Enhancing Contrastive Learning Inspired by the Philosophy of "The Blind Men and the Elephant"

Yudong Zhang[1,2], **Ruobing Xie**[2,*], **Jiansheng Chen**[3,*], **Xingwu Sun**[2,4], **Zhanhui Kang**[2], **Yu Wang**[1,*]
[1]Tsinghua University
[2]Tencent
[3]University of Science and Technology Beijing
[4]University of Macau
zhangyd16@mails.tsinghua.edu.cn, xrbsnowing@163.com, jschen@ustb.edu.cn, sunxingwu01@gmail.com, kegokang@tencent.com, yu-wang@mail.tsinghua.edu.cn

Abstract

Contrastive learning is a prevalent technique in self-supervised vision representation learning, typically generating positive pairs by applying two data augmentations to the same image. Designing effective data augmentation strategies is crucial for the success of contrastive learning. Inspired by the story of the blind men and the elephant, we introduce JointCrop and JointBlur. These methods generate more challenging positive pairs by leveraging the joint distribution of the two augmentation parameters, thereby enabling contrastive learning to acquire more effective feature representations. To the best of our knowledge, this is the first effort to explicitly incorporate the joint distribution of two data augmentation parameters into contrastive learning. As a plug-and-play framework without additional computational overhead, JointCrop and JointBlur enhance the performance of SimCLR, BYOL, MoCo v1, MoCo v2, MoCo v3, SimSiam, and Dino baselines with notable improvements.

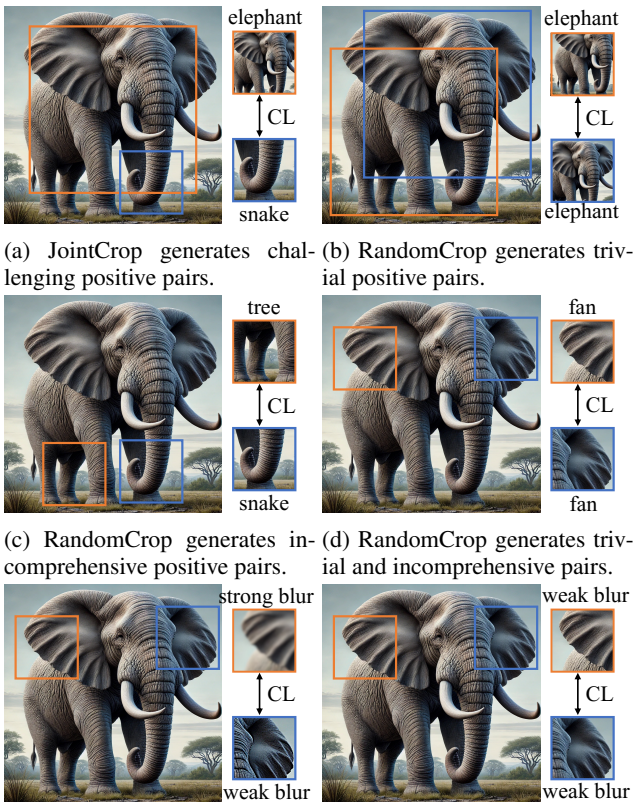
Code — https://github.com/btzyd/JointCrop
Extended version — http://arxiv.org/abs/2412.16522

1 Introduction

Self-supervised learning (SSL) (Caron et al. 2020; Pang et al. 2022) has garnered significant attention in recent years as obtaining large amounts of labeled data is expensive. Contrastive learning, a widely-utilized SSL method, can even outperform supervised learning on tasks such as image classification, object detection, and semantic segmentation (Chen, Xie, and He 2021; Caron et al. 2021).

Contrastive learning (CL) produces self-supervised signals through pretext tasks and utilizes these signals to train the encoder. A common pretext task is instance discrimination (ID) (Wu et al. 2018), which involves a classification problem at the instance level. A pair of positive views is generated by applying two independently distributed data augmentations to a single image, aiming to maximize the similarity of their representations. ID is widely utilized in popular CL methods such as SimCLR (Chen et al. 2020a), MoCo (He et al. 2020; Chen et al. 2020b), BYOL (Grill et al. 2020), SimSiam (Chen and He 2021), SWAV (Caron et al. 2020), and DINO (Caron et al. 2021).

The design of positive pairs is crucial for CL. Previous studies have employed various data augmentations to generate positive pairs, including color distortion, puzzle transformations, and adversarial attacks (Jiang et al. 2020; Kim,



(a) JointCrop generates challenging positive pairs.
(b) RandomCrop generates trivial positive pairs.
(c) RandomCrop generates incomprehensive positive pairs.
(d) RandomCrop generates trivial and incomprehensive pairs.
(e) JointBlur generates challenging positive pairs.
(f) GaussianBlur generates trivial positive pairs.

Figure 1: The motivation of our paper. We use the philosophy of *the blind men and the elephant* to analyze contrastive learning between positive sample pairs.

*Corresponding authors.

Tack, and Hwang 2020; Ho and Nvasconcelos 2020). The InfoMin (Tian et al. 2020) approach reduces the mutual information between positive pairs while preserving task-relevant information to enhance transfer performance. Zhu *et al.* (Zhu et al. 2021) obtain additional positive views via positive extrapolation. ContrastiveCrop (Peng et al. 2022) crops images based on heatmaps to ensure the presence of objects in the cropped views.

Let's begin by revisiting the time-honored story of *the Blind Men and the Elephant* (Saxe 1884). In this tale, six blind men each encounter a different part of an elephant. The man who touches the side believes the elephant is a wall, while those who feel the tusks, trunk, knees, ears, and tail conclude that it resembles a spear, snake, tree, fan, and rope, respectively. Despite their stubbornness and arguments, none have seen the entire elephant. In the context of this story, Contrastive learning enhances the understanding of elephants by decreasing the feature distance between positive pairs, much like how people deepen their understanding of elephants through argumentation.

From the standpoint of difficulty of positive pairs, as Zhu suggests, more challenging positive pairs can help contrastive learning acquire better representations (Zhu et al. 2021). Establishing connections between the whole and the parts of an elephant is more challenging than dealing with incomprehensive or trivial cases. Let's use the elephant example to illustrate this: (1) In Fig. 1b, both blind men perceive the whole elephant. Although they both have a global perception, the task is trivial and not sufficiently challenging. (2) In Fig. 1c, the two blind men perceive an elephant's leg and trunk, mistaking them for a tree and a snake, respectively. Since neither has global information, their interaction results in an incomprehensive and one-sided perception of the elephant, leaving them unaware that it is an elephant at all. (3) In Fig. 1d, both blind men perceive the elephant's ears and mistake them for fans. This interaction is neither constructive nor sufficiently challenging, and it fails to lead to a comprehensive understanding of the elephant. (4) In Fig. 1a, one blind person perceives the whole elephant while the other focuses on the trunk. Through their interaction, they both develop a deeper understanding of the elephant as a whole, as well as the specific details of the trunk. This task is both non-trivial and challenging.

In summary, we argue that CL can benefit from forming more challenging connections between global and local information. However, we have observed that existing CL methods often fall short in generating sufficiently diverse samples, resulting in many samples that are inherently similar or lack comprehensiveness. By analyzing the distribution of area ratios between pairs of positive samples in current CL methods, we found that 80% of RandomCrop pairs have area ratios within 1:2. This leads to a large number of cases similar to those illustrated in Figs. 1b to 1d.

In addition to RandomCrop, we can also consider the data augmentation technique GaussianBlur, which is commonly used in contrastive learning. This technique is akin to putting myopic glasses on the observer; despite the blur, it does not hinder the observer's ability to recognize the elephant or its parts. In Fig. 1e, the combination of normal observations (weak blur) and myopic observations (strong blur) results in challenging samples. Conversely, the positive pairs in Fig. 1f, both employing weak blur, are more trivial.

We refer to the methods we used to generate more challenging positive pairs as JointCrop (Fig. 1a) and JointBlur (Fig. 1e). Previous studies have generated more challenging positive pairs by employing stronger or more diverse data augmentations; however, the two data augmentations applied to positive sample pairs remain independent. In contrast, our proposed methods, JointCrop and JointBlur, **first establish a specific relationship between the positive pairs and then determine the parameters used to generate the two positive samples**. In other words, while previous studies assume that the joint distribution of the two data augmentations is simply the product of their marginal distributions, our approach ensures that the two data augmentations of the positive pairs are interdependent.

Specifically, we intentionally manage a certain metric that effectively measures the difficulty level between positive pairs and use this metric to control the parameters of the two data augmentations. This metric indirectly induces a correlation between the two augmentation parameters, effectively allowing us to control their joint distribution. Consequently, we refer to our methods as JointCrop and JointBlur, which build upon RandomCrop and GaussianBlur, respectively.

Our plug-and-play JointCrop and JointBlur methods are agnostic to CL methods and do not require considerations such as the use of negative samples. Additionally, they incur negligible additional computational overhead during training. As a "free lunch", our JointCrop and JointBlur offer non-trivial improvements over the SimCLR, BYOL, MoCo v1, MoCo v2, MoCo v3, SimSiam, and Dino baselines. Furthermore, our JointCrop and JointBlur can also be used in conjunction with existing techniques for enhancing contrastive learning, such as Multi-Crop and ContrastiveCrop, to further improve the performance of contrastive learning.

The main contributions of this study are summarized as follows: (1) To the best of our knowledge, this is the first study to explicitly introduce the correlation of data augmentations between positive pairs in contrastive learning. (2) We introduce JointCrop and JointBlur, which generate more challenging samples by controlling the distribution of area ratios and GaussianBlur kernels between positive pairs. Additionally, we abstracted a unified framework, JointAugmentation, from both methods, paving the way for this concept to be applied to a broader range of data augmentations. (3) As a "free lunch", our plug-and-play approach incurs no additional computational cost and enhances baselines across various datasets and popular contrastive learning methods.

2 Related Work

2.1 Contrastive Learning

Contrastive learning is a self-supervised learning approach pretrained by pretext tasks with unlabeled data. Previous studies have designed challenging augmented samples to supervise the encoder in learning better feature representations (Bachman, Hjelm, and Buchwalter 2019; Misra and Maaten 2020; Wu et al. 2018; Ye et al. 2019). CL has achieved

strong performance in the case of learning feature representations without labels, and the pretrained models are easy to transfer to downstream tasks such as classification, object detection, and instance segmentation. Contrastive learning achieves strong performance across many tasks (Liang et al. 2024; Feng and Patras 2023; Chanchani and Huang 2023; Xiao et al. 2024; Sarto et al. 2023; Li et al. 2023; Wu, Zhuang, and Chen 2024; Park et al. 2024).

CL can be categorized into two types based on the explicit use of negative samples. The contrastive learning methods that utilize both positive and negative samples include SimCLR (Chen et al. 2020a) and MoCo (He et al. 2020). The core idea of these methods is maximizing the similarity between positive pairs, while minimizing the similarity between non-positive pairs. The CL methods using only positive pairs, such as BYOL (Grill et al. 2020) and SimSiam (Chen and He 2021), uses siamese network structures and feeds pairs of positive views into them. Special designs, such as stop-grad (Chen and He 2021), momentum encoder (He et al. 2020), and a predictor, are necessary to prevent model collapse in the absence of negative samples.

Some studies categorize BYOL and SimSiam as non-contrastive methods. However, to explore the generalizability of our methods, we consider these representational learning approaches as CL methods, as they work by reducing the feature distance between positive pairs.

2.2 Design of Positive Pairs

Regardless of whether negative samples are used, the design of generating pairs of positive views is critical to CL. A popular method to generate a pair of two positive views is applying two data augmentations on a specific image. Several studies have explored the design of positive pairs. SimCLR (Chen et al. 2020a) examines the effectiveness of different combinations of multiple augmentation method and finds that the most useful augmentation methods are Crop and Color. Some studies (Jiang et al. 2020; Kim, Tack, and Hwang 2020; Ho and Nvasconcelos 2020) introduce adversarial attacks and use adversarial examples as positive or negative samples. Several methods have been proposed to craft positive pairs for contrastive learning. For example, ContrastiveCrop (Peng et al. 2022) leverages model heatmaps to guide the cropping region, thereby reducing the likelihood of excluding objects from the cropped area. Similarly, MultiCrop (Caron et al. 2020) replaces a single high-resolution sample with multiple low-resolution crops, thereby improving contrastive learning performance without a substantial increase in computational cost. InfoMin (Tian et al. 2020) finds the sweet spot of mutual information between views and generates positive pairs. However, most of these methods do not explicitly address the questions of whether the two augmentations should be correlated and how they should be correlated. In this work, we propose JointCrop and JointBlur, which introduce the correlation between the augmentation parameters of positive pairs and consider their joint distribution, leading to more challenging views for CL without incurring additional overhead.

3 Method

3.1 Preliminaries

We briefly review the pipeline of CL. For an input image I, CL generates a pair of positive samples by applying the data augmentation $\mathcal{T}$ twice, as shown in Eq. (1), where the cumulative distribution function $F(\boldsymbol{t})$ is the distribution used to sample the augmentation parameters $\boldsymbol{t}$. The views $v_{I,1}$ and $v_{I,2}$ form a pair of positive views, while other views from the images other than I are considered negative samples.

$$\begin{gathered} v_{I,1} = \mathcal{T}\left(I; \boldsymbol{t_1}\right), v_{I,2} = \mathcal{T}\left(I; \boldsymbol{t_2}\right) \\ \boldsymbol{t_1} \sim F(\boldsymbol{t_1}), \boldsymbol{t_2} \sim F(\boldsymbol{t_2}) \end{gathered} \tag{1}$$

In CL methods without negative samples, such as BYOL (Grill et al. 2020) and SimSiam (Chen and He 2021), the representations of $v_{I,1}$ and $v_{I,2}$ are expected to be sufficiently similar. While in SimCLR (Chen et al. 2020a) and MoCo (He et al. 2020), which are CL methods with negative samples, the representations of $v_{I,1}$ and $v_{I,2}$ are expected to be sufficiently close, and their representations are expected to be as distant as possible from the other negative views.

3.2 JointCrop

To introduce our JointCrop method, we first review the RandomCrop pipeline. Initially, the area s is randomly selected within a specified range, defined by $s \sim \mathcal{U}[s_{\text{min}}, s_{\text{max}}]$. Next, the aspect ratio r is also randomly selected. Given the area s and aspect ratio r, we can uniquely determine the width $w = \sqrt{s \times r}$ and height $h = \sqrt{s/r}$. Following this, crop positions i and j are selected using $i \sim \mathcal{U}[0, \mathrm{W} - w]$ and $j \sim \mathcal{U}[0, \mathrm{H} - h]$, where W and H are the image's width and height, respectively. By repeatedly applying this sampling procedure twice to an image I, we obtain a positive pairs $v_{I,1} = \text{Crop}\left(I; \boldsymbol{t_1} = (i_1, j_1, h_1, w_1)\right)$ and $v_{I,2} = \text{Crop}\left(I; \boldsymbol{t_2} = (i_2, j_2, h_2, w_2)\right)$. Based on our previous analysis in Fig. 1, the area ratio $s_r = \frac{s_2}{s_1} = \frac{h_2 w_2}{h_1 w_1}$ between positive pairs can significantly impact contrastive learning performance. To investigate this, we define a quantitative measure of the difficulty of data augmentation, termed Statistical Difficulty Factor (SDF). SDF measures the cosine similarity between all positive pairs generated by a data augmentation method $\mathcal{T}$ across the entire dataset $\mathcal{D}$, using an already trained contrastive learning SimSiam model f.

$$\text{SDF}(\mathcal{T}) = \mathbb{E}_{I \in \mathcal{D}}\left[\cos\left(f\left(v_{I,1}\right), f\left(v_{I,2}\right)\right)\right] \tag{2}$$

We measured the values of SDF for RandomCrop and fixed area ratios ranging from 1:1 to 1:5. Figure 2 confirms our analysis: positive pairs become more challenging as the area ratio increases, while RandomCrop often results in trivial positive pairs. Therefore, we can use the area ratio as a means to control the difficulty level between positive pairs.

To further investigate the distribution of area ratios in the RandomCrop, we aim to find the distribution of $s_r = s_2/s_1$ in RandomCrop. This can be formulated as a mathematical problem: given $s_1 \sim \mathcal{U}[s_{\text{min}}, s_{\text{max}}]$ and $s_2 \sim \mathcal{U}[s_{\text{min}}, s_{\text{max}}]$ (with a typical setup in RandomCrop being $s_{\text{min}} = 0.2$ and $s_{\text{max}} = 1$), we seek the distribution of $s_r = s_2/s_1$, the derivation of which is shown in Appendix H. Since s_r takes

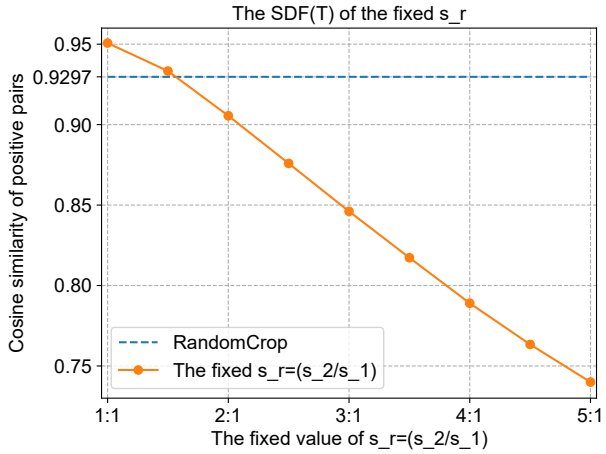


Figure 2: The statistical difficulty between the positive pairs generated by different fixed area ratios $s_r = s_2/s_1$.

values in the range $[0.2, 5]$, and to create symmetry in its probability density map about $s_r = 1$, we plot the probability density map of $\log s_r$ as the green dotted line in Fig. 3. We find that RandomCrop is more likely to produce samples with similar areas; for example, the probability that the area ratios between positive pairs exceed 2:1 is only 18.75%.

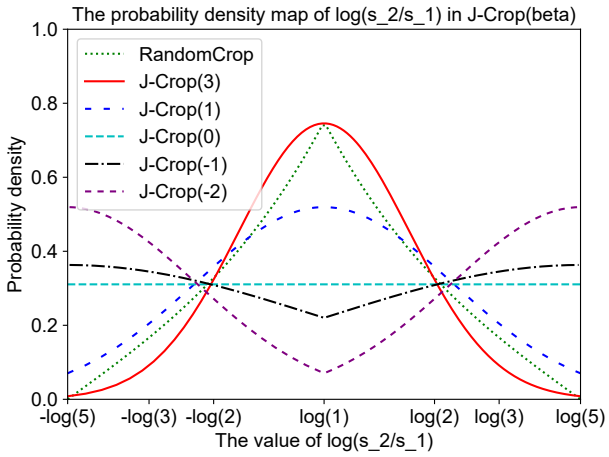


Figure 3: The probability density map of JointCrop, which controls the area ratios of positive pairs obeying a series of distributions JC(β) controlled by β. The smaller β leads to the higher probability that the ratios are far from 1.

Larger area ratios between pairs of two views indicate more challenging samples, which can facilitate learning better representations. However, RandomCrop does not provide a sufficient number of samples with large area ratios. The goal of JointCrop is to make the probability density map of $\log s_r$ "shorter" and "fatter" compared to that of RandomCrop, as depicted by the green dotted line in Fig. 3. A straightforward approach is to control s_r to be as far from 1 as possible before sampling s_1 and s_2. To achieve this, our proposed JointCrop method controls the distribution of $s_r = \frac{s_2}{s_1}$ instead of sampling s_1 and s_2 from independent uniform distributions, as in RandomCrop. In other words, JointCrop manages the joint distribution of s_1 and s_2.

Specifically, we define a series of distributions, JC(β). For each generation of two positive samples from a single image, JointCrop first samples $\log s_r \sim$ JC(β). Then, it samples $s_1 \sim \mathcal{U}\left[\max\left(s_{\min}, \frac{s_{\min}}{s_r}\right), \min\left(\frac{s_{\max}}{s_r}, s_{\max}\right)\right]$. The value of s_2 is then directly calculated as $s_2 = s_1 \times s_r$. Because RandomCrop limits the minimum crop scale to $s_{\min}$ and the maximum crop scale to $s_{\max}$, we ensure that $s_1, s_2 \in [s_{\min}, s_{\max}]$ in JointCrop by controlling the upper and lower bounds of the distribution of s_1. If this sampling yields $s_r > 1$, only the first term of $\max(\cdot,\cdot)$ and $\min(\cdot,\cdot)$ is effective, otherwise the second term is effective.

We define JC(β) in terms of a series of variants of the truncated Gaussian distribution, as shown in Alg. 1 and Fig. 3. The truncated Gaussian distribution $\mathcal{N}_T(\mu, \sigma, p, q)$ has four parameters, μ and σ denote the mean and standard deviation, respectively, while p and q represent the truncated minimum and maximum values. Since $\log s_r$ is symmetric about 0 and takes values in the range $[-s_b, s_b]$, where $s_b = \log \frac{s_{\max}}{s_{\min}}$, we define JC($\beta > 0$) $= \mathcal{N}_T(0, \frac{1}{\beta}s_b, -s_b, s_b)$. We generalize the definition of JC(β) to $\beta = 0$, meaning $\sigma = \frac{1}{\beta}s_b \to \infty$, and in this case, JC(0) approaches a uniform distribution $\mathcal{U}[-s_b, s_b]$. To create more challenging samples than JC(0), we define JC($\beta < 0$) by flipping the left and right halves of JC($|\beta|$) about $-\frac{s_b}{2}$ and $\frac{s_b}{2}$, respectively. The probability density map of JC(β) is depicted in Fig. 3. As β decreases, the distribution of JC(β) becomes "shorter" and "fatter", which implies the generation of more challenging positive pairs.

The process of generating positive pairs using the area ratios $\log s_r \sim$ JC(β) is referred to as J-Crop(β), with the steps detailed in Alg. 1 in Appendix A. We feed the positive pairs generated by J-Crop(β) into a pre-trained SimSiam encoder and measure SDF($\mathcal{T}$) of J-Crop(β). The results, shown in Fig. 4, indicate that smaller β values result in more challenging samples compared to RandomCrop.

We used the samples generated by J-Crop(β) to train SimSiam from scratch for 500 epochs on the Tiny-ImageNet dataset, using ResNet-18 as the backbone. The training losses for different β in J-Crop(β) are shown in Fig. 5. A larger loss indicates more challenging samples, and our JointCrop method indeed provides more challenging samples compared to RandomCrop. The smaller the β, the more challenging the samples are.

3.3 JointBlur

As analyzed in Figs. 1e and 1f, we aim to develop a data augmentation method, JointBlur, that is more challenging than GaussianBlur. Chen *et al.* (Chen et al. 2020a) initially found that using GaussianBlur improves the classification accuracy on ImageNet-1K with ResNet-50 trained for 100 epochs. Subsequent contrastive learning methods have adopted these settings, where the image is blurred with a 50% probability using a Gaussian kernel with a standard de-

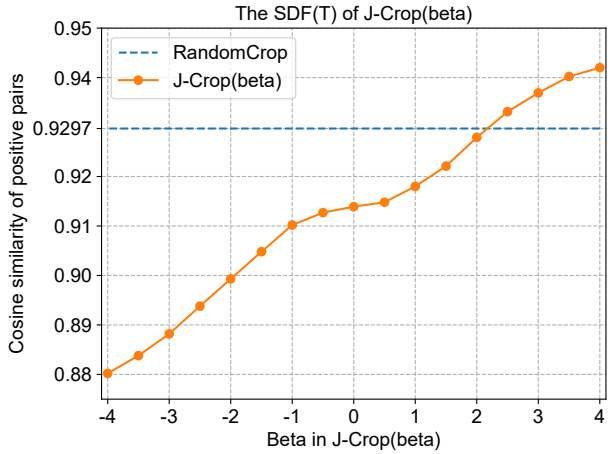


Figure 4: The SDF($\mathcal{T}$) between the positive pairs generated by J-Crop(β) is measured using the already trained SimSiam encoder on the whole ImageNet-1K training dataset.

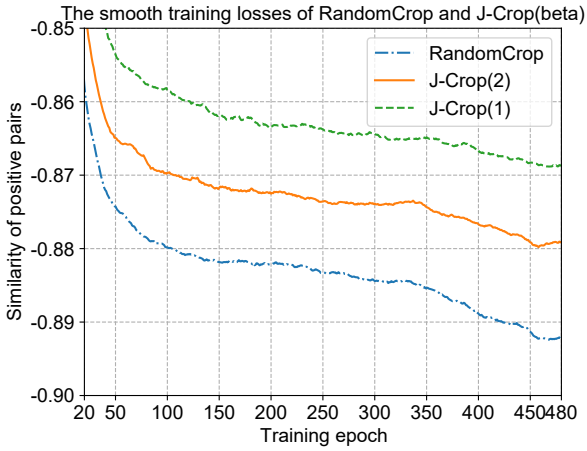


Figure 5: Training losses during SimSiam training on Tiny-ImageNet with samples generated by J-Crop(β). We smooth the losses using a sliding window with a window size of 20. Our JointCrop creates positive pairs that are more challenging than those generated by RandomCrop.

viation $\sigma \sim \mathcal{U}[0.1, 2.0]$, and the kernel size is set to 10% of the image's height and width.

GaussianBlur acts as a low-pass filter that alters the texture information of an image to varying degrees depending on its standard deviation. Our goal is to create more challenging samples for contrastive learning by distinguishing the blur levels more clearly between the two views.

Similar to JointCrop, JointBlur controls the ratio of σ_1 and σ_2, such that $\sigma_1, \sigma_2 \sim \text{JC}(\beta)$ as shown in Alg. 1 and Fig. 3. In this context, we replace s_{min} and s_{max} in Alg. 1 with σ_{min} and σ_{max}, set to default values of 0.1 and 2.0, respectively, which are the lower and upper bounds for the standard deviation of the Gaussian kernel.

3.4 A Unified JointAugmentation Framework

In this section, we aim to abstract the common concept found in JointCrop and JointBlur into a unified framework called JointAugmentation. This approach will facilitate the application of this idea to other data augmentation methods.

In previous studies (He et al. 2020; Chen et al. 2020a; Grill et al. 2020; Chen and He 2021; Caron et al. 2020), positive sample pairs were randomly sampled with data augmentation parameters $\boldsymbol{t_1}$ and $\boldsymbol{t_2}$ independently, meaning $f(\boldsymbol{t_2}|\boldsymbol{t_1}) = f(\boldsymbol{t_2})$, where f represents the probability density function. In other words, when generating $v_{I,2}$, previous contrastive learning work did not utilize the known $\boldsymbol{t_1}$ and instead randomly and independently sampled to obtain $\boldsymbol{t_2}$, as described in Eq. (1). This approach could result in positive pairs that are not sufficiently challenging.

As suggested by Zhu *et al.*, more challenging samples may help CL learn better representations (Zhu et al. 2021). To efficiently generate more challenging samples, JointAugmentation generates positive samples as described in Eq. (3), where G represents the cumulative distribution function.

$$\begin{aligned} v_{I,1} = \mathcal{T}\left(I; \boldsymbol{t_1}\right), v_{I,2} = \mathcal{T}\left(I; \boldsymbol{t_2}\right) \\ \boldsymbol{t_1}, \boldsymbol{t_2} \sim G(\boldsymbol{t_1}, \boldsymbol{t_2}), f(\boldsymbol{t_2}|\boldsymbol{t_1}) \neq f(\boldsymbol{t_2}) \end{aligned} \tag{3}$$

The main difference between previous methods and JointAugmentation lies in whether the known information about $v_{I,1}$ (*i.e.* $\boldsymbol{t_1}$) is considered when generating $v_{I,2}$ (*i.e.* sampling $\boldsymbol{t_2}$). In previous methods, the joint distribution of $\boldsymbol{t_1}$ and $\boldsymbol{t_2}$ is assumed to be the product of their marginal distributions, *i.e.* $f(\boldsymbol{t_1}, \boldsymbol{t_2}) = f(\boldsymbol{t_1}) \cdot f(\boldsymbol{t_2})$. In contrast, JointAugmentation does not follow this assumption, as, $f(\boldsymbol{t_1}, \boldsymbol{t_2}) \neq f(\boldsymbol{t_1}) \cdot f(\boldsymbol{t_2})$. This interdependence in sampling $\boldsymbol{t_2}$ based on $\boldsymbol{t_1}$ is what gives JointAugmentation its name.

In JointCrop, $\boldsymbol{t}$ comprises four parameters: i, j, w, and h. We indirectly control the joint distribution $G(\boldsymbol{t_1}, \boldsymbol{t_2})$ by managing the area ratio $s_r = \frac{s_2}{s_1} = \frac{h_2 w_2}{h_1 w_1}$. For JointBlur, $\boldsymbol{t}$ represents the GaussianBlur kernel, and we similarly manage the joint distribution $G(\boldsymbol{t_1}, \boldsymbol{t_2})$ by controlling the ratio of the GaussianBlur kernel $\frac{\sigma_2}{\sigma_1}$. For other data augmentation methods, we begin with the augmentation parameter $\boldsymbol{t}$ and indirectly control the joint distribution of the two augmentation parameters $G(\boldsymbol{t_1}, \boldsymbol{t_2})$ by regulating certain aspects related to difficulty.

4 Experiments

4.1 Experiment Settings

Datasets. We conduct experiments on the STL-10 (Coates, Ng, and Lee 2011), Tiny-ImageNet, and ImageNet-1K (Deng et al. 2009). We also evaluate the performance of downstream tasks on PASCAL VOC (Everingham et al. 2010) and COCO (Lin et al. 2014). For specific ablation experiments, we utilize the ImageNet-100, which is created by randomly selecting 100 classes from ImageNet-1K.

Models. We perform experiments on several recent and popular CL methods, including SimCLR (Chen et al. 2020a), MoCo (He et al. 2020; Chen et al. 2020b; Chen, Xie, and He 2021), SimSiam (Chen and He 2021), BYOL (Grill et al. 2020), and Dino (Caron et al. 2021). For all these methods,

we use ResNet as the backbone. The experimental setup remains consistent with the baseline, except for the modifications introduced by our JointCrop and JointBlur methods.

Strategies for Selecting Hyperparameters β. As a plug-and-play method, we aim to use a common hyperparameter β rather than having different hyperparameter choices for various situations. Therefore, unless specifically mentioned for ablation experiments, we set $\beta = 0$ for both JointCrop and JointBlur. While carefully adjusting hyperparameters could potentially enhance performance, this is beyond the scope of the current work.

Pretraining on Small Datasets. On small datasets such as STL-10 and Tiny-ImageNet, we use ResNet-18 as the backbone and train for 500 epochs with a batch size of 512 and a cosine annealing learning rate of 0.5.

Pretraining on ImageNet-1K. On ImageNet-1K, we use ResNet-50 as the backbone. For each baseline, we adhered to the experimental setup as described in the original papers.

Linear Evaluation. We evaluate the performance of the pre-trained model using linear evaluation. Specifically, we assess the top-1 accuracy of a linear classifier on the validation sets to gauge the model's performance. For linear classification on smaller datasets, such as STL-10 and Tiny-ImageNet, we use a small initial learning rate of 10. For linear classification on ImageNet-1K, we adhere to the evaluation settings outlined in the original papers.

4.2 Results of JointCrop and JointBlur

Results on ImageNet-1K. We applied JointBlur and JointCrop separately to the data augmentation process for ImageNet, and the results are presented in Tab. 1. The consistent and non-trivial improvements observed demonstrate the effectiveness of our JointCrop and JointBlur methods. We also provide results for MoCo v3 pre-trained for 300 epochs. Even under long-term training, our JointCrop continues to show significant improvements.

Model	Batch Size	Epoch	Baseline	Baseline +J-Crop(0)	Baseline +J-Blur(0)
SimCLR	512	100	60.7	**62.16**(+1.46)	61.40(+0.70)
SimSiam	256	100	68.1	**68.51**(+0.41)	68.31(+0.21)
MoCo v1	256	200	60.6	**63.29**(+2.69)	62.87(+2.27)
MoCo v2	256	200	67.5	67.70(+0.20)	**67.87**(+0.37)
MoCo v3	4096	100	68.9	**69.47**(+0.57)	-
		300	72.8	**73.23**(+0.43)	-

Table 1: Linear classification results of JointCrop and JointBlur on ImageNet-1K. All baseline results were sourced from their papers. Since MoCo v3 uses Gaussian Blur only with 10% probability when generating $v_{I,2}$, JointBlur does not have a significant impact on MoCo v3, as marked by "-".

Results on Small Datasets. The results of the baselines and JointCrop on small datasets are shown in Tab. 2. The results show that our proposed JointCrop can consistently improve the performance on small datasets. Due to the low resolution of images in smaller datasets, their original data augmentations do not include GaussianBlur. Consequently, using JointBlur on these small datasets is not suitable.

Dataset	Method	SimCLR	BYOL	MoCo v2	SimSiam
STL-10	Baseline	89.40	91.71	88.11	88.74
	+J-Crop(0)	**90.20**(+0.80)	**92.28**(+0.57)	**89.78**(+1.67)	**89.08**(+0.34)
Tiny-IN	Baseline	45.25	48.91	46.07	44.17
	+J-Crop(0)	**47.53**(+2.28)	**49.91**(+1.00)	**48.45**(+2.38)	**45.73**(+1.56)

Table 2: Linear classification results on small datasets. Our JointCrop consistently provides non-trivial improvements.

4.3 Results of Downstream Tasks

Object Detection on PASCAL VOC. Our experimental settings are the same as MoCo v1, that is, the detector is Faster R-CNN (Ren et al. 2015) with a backbone of R50-C4 (He et al. 2017) with 200 epochs of pre-training. We fine-tune the pretrained model with all layers end-to-end for 24K iterations using the detectron2 codebase (Wu et al. 2019) on the `trainval2007+2012` split and evaluate on `test2007`. Our method achieves improvements of $+0.8$AP and $+0.2$AP over MoCo v1 and MoCo v2 baselines, as shown in Tab. 3.

Object Detection and Instance Segmentation on COCO. We use Mask R-CNN (He et al. 2017) with a R50-C4 backbone for object detection and instance segmentation on COCO. We fine-tune all layers end-to-end for 90K iterations, that is, $1\times$ schedule on `train2017` and evaluate on `val2017`. As a result, our proposed JointCrop achieves improvements of $+0.4$AP and $+0.2$AP compared with MoCo v1 and MoCo v2 baselines, as shown in Tab. 4.

4.4 Ablation Studies of Hyper-Parameter

Our proposed JointCrop method controls the statistical difficulty of positive pairs by adjusting β in J-Crop(β). We investigate the influence of β in J-Crop(β) on Tiny-ImageNet, as shown in Tab. 5. A smaller β results in more challenging samples. Table 5 reveals an optimal point, β_{op}. Values of $\beta > \beta_{op}$ lead to oversimplified samples, while values less than $\beta < \beta_{op}$ produce samples that are too challenging, hindering the learning of better representations. As a plug-and-play framework, JointCrop is designed to be insensitive to hyperparameters and should not require a complex hyperparameter selection strategy. Across a broad range of $\beta \in [-2, 2]$, JointCrop consistently enhances the baseline. The best results, corresponding to specific β values, are highlighted in bold. Our generic hyperparameter, $\beta = 0$, is marked with yellow cells.

4.5 Generalization to Other Datasets and Augmentation Methods

Pre-training on Non-Object-Centered and Multi-Object Datasets. Our pre-training primarily utilizes the ImageNet-1K dataset, which is characterized by being single-object and object-centered. However, existing pre-training approaches often use images sourced from the web, which do not necessarily share these properties. This raises the question of whether our approach can be effectively applied to a broader range of contrastive learning pre-training scenarios

Method	AP	AP_{50}	AP_{75}
MoCo v1 Baseline	55.9	81.5	62.6
MoCo v1+J-Crop(0)	**56.7**	**81.9**	**63.3**
MoCo v2 Baseline	57.0	82.4	63.6
MoCo v2+J-Crop(0)	**57.2**	**82.5**	**63.9**

Table 3: Results of transferring to PASCAL VOC.

Method	COCO instance seg.			COCO detection		
	AP^{mk}	AP^{mk}_{50}	AP^{mk}_{75}	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}
MoCo v1 Baseline	33.6	54.8	35.6	38.5	58.3	41.6
MoCo v1+J-Crop(0)	**34.0**	**55.3**	**36.1**	**38.9**	**58.5**	**42.2**
MoCo v2 Baseline	34.0	55.4	36.0	39.0	58.5	42.3
MoCo v2+J-Crop(0)	**34.2**	**55.5**	**36.4**	**39.2**	**58.8**	**42.6**

Table 4: Results of transferring to COCO detection and segmentation. We fine-tune 90K iterations, *i.e.*, $1\times$ schedule on `train2017` and evaluate on `val2017`.

that involve non-object-centered and multi-object images. To explore this, we utilized the COCO dataset, which is non-object-centered and contains multiple objects, for pre-training. We then fine-tuned the model on ImageNet-100. As shown in Table 6, even when pre-training on non-object-centered and multi-object datasets, our JointCrop method demonstrates a significant improvement over the baseline.

Generalizing JointAugmentation to other data augmentations. In Sec. 3.4, we abstract the ideas of JointCrop and JointBlur into a unified framework, JointAugmentation, which we attempt to generalize to other popular data augmentation approaches. ColorJitter performs random distortion of the image's color, with brightness b and contrast c being two key parameters. For example, brightness is sampled as $b \sim \mathcal{U}[1-b_f, 1+b_f]$, where $b_f \in [0,1]$ and defaults to $b_f = 0.4$. The baseline ColorJitter independently performs the same operation twice to obtain b_1, c_1 and b_2, c_2. Similar to JointCrop, JointColor controls the joint distribution of b_1 and b_2 (c_1 and c_2) through the ratios b_1/b_2 (c_1/c_2). The non-trivial improvements presented in Tab. 7 further demonstrate the generalizability of our JointAugmentation framework to other data augmentation methods.

4.6 Potential Combinations With Other Methods

In-depth analysis of JointCrop and MultiCrop. JointCrop and MultiCrop have different motivations and methods. They can be used in combination leading to better representations, as in Appendix D.

Combined use of JointCrop and JointBlur. The combination of JointCrop and JointBlur requires more fine-grained considerations, otherwise it may produce overly difficult samples, as in Appendix F.

Combined use of JointCrop and InfoMin. The combination of JointCrop with InfoMin further improves the InfoMin baseline from 67.4 to 67.81.

In-depth analysis of JointCrop and ContrastiveCrop. We provide an in-depth analysis for ContrastiveCrop and our JointCrop in Appendix G, and try to combine the two.

Method	SimCLR	BYOL	MoCo v2	SimSiam
Baseline	45.25	48.91	46.07	44.17
+J-Crop(2)	46.69	50.19	46.65	45.45
+J-Crop(1)	47.77	49.98	47.81	45.10
+J-Crop(0)	47.53	49.91	48.45	45.73
+J-Crop(-1)	47.29	49.54	48.66	**45.91**
+J-Crop(-2)	**47.95**	**51.02**	**48.78**	45.62

Table 5: Linear evaluation accuracy on Tiny-ImageNet w.r.t. J-Crop(β). The yellow cells indicate the common hyperparameters $\beta = 0$. Within the range of $\beta \in [-2, 2]$, JointCrop consistently improves the baseline, which demonstrates the plug-and-play JointCrop is not sensitive to hyperparameters.

Method	MoCo v1	+J-Crop(0)	MoCo v2	+J-Crop(0)
Accuracy	59.62	61.92	56.12	57.92

Table 6: Results of IN-100 when pre-training on COCO.

Method	Baseline	JointColor-b	JointColor-c
MoCo v1	63.18	63.80	63.68

Table 7: Results of JointColor on ImageNet-100.

Analysis of positive pair distances. JointCrop actually controls not only the area ratio, but also implicitly controls the distance between positive samples, as in Appendix E.

4.7 Computational Complexity Analysis

We train 5 epochs for the MoCo v1 baseline, JointCrop, and JointBlur on and calculated the average running time. Table 8 illustrates our JointCrop and JointBlur introduce little additional computational complexity. This does not indicate our method is faster, since the time differences are minor.

Method	MoCo v1	+JointCrop	+JointBlur
Time (seconds)	788.2($\pm$2.2)	782.8($\pm$1.8)	785.8($\pm$1.8)

Table 8: Training time of baseline and our methods.

5 Conclusion

In this work, we propose JointCrop and JointBlur, which explore the correlation between two augmentations of positive pairs and generates more challenging positive pairs by controlling the joint distribution of these augmentation parameters. We also integrated both approaches into a unified framework called JointAugmentation, paving the way for applying this concept to other forms of data enhancement. The effectiveness of our method has been demonstrated across multiple popular contrastive learning methods. We hope our work will inspire further research on data augmentations in contrastive learning.

Our limitations and future work are in Appendix C.

Acknowledgements

The authors would like to thank Xuefei Ning, Cheng Yu, Youze Xue and Yu Shang for their help in revising the paper.

This work was supported by the National Natural Science Foundation of China (62376024, 62325405), the Young Elite Scientists Sponsorship Program by CAST (2023QNRC001) and Beijing National Research Center for Information Science and Technology (BNRist, BNR2024TD03001).

References

Bachman, P.; Hjelm, R. D.; and Buchwalter, W. 2019. Learning representations by maximizing mutual information across views. *NeurIPS*, 32: 15535–15545.

Caron, M.; Misra, I.; Mairal, J.; Goyal, P.; Bojanowski, P.; and Joulin, A. 2020. Unsupervised learning of visual features by contrasting cluster assignments. *NeurIPS*, 33: 9912–9924.

Caron, M.; Touvron, H.; Misra, I.; Jégou, H.; Mairal, J.; Bojanowski, P.; and Joulin, A. 2021. Emerging properties in self-supervised vision transformers. In *ICCV*, 9650–9660.

Chanchani, S.; and Huang, R. 2023. Composition-contrastive Learning for Sentence Embeddings. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 15836–15848.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020a. A simple framework for contrastive learning of visual representations. In *ICML*, 1597–1607. PMLR.

Chen, X.; Fan, H.; Girshick, R.; and He, K. 2020b. Improved baselines with momentum contrastive learning. arXiv:2003.04297.

Chen, X.; and He, K. 2021. Exploring simple siamese representation learning. In *CVPR*, 15750–15758.

Chen, X.; Xie, S.; and He, K. 2021. An empirical study of training self-supervised vision transformers. In *ICCV*, 9640–9649.

Coates, A.; Ng, A.; and Lee, H. 2011. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 215–223. JMLR Workshop and Conference Proceedings.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *CVPR*, 248–255.

Everingham, M.; Van Gool, L.; Williams, C. K.; Winn, J.; and Zisserman, A. 2010. The pascal visual object classes (voc) challenge. *IJCV*, 88(2): 303–338.

Feng, C.; and Patras, I. 2023. MaskCon: Masked contrastive learning for coarse-labelled dataset. In *CVPR*, 19913–19922.

Grill, J.-B.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P.; Buchatskaya, E.; Doersch, C.; Avila Pires, B.; Guo, Z.; Gheshlaghi Azar, M.; et al. 2020. Bootstrap your own latent-a new approach to self-supervised learning. *NeurIPS*, 33: 21271–21284.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 9729–9738.

He, K.; Gkioxari, G.; Dollár, P.; and Girshick, R. 2017. Mask r-cnn. In *ICCV*, 2961–2969.

Ho, C.-H.; and Nvasconcelos, N. 2020. Contrastive learning with adversarial examples. *NeurIPS*, 33: 17081–17093.

Jiang, Z.; Chen, T.; Chen, T.; and Wang, Z. 2020. Robust pre-training by adversarial contrastive learning. *NeurIPS*, 33: 16199–16210.

Kim, M.; Tack, J.; and Hwang, S. J. 2020. Adversarial self-supervised contrastive learning. *NeurIPS*, 33: 2983–2994.

Li, Q.; Joty, S.; Wang, D.; Feng, S.; Zhang, Y.; and Qin, C. 2023. Contrastive learning with generated representations for inductive knowledge graph embedding. In *Findings of the Association for Computational Linguistics: ACL 2023*, 14273–14287.

Liang, P. P.; Deng, Z.; Ma, M. Q.; Zou, J. Y.; Morency, L.-P.; and Salakhutdinov, R. 2024. Factorized contrastive learning: Going beyond multi-view redundancy. *Advances in Neural Information Processing Systems*, 36.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *ECCV*, 740–755. Springer.

Misra, I.; and Maaten, L. v. d. 2020. Self-supervised learning of pretext-invariant representations. In *CVPR*, 6707–6717.

Pang, B.; Zhang, Y.; Li, Y.; Cai, J.; and Lu, C. 2022. Unsupervised Visual Representation Learning by Synchronous Momentum Grouping. In *ECCV*, 265–282. Springer.

Park, J.; Gwak, D.; Choo, J.; and Choi, E. 2024. Self-Supervised Contrastive Learning for Long-term Forecasting. In *ICLR*.

Peng, X.; Wang, K.; Zhu, Z.; Wang, M.; and You, Y. 2022. Crafting better contrastive views for siamese representation learning. In *CVPR*, 16031–16040.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *NeurIPS*, 28.

Sarto, S.; Barraco, M.; Cornia, M.; Baraldi, L.; and Cucchiara, R. 2023. Positive-augmented contrastive learning for image and video captioning evaluation. In *CVPR*, 6914–6924.

Saxe, J. G. 1884. *The Poems of John Godfrey Saxe*. Houghton, Mifflin.

Tian, Y.; Sun, C.; Poole, B.; Krishnan, D.; Schmid, C.; and Isola, P. 2020. What makes for good views for contrastive learning? *NeurIPS*, 33: 6827–6839.

Wu, L.; Zhuang, J.; and Chen, H. 2024. Voco: A simple-yet-effective volume contrastive learning framework for 3d medical image analysis. In *CVPR*, 22873–22882.

Wu, Y.; Kirillov, A.; Massa, F.; Lo, W.-Y.; and Girshick, R. 2019. Detectron2. https://github.com/facebookresearch/detectron2. Accessed: 2024-12-21.

Wu, Z.; Xiong, Y.; Yu, S. X.; and Lin, D. 2018. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, 3733–3742.

Xiao, T.; Zhu, H.; Chen, Z.; and Wang, S. 2024. Simple and asymmetric graph contrastive learning without augmentations. *NeurIPS*, 36.

Ye, M.; Zhang, X.; Yuen, P. C.; and Chang, S.-F. 2019. Unsupervised embedding learning via invariant and spreading instance feature. In *CVPR*, 6210–6219.

Zhu, R.; Zhao, B.; Liu, J.; Sun, Z.; and Chen, C. W. 2021. Improving contrastive learning by visualizing feature transformation. In *ICCV*, 10306–10315.