to determine whether an image is rotation-invariant or rotation-variant. Our method can be generalized to many rotation-based methods, not limited to contrastive learning. Furthermore, in this work, we focus on rotation. In addition, the problem of augment-invariance exists in various augmentations other than rotation. Therefore, it is intriguing to consider generalizing our PNDA to apply to other augmentations.

7. Conclusion

In this paper, we propose a novel augmentation strategy called adaptive Positive or Negative Data Augmentation (PNDA), which treats rotation as the better one of either positive or negative considering the semantics of each image. To achieve PNDA, we tackle a novel task for sampling rotation-agnostic images for which rotation is treated as positive. Our experiments demonstrated that rotation PNDA improves the performance of contrastive learning. PNDA might increase accuracy in augmentation other than rotation, which was previously considered ineffective. We think this perspective will facilitate future work.

Acknowledgement

This work was partially supported by JST JPMJCR22U4 and JSPS KAKENHI 20J22372, Japan.

References

[1] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *NeurIPS*, 2020.

[2] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021.

[3] Chengwei Chen, Yuan Xie, Shaohui Lin, Ruizhi Qiao, Jian Zhou, Xin Tan, Yi Zhang, and Lizhuang Ma. Novelty detection via contrastive learning with negative data augmentation. In *IJCAI*, 2021.

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020.

[5] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[6] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *CVPR*, 2021.

[7] Taco Cohen and Max Welling. Group equivariant convolutional networks. In *ICML*, 2016.

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[9] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. In *ICCV*, 2021.

[10] Aleksandr Ermolov, Aliaksandr Siarohin, Enver Sangineto, and Nicu Sebe. Whitening for self-supervised representation learning. In *ICML*, 2021.

[11] Zeyu Feng, Chang Xu, and Dacheng Tao. Self-supervised representation learning by rotation feature decoupling. In *CVPR*, 2019.

[12] Songwei Ge, Shlok Mishra, Chun-Liang Li, Haohan Wang, and David Jacobs. Robust contrastive learning using negative samples with diminished semantics. In *NeurIPS*, 2021.

[13] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *ICLR*, 2018.

[14] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila 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. In *NeurIPS*, 2020.

[15] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377*, 2021.

[16] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[18] Joao F Henriques and Andrea Vedaldi. Warped convolutions: Efficient invariance to spatial transformations. In *ICML*, 2017.

[19] Tri Huynh, Simon Kornblith, Matthew R. Walter, Michael Maire, and Maryam Khademi. Boosting contrastive self-supervised learning with false negative cancellation. *arXiv preprint arXiv:2011.11765*, 2020.

[20] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In *NeurIPS*, 2020.

[21] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

[22] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. 2015.

[23] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. In *ICLR*, 2017.

[24] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, 2016.

[25] Kento Nozawa and Issei Sato. Understanding negative samples in instance discriminative self-supervised representation learning. In *NeurIPS*, 2021.

[26] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, and Kate Saenko. Universal domain adaptation through self-supervision. In *NeurIPS*, 2020.

[27] Abhishek Sinha, Kumar Ayush, Jiaming Song, Burak Uzkent, Hongxia Jin, and Stefano Ermon. Negative data augmentation. In *ICLR*, 2021.

[28] Jihoon Tack, Sangwoo Mo, Jongheon Jeong, and Jinwoo Shin. Csi: Novelty detection via contrastive learning on distributionally shifted instances. In *NeurIPS*, 2020.

[29] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? In *NeurIPS*, 2020.

[30] Guangrun Wang, Keze Wang, Guangcong Wang, Philip H.S. Torr, and Liang Lin. Solving inefficiency of self-supervised representation learning. In *ICCV*, 2021.

[31] Yifei Wang, Zhengyang Geng, Feng Jiang, Chuming Li, Yisen Wang, Jiansheng Yang, and Zhouchen Lin. Residual relaxation for multi-view representation learning. In *NeurIPS*, 2021.

[32] Kai Xiao, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. Noise or signal: The role of image backgrounds in object recognition. In *ICLR*, 2021.

[33] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Y. Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, 2019.

[34] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018.

[35] Mingkai Zheng, Fei Wang, Shan You, Chen Qian, Changshui Zhang, Xiaogang Wang, and Chang Xu. Weakly supervised contrastive learning. In *ICCV*, 2021.

[36] Mingkai Zheng, Shan You, Fei Wang, Chen Qian, Changshui Zhang, Xiaogang Wang, and Chang Xu. ReSSL: Rela-

tional self-supervised learning with weak augmentation. In *NeurIPS*, 2021.