Feature Augmentation for Self-supervised Contrastive Learning: A Closer Look

Yong Zhang^{†, §}, Rui Zhu[‡], Shifeng Zhang[§], Xu Zhou[§], Shifeng Chen[†], and Xiaofan Chen^{§, ⊠}

†Shenzhen Institute of Advanced Technology, CAS [‡]The Chinese University of Hong Kong, Shenzhen [§]Sangfor Technologies Inc.

{yongzhang, ruizhu}@link.cuhk.edu.cn, {zhangshifeng, zhouxu, chenxiaofan}@sangfor.com.cn, shifeng.chen@siat.ac.cn

Abstract—Self-supervised contrastive learning heavily relies on the view variance brought by data augmentation, so that it can learn a view-invariant pre-trained representation. Beyond increasing the view variance for contrast, this work focuses on improving the diversity of training data, to improve the generalization and robustness of the pre-trained models. To this end, we propose a unified framework to conduct data augmentation in the feature space, known as feature augmentation. This strategy is domain-agnostic, which augments similar features to the original ones and thus improves the data diversity. We perform a systematic investigation of various feature augmentation architectures, the gradient-flow skill, and the relationship between feature augmentation and traditional data augmentation. Our study reveals some practical principles for feature augmentation in self-contrastive learning. By integrating feature augmentation on the instance discrimination or the instance similarity paradigm, we consistently improve the performance of pre-trained feature learning and gain better generalization over the downstream image classification and object detection task.

Index Terms—feature augmentation, self-supervised contrastive learning

I. Introduction

Data augmentation (DA) has become a widely employed regularization strategy for model generalization in supervised learning, which is essential especially when encountering insufficient training data. Though numerous unlabeled data are usually available in the self-supervised learning setting, data augmentation also plays an important role in some self-supervised contrastive learning paradigms [1]–[6]. For example, as shown in Fig. 1, given an image, data augmentation is leveraged to generate various views (i.e., a particular feature group in multi-view learning [7]) to form contrastive pairs for feature learning. By maximizing the mutual information among the augmented views from one instance (i.e., positive pairs), the contrastive pre-trained representation can even achieve comparable transfer performance with its supervised counterpart [8].

The success of self-supervised contrastive feature learning relies on how data augmentation is structured. SimCLR [2] and BYOL [4] have explored various data augmentation strategies by introducing some image transformations to increase the view variance. However, these data augmentation

⊠ indicates orresponding author. This work is supported by Shenzhen Science and Technology Innovation Commission (JCYJ20200109114835623, JSGG20220831105002004), and Guangdong Provincial Key Laboratory of Cloud Security Key Technology (2022B1212020006).

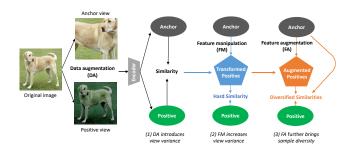


Fig. 1. In self-supervised contrastive learning, data augmentation (DA) introduces view variance usually by transforming the original input. Instead, feature augmentation (FA) aims to further increase the sample diversity in the feature space after the encoder. Feature manipulation (FM) is a special case of FA that focuses on mining hard examples.

strategies have several intrinsic limitations: (1) Data augmentation is generally domain-specific. For example, colorization may work effectively for images but not for videos. Hence, designing an effective DA approach requires experienced domain knowledge [9], [10]. (2) Data augmentation incurs task bias. Previous works [11]–[14] have shown that pretrained models can transfer well to the downstream task of image classification rather than object detection. (3) Data augmentation is inflexible. For instance, data augmentation via geometric transformations (e.g., cropping, color distortion) typically needs a grid search to find the optimal parameter setting. Previous works [2]–[4] have to conduct a customized search to find the sweet-spot parameter of an augmentation.

To tackle the aforementioned limitations, feature augmentation (FA) [15] is introduced to conduct "data augmentation" in the feature space instead of the original input space. This means we can search feature points to enlarge training data. Notably, since FA does not directly work on the raw data, it is a domain-agnostic augmentation. Secondly, FA can better tackle the task-bias problem thanks to fewer hand-designed operations. Thirdly, as we see from Fig. 1, FA is performed by merely manipulating the encoder's feature vectors, which makes FA very flexible.

Feature augmentation for self-supervised learning [16] is an underexplored problem so far. Previous works [17]–[19] propose to adapt the distribution of training data by transforming the original data in the feature space. This is often referred to as feature manipulation (FM), including methods like manifold-mixup [17], [18] and nearest-neighbor operation [19], [20]. Basically, FM can be regarded as a special case of FA. As shown in Fig. 1, except for searching hard examples

in FM, FA further pursues the diversity of samples to obtain better robustness and generalization. To achieve general FA, there are still three open problems: (1) Architecture: which network architecture is suitable for FA? SimCLR [2] and BYOL [4] utilize a feature projector and a predictor, while MoCoV1 [21] only has a light MLP head. How to use the projector and predictor and where to perform FA need further investigations. (2) Gradient flow: how to control the flow of gradient back-propagation? The gradients in DA propagate through the main encoder, and then the model learns the view variance from DA. Unlike DA conducted at the input level, FA performs feature-level augmentation. Hence, FA cannot propagate through the main encoder, which makes the model easier to overfit. Therefore, it is essential to control the gradient flow to avoid overfitting. (3) The relationship between FA and DA: the nearest neighbors of a sample instance can serve as a proxy for strong DA [22]. However, the relationship between FA and DA is still unclear. What kind of DA setting is good for FA? Could FA make up a deficiency when FA is insufficient?

To address these problems, we have made a systematic study on FA for self-supervised contrastive learning. Concretely, we integrate both the instance discrimination (cross-entropy-based, e.g., [2]) and the instance similarity (prediction-based, e.g., [4]) self-supervised contrastive learning paradigms [16] into a unified framework. Then, we thoroughly investigate the network architecture for optimal learning performance. We conduct an empirical evaluation of several proposed FA methods to verify their effectiveness. During this study, we also find out some practical principles for effectively applying FA in self-supervised contrastive pre-training.

In summary, we have made the following contributions:

- We propose a unified FA framework for self-supervised contrastive learning, which enables us to explore various network layouts of FA to achieve optimal performance.
- We conduct a systematic study on FA from three perspectives: the network architecture, the gradient flow, and the relationship between FA with DA.
- Our proposed FA framework brings consistent performance improvements on the downstream image classification task as well as better generalization for object detection, by integrating the instance discrimination and the instance similarity pre-training paradigms.

II. RELATED WORK

Data augmentation for contrastive learning. Data augmentation (DA) [23] is the most indispensable part of self-supervised contrastive learning as mentioned in [24]. For example, in terms of DA's difficulty: SimCLR [2] investigates the importance of DA and finds out that contrastive pre-training needs stronger DA than the supervised counterpart. Subsequently, InfoMin [3] suggests increasing DA to the "sweet spot" of mutual information can improve transfer performance. In terms of the number of augmented views, SimCLR [2], MoCo [21], BYOL [4] and most contrastive methods [3], [14], [25], [26] apply two views to construct one

positive pair. Beyond two views, SwAv [1] proposes multicrop to augment additional small views to construct multiple positive pairs. Besides, ReSSL [27] and MSF [22] discuss their superiority based on weak/strong data augmentation settings.

Mixup based contrastive pre-training. Mixup [28] offers a solution for domain-agnostic pre-train. By simply interpolating two data points, mixup could be a general DA strategy without the specific even laborious design for a certain modality or a particular domain (e.g., color jittering for natural image and word mask for language). I-Mix [29] and DACL [10] verify the efficacy of Mixup in both vision and non-vision modalities. In the medical image domain, C2L [30] proposes a large-scale self-supervised contrastive model and shows the superiority of Mixup. For the natural image domain, Unmix [31] empirically reveals the flexibility, universality and consistent performance improvement of mixup on a variety of main-stream self-supervised methods.

Feature manipulation in self-supervised learning. We have mentioned the application of feature manipulation on visual self-contrastive leaning [18]-[20], [22]. Besides, Wu et.al [32] propose feature hallucination which focuses on augmenting an extra positive view in the feature space for self-supervised contrastive learning. Metaug [33] performs a meta-learning strategy to construct the augmentation generator updated by the reward from the encoder and directly augment the discriminative features in the feature space. From the new perspective for energy-based contrastive learning [34], VEM [35] generates negative features from the energy-based feature distribution in self-supervised contrastive learning. More recently, I-JEPA [36] proposed to perform prediction in the feature space of the joint-embedding architecture, which is somewhat similar to the philosophy of such a feature-level augmentation method.

Feature augmentation beyond self-supervised learning. The idea of augmenting samples in feature space has been explored for several years. DeVries et.al [15] first investigate FA as a domain-independent and general-purpose strategy to improve generalization with limited labeled samples in various modalities (e.g., speech, sensor processing, motion capture, and images). In the NLP community, Kumar et.al [37] compare six FA methods on the few-shot intent classification task and FA provides an effective way to improve the performance with very few available samples. Similarly in low-shot visual learning, Hariharan et.al [38] present a FA method for hallucinating additional examples for the data-starved categories by transferring variation from the base categories. In supervised metric learning to improve the performance, the embedding expansion technique [39] selects the hardest negative pair from the interpolation between positives and negatives. DVML [40] and DAML [41] generate new hard examples by variational and adversarial generators.

III. FEATURE AUGMENTATION FOR CONTRASTIVE LEARNING: APPROACH & EXPLORATION EXPERIMENTS

In this section, we first represent the basic contrastive learning framework with FA. Next, we illustrate the adopted FA methods for contrastive learning. After that, we empirically evaluate the three unsolved problems: 1) we leverage the projector and predictor modules to equip with FA and select the proper architecture; 2) we validate the significance of the stop-gradient for the gradient-flow problem; 3) we empirically explore the relationship between FA and DA, based on which we confirm the ultimate parameter setting of data augmentation. Finally, we apply FA on the instance similarity baseline, BYOL, and achieve consistent improvement.

A. The Basic Framework with Feature Augmentation

The feature-level self-supervised works are based on different contrastive baselines. For example, the design of NNCLR and MSF relies on SimCLR and BYOL, with the heavy projector and predictor MLP head. But MoChi comes from MoCov1 with a light MLP head. Such a misalignment of architecture confuses the implementation of FA. To find a better architecture for FA, we start to build FA on the basic structure of SimCLR.

SimCLR is a basic structure of self-supervised contrastive learning, as shown in Fig. 2(a) (bottom part). Concretely, two pipelines of random data augmentation T_{anchor} and $T_{positive}$ transform the same image into two views, which are forwarded by the same encoder+projector to get the positive feature pair (z_i, z_i^+) , while the negatives come from all the other features in the current mini-batch. Notice that, following the setting of SimCLR, the encoder is ResNet-50 and the projector is a two-layer non-linear MLP *without BN*. Then we adopt the contrastive loss for the original data sample:

$$\mathcal{L}_{i}^{\text{original}}(z_{i}, z_{i}^{+}) = -\log \frac{\exp \left(\ell_{2}(z_{i}) \cdot \ell_{2}(z_{i}^{+})/\tau\right)}{\sum_{k=1}^{N} \exp \left(\ell_{2}(z_{i}) \cdot \ell_{2}(z_{k}^{+})/\tau\right)}$$
(1)

For one mini-batch with N samples, the overall loss is $\mathcal{L}^{\text{original}} = \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}_{i}^{\text{original}}$, which corresponds to the original contrastive loss.

The original contrastive loss only learns the variance brought by the prior data augmentation pipelines. Because of the shortage of views to construct more positive pairs, previous works provide practical solutions. For example, multi-crop [1] augments additional small views to compare with two large views from the same image, to learn a more robust model. Though increasing the number of views from the input level is direct and powerful, the training memory/time/computation increases significantly with more views generated from the DA pipeline. Hence, to improve the sample diversity of the feature representation and step forward to multiple positive pairs with a small cost, we propose to use feature augmentation for the contrastive loss, as shown in Fig. 2(a), After the feature augmentation, the new positive pair $(z_i, FA(z_i^+))$ will participate in the contrastive learning process. So we have:

$$\mathcal{L}_{i}^{\text{FA}}(z_{i}, FA(z_{i}^{+})) = -\log \frac{\exp(\ell_{2}(z_{i}) \cdot \ell_{2}(FA(z_{i}^{+}))/\tau)}{\sum_{k=1}^{N} \exp(\ell_{2}(z_{i}) \cdot \ell_{2}(FA(z_{k}^{+}))/\tau)}$$
(2)

The average loss of the mini-batch is $\mathcal{L}^{FA} = \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}_{i}^{FA}$. The final loss of FA is the average of \mathcal{L}^{FA} and $\mathcal{L}^{\text{original}}$ to combine the original and FA view. Note that ℓ_{2} normalization is applied in each contrastive loss, no matter for the original feature or the augmented feature. We ensure that the contrastive learning process happens in the unit sphere.

We have also attempted to construct the positive pair of two FA views, namely $(FA(z_i), FA(z_i^+))$, which is too difficult for the model and even leads to non-convergence for some FA methods. Therefore, we follow the positive pair setting of $(z_i, FA(z_i^+))$ to contrast one FA view with the original one.

B. Feature Augmentation Methods

After the introduction of the basic FA framework, we now start to discuss the detailed FA methods. The motivation of recent self-contrastive works with feature space manipulation can be summarized as "increasing the view variance to learn an invariant representation" from the InfoMin principle [3]. For example, they leverage hard example mining [18], [19] and the nearest neighbor to replace the original feature [20], [22]. The purpose is to make the pre-training process harder and boost the model performance. However, our goal of FA is to augment more feature points close to the original feature. Using the augmented feature views, more contrastive learning processes can be organized for the pre-training and thus improve the robustness and generalization. Hence, we discard hard-example-mining methods which may be "out-ofmanifold" and lead to non-convergence, like the extrapolation or large-scale mask.

Instead, we integrate three feature augmentation methods from previous works:

- (1) Nearest neighbor (NN): NNCLR and MSF utilize the NN method to introduce more variance. It significantly boosts the performance of image classification. Following their setting, we employ a feature bank [42] with the size 65536 to store the ℓ_2 normalization features from previous iterations. And we select the most similar k features as the augmentations.
- (2) Random dropout mask (Mask): SimCSE [43] utilizes random dropout masks on features to construct different positive embeddings. We apply random discrete dropout masks (20%) on features to bring more robustness to the pre-trained model, i.e., the masked 256-d feature will retain 80% original value while another 20% will be zeros. The augmented k features have k various random masks. We have tried the mask rate of 20% and 50% in our experiments, and the latter is harmful to the pre-training performance.
- (3) Mixup interpolated noise: It seems that mix-up is the most popular strategy in self-contrastive learning, no matter in the data-level [10], [29], [31] or the feature-level [18], [19]. Our motivation is to introduce noise and augment a few samples near the original features. Hence, we employ a mild mixup strategy among them (i.e., DACL [19]) but in the feature space by:

$$f_{aug} = \lambda \cdot f_{original} + (1 - \lambda) \cdot f_{noise}$$
 (3)

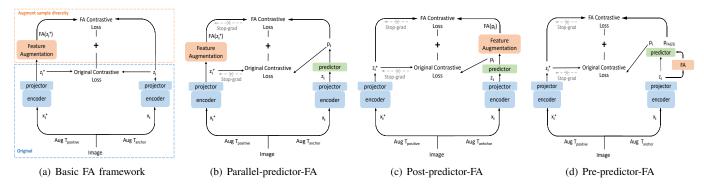


Fig. 2. Contrastive learning architectures with feature augmentation (FA). (a) is the basic framework. (b)-(d) extend (a) with additional predictors.

TABLE 1

IMAGENET-100 PERFORMANCE OF VARIOUS CONTRASTIVE ARCHITECTURES WITH FEATURE AUGMENTATION. (GREEN FONTS) INDICATE INCREASES OVER BASELINES, WHILE (GREY FONTS) INDICATE DECLINES. THE PARALLEL-PREDICTOR-FA GETS THE LARGEST PERFORMANCE GAINS.

Method	No FA	Mask	NN	NN noise	Batch noise	Gaussian noise
Basic	77.1	77.4 (+0.3)	75.9 (-1.2)	77.4 (+0.3)	77.0 (-0.1)	77.3 (+0.2)
+strong proj		78.4 (+0.6)	77.7 (-0.1)	78.0 (+0.2)	78.2 (+0.4)	78.8 (+1.0)
Parallel-predictor-FA	78.4	78.3 (-0.1)	79.6 (+1.2)	79.3 (+0.9)	79.2 (+0.8)	78.9 (+0.5)
Post-predictor-FA	78.4	79.1 (+0.7)	78.7 (+0.3)	77.9 (-0.5)	79.1 (+0.7)	78.5 (+0.1)
Pre-predictor-FA	78.4	79.3 (+0.9)	77.0 (-1.4)	78.5 (+0.1)	78.8 (+0.4)	78.6 (+0.2)

, where $\lambda \sim U(\alpha, 1.0)$ is randomly sampled from a uniform distribution. We set $\alpha=0.85$ so that the augmented feature is less affected by the noise feature. We provide three choices of f_{noise} : (i) NN noise: the k nearest neighbor of $f_{original}$; (ii) Batch noise: random selecting k features from the same batch of $f_{original}$; (iii) Gaussian noise: random sampling k gaussian noise from $\mathcal{N}(0,0.2)$. We choose three kinds of these noises with different randomness to evaluate the efficacy of FA.

Preliminary experimental evaluation of the basic framework with FA: Particularly, we focus on three unsolved open problems (sectionI) and provide empirical solutions, to better equip FA for self-supervised contrastive models. We report the performance of the basic framework with FA on ImageNet-100 in Table I. Please refer to the relative performance gain over the baseline. In the first row of Table I (i.e., Basic), it appears that FA only brings a little performance to the contrastive pretraining, while some FA methods drop a lot (e.g., NN method). It is noteworthy that NNCLR applies the enhanced projector (three non-linear MLPs followed by batch normalization [44]). To align the experimental settings, we replace this 3-layer BN projector with a strong projector in the basic framework. As shown in Table I (i.e., +strong proj), using a strong projector indeed improves the performance of FA. We attribute that a strong projector with a large capacity arouses the efficacy of FA. Thus, we continue to use this strong projector to perform the remaining experiments.

C. Feature Augmentation Architectures with Predictors

We extend the basic framework (Fig. 2(a)) to three additional network layouts as shown in Fig. 2, by equipping FA with predictor modules. This is inspired by the utilization of the predictor after the projected features in BYOL [4]. BYOL is a kind of instance similarity learning method without discriminating the negative samples, where the model tends to collapse during pre-training. Directly measuring the similarity

of (z_i, z_i^+) in projector space can lead to model collapse if they are constant across views. The predictor further introduces more transformations by a parameterized non-linear MLP layer. Thus, the predictor is applied asymmetrically on one single view z_i and gets the prediction p_i . As a result, the learning objective becomes more challenging when maximizing the similarity between the views in different feature spaces (p_i, z_i^+) rather than views in the same feature space (z_i, z_i^+) . Besides, MSF and NNCLR also apply the predictor after the projected features with feature manipulations. Such cross-level positive pair and contrasting improve the view variance for the pre-training process.

Therefore, we equip FA with the predictor for contrasting views in various feature spaces We can regard FA and the predictor module as two kinds of transformations: the predictor is a parametric transformation by a learnable non-linear MLP head, and FA augments the samples close to the original feature. Naturally, we ask: where to add the predictor and FA, by applying the two transformation strategies on both projected views or a single view? Regarding this, we have designed three architectures as shown in Fig. 2(b), 2(c) and 2(d):

(1) Parallel-predictor-FA: The first idea is to separate FA and the predictor into two views independently, which looks like parallelly applying two kinds of transformation methods on two branches. As shown in Fig. 2(b), we apply FA on the positive view and projector on the anchor view to construct the cross-level positive pair $(p_i, FA(z_i^+))$. Compared with the basic framework $(z_i, FA(z_i^+))$, the predictor brings the transformation from the MLP layer with a more challenging robust contrastive process. The final augmented loss is the average of the cross-level loss $\mathcal{L}^{\text{FA}}(p_i, FA(z_i^+))$ and $\mathcal{L}^{\text{original}}(p_i, z_i^+)$, which combines the original and FA view. In a word, the Parallel-predictor-FA architecture contrasts the

TABLE II IMAGENET-100 performance of each architecture with FA that does not use the stop-gradient operation.

Method	Baseline	Mask	NN	NN noise	Batch noise	Gaussian noise
Parallel-predictor-FA	78.4	78.2 (\dagger)	77.7 (\()	77.9 (↓)	78.3 (↓)	77.7 (\psi)
Post-predictor-FA	78.4	77.4 (\()	78.3 (\psi)	78.3 (↓)	78.6 (†)	78.0 (\psi)
Pre-predictor-FA	78.4	$77.8~(\downarrow)$	$77.9~(\downarrow)$	78.5 (†)	78.4 (↓)	78.0 (\lambda)

TABLE III IMAGENET-100 performance of FA when applying different settings of data augmentation pipeline

Method	Baseline	Mask	NN	NN noise	Batch noise	Gaussian noise
SymmWeakAug	74.4	75.6 (+1.2)	77.6 (+2.2)	76.1 (+1.7)	75.4 (+1.0)	75.9 (+0.5)
SymmStrongAug	78.0	78.4 (+0.4)	78.9 (+0.9)	78.3 (+0.3)	78.5 (+0.5)	78.5 (+0.5)
AsymmStrongAug	77.8	78.5 (+0.7)	79.4 (+1.6)	79.1 (+1.3)	79.4 (+1.6)	78.8 (+1.0)

predicted anchor view with the FA view and the original one.

- (2) **Post-predictor-FA:** In this model, we apply the predictor and FA on the same side. As shown in Fig. 2(c), we use FA after the predictor. This design applies two transformations (the predictor followed by FA) on the anchor view, and it contrasts with the positive projected view $(FA(p_i), z_i^+)$. The final augmented loss is the average of the cross-level loss $\mathcal{L}^{\text{FA}}(FA(p_i), z_i^+)$ and $\mathcal{L}^{\text{original}}(p_i, z_i^+)$.
- (3) **Pre-predictor-FA:** Another form of performing both FA and predictor on the same side is to first apply FA and then predictor on the anchor view. The augmented cross-level positive pair $(p_{FA(z_i)}, z_i^+)$ also contrasts the double-transformed anchor view with the positive projected view. The final augmented loss is the average of the cross-level loss $\mathcal{L}^{\text{FA}}(p_{FA(z_i)}, z_i^+)$ and $\mathcal{L}^{\text{original}}(p_i, z_i^+)$.

Experimental evaluation of different FA architectures: We perform experiments on ImageNet-100 to evaluate which architecture could better utilize the predictor and enhance the efficacy of FA. Please refer to the relative performance gains over the baseline. As shown in Table I, applying FA and projector parallel on two views can provide the maximum benefit on most FA methods except the mask. Actually, the mask method can benefit from single-side architecture (79.1%) and 79.3% for Post/Pre-predictor-FA architecture). When only one view experiences too many transformations, the performance improvement will not be comparable with the Parallelpredictor-FA architecture and even drops a lot for the NN method. To prevent the scenario of out-of-manifold, we discard Post/Pre-predictor-FA architecture and adopt the Parallelpredictor-FA architecture with good performance. Please notice that, when the baseline is strong (i.e., with higher performance), it generally becomes harder to gain further performance improvement. Hence, the Parallel-predictor-FA architecture indeed enhances the efficacy of FA. Besides, such Parallel-predictor-FA architecture has been utilized in existing self-contrastive pre-training works, such as OBoW [45], DINO [46], BYOL, MSF and NNCLR. However, these methods have not validated that this architecture is suitable for FA.

The importance of stop-gradient: Besides the predictor, BYOL also applies the momentum encoder and stop-gradient to make z_i^+ stable and more challenging for the instance similarity task. However, our framework is a single-encoder

model. We could not apply the EMA (i.e., exponential moving average, the updating strategy of the momentum encoder) to stabilize the positive projected view z_i^+ . Therefore we only employ the stop-gradient to control the objective loss. The effect of stop-gradient can be understood as an optimization strategy to prevent over-fitting. The non-parametric transformation of FA might be overfitted during the pre-training, so we apply the stop-gradient to the projected features (blocking the gradient flow of z_i^+ or $FA(z_i^+)$) to prevent it from the model collapse. As shown in Table II, when we remove the stop-gradient of three predictor-based architectures, the performance gains of FA become almost negative. Without stop-gradient, the benefits of FA strategy start to vanish, and even harm the final performance. Therefore, we conclude that stop-gradient is a key ingredient for the success of feature augmentation.

D. Exploring the Relationship with Data Augmentation

Even though previous works have extensively discussed the data augmentation [2], [4], [22], the relationship between DA and FA is still unclear. For example, although DA and FA are two independent components, can FA make up a deficiency of DA? Moreover, what kind of DA pipeline setting is suitable for an effective feature augmentation? In this section, we discuss their relationships to find out a better data augmentation setting for FA. Based on the Parallel-predictor-FA architecture, we design three DA settings:

- (1) Symmetric weak augmentation (SymmWeakAug): In this setting, we remove two DA strategies, i.e., Gaussian blurring [2] and solarization [4]. These are relatively violent means compared with the color or grey change. Notably, the parameter setting of DA is symmetric.
- (2) Symmetric strong augmentation (SymmStrongAug): In this setting, we adopt all the DA strategies proposed in BYOL with different parameters. Similarly, the parameter setting of DA is symmetric, namely $T_{anchor} = T_{positive}$.
- (3) Asymmetric strong augmentation (AsymmStrongAug): This setting also employs all the DA strategies in BYOL. But the parameter setting of the DA pipeline is asymmetric, $T_{anchor} \neq T_{positive}$. We keep all strategies in one pipeline while removing Gaussian blurring and solarization from another pipeline.

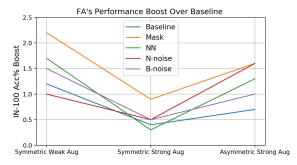


Fig. 3. The performance boost of feature augmentation (FA) over baselines when applying different settings of data augmentation (DA). FA makes up a deficiency for DA, and asymmetric DA setting is suitable for FA.

The detailed evaluation results are listed in Table III. For a better comparison, we also draw the performance boost of FA over the baseline in Fig. 1. From SymmStrongAug to SymmWeakAug, the relative performance boost of FA has been improved a lot. This indicates that FA can better improve DA effectively. The SymmWeakAug+NN setting (77.6%) can even catch up with the performance of the baseline with strong augmentation (78.0%). MSF and NNCLR also demonstrate that their feature space manipulation is helpful in the weak-DA setting.

In addition, from SymmStrongAug to AsymmStrongAug, the FA's relative performance has been significantly boosted across all FA methods. This comparison demonstrates that the asymmetric parameter setting of the DA pipeline is more appropriate for feature augmentation. Hence, we employ BYOL's DA parameters as the default setting for our FA experiments except those particularly mentioned. The asymmetric DA could introduce more variance for the positive pair during pretraining. By combining asymmetric DA, FA and predictor, we can construct a better contrastive model.

E. Upgrading BYOL with Feature Augmentation

After previous analysis and empirical evaluation, we adopt the Parallel-predictor-FA architecture, stop-gradient strategy and the data augmentation setting for BYOL [4]. Next, we evaluate the efficacy of FA on the instance similarity framework, BYOL. The Parallel-predictor-FA architecture is very similar to BYOL's, which directly adds FA on the top of the momentum encoder and applies cross-level contrast. As shown in Table IV, all FA methods bring consistent improvements based on the BYOL baseline. We utilize the original structure of BYOL without a strong projector. This performance boost indicates that the Parallel-predictor-FA architecture and stop-gradient are practical for feature augmentation.

Augment more samples and free the loss: With the help of FA, we can obtain more samples similar to the original feature point. As a result, we augment more samples to see whether FA can provide more diverse positives in the feature space. We try to augment 4 samples in the feature space. The final loss is the average of the original loss and four augmented losses. As listed in the second row of Table IV, augmenting 4 samples does not gain better performance than only augmenting one sample. We conjecture that the average

$\label{thm:table_iv} \textbf{TABLE IV} \\ \textbf{IMAGENET-100 PERFORMANCE ON BYOL WITH FEATURE}$

AUGMENTATION. THE BASELINE'S ACCURACY IS 77.4%. FA BRINGS IMPROVEMENT FOR BYOL IN VARIOUS DEGREES. AUGMENTING MORE SAMPLES AND FREEING THE LOSS FURTHER IMPROVES THE ACCURACY

Method	Mask	NN	NN noise	Batch noise	Gaussian noise
Aug 1	77.9 (+0.5)	78.7 (+1.3)	79.0 (+1.6)	78.8 (+1.4)	78.3 (+0.9)
Aug 4	78.1 (+0.7)	77.7 (+0.3)	79.0 (+1.6)	78.4 (+1.0)	78.9 (+1.5)
Aug 1 free	79.6 (+2.2)	79.5 (+2.1)	79.5 (+2.1)	79.2 (+1.8)	79.1 (+1.7)
Aug 4 free	79.8 (+2.4)	79.1 (+1.7)	78.7 (+1.3)	78.8 (+1.4) 78.4 (+1.0) 79.2 (+1.8) 79.4 (+2.0)	79.1 (+1.7)

operation in the final loss may limit the effect of FA. Thus, we free the loss (i.e., remove the average operation in the final loss) and use the $2\times$ or $5\times$ of the previous loss when augmenting 1 or 4 samples. The final two rows in Table IV show the consistent improvement of free loss over the baseline. The free-loss strategy gains higher accuracy than non-free loss, which suggests FA has great potential to boost a pre-trained model.

IV. EXPERIMENTS: PRE-TRAINING & TRANSFERRING

In this section, we report the results of self-supervised contrastive learning (on ImageNet2012 [47] with 1,000 classes) over both the instance discrimination (i.e., cross-entropy-based) [2] and instance similarity (i.e., prediction-based) [4] paradigms, and transfer learning (object detection on PASCAL VOC [48]) to evaluate the generalization of FA. Pytorch [49] and the Pytorch-Lightning library [50] are used for all experiments.

A. Self-supervised Pre-training on ImageNet

Instance discrimination: We first evaluate the efficacy of FA based on instance discrimination contrastive learning using the Parallel-predictor-FA architecture. Following the same setting of NNCLR [20], we use ResNet-50 as the backbone, with a strong projector and a 3-layer-MLP followed by BN and ReLU activations (no ReLU in the last layer of the projector). The size of the projector is [2048, 2048, 256]. The predictor is a 2-layer-MLP with the size of [4096, 256] and no BN or ReLU in the last layer. We apply the SGD optimizer (lr=0.4) with cosine scheduler, momentum=0.9, weight decay= 10^{-5} , cosine warm-up 10 epochs) to learn the pre-trained model for 100 or 200 epochs. The batch size is 512 due to the limitation of computation capability and we use accumulating gradient to make up the gap to the large batch size (i.e., 4096 in NNCLR). LARS [51] is also applied with the trust coefficient 0.02. As aforementioned in Section III-D, we directly adopt the setting of the DA pipeline in BYOL [4]. The temperature τ in contrastive loss is 0.2. Notice that Sync BN and gradient clip are applied. We augment only one sample. No free loss is utilized in the instance discrimination model.

Instance similarity: We apply the SGD optimizer (lr=0.45 with cosine scheduler, momentum=0.9, weight decay=10⁻⁶, cosine warm-up 10 epochs) and train for 100 epochs. The LARS's trust coefficient is set to 0.01 and here gradient clip is unnecessary. We augment one and four samples and employ the free loss for this model. The momentum parameter starts

Instance Discrimination	Batch Size	Epochs	Linear Acc%	Instance Similarity	Batch	Aug num	Linear Acc%
Official reported r	esults 4096	100/200	69.4/70.7				
Baseline + Mask + NN + NN noise + Batch noise + Gaussian noise	512 512 512 512 512 512 512	100/200 100/200 100/200 100/200 100/200 100/200	66.0/68.4 67.2/69.4 68.6/70.9 67.3/69.3 67.4/69.5 67.4/69.4	BYOL [†] +Mask +NN +NN noise +Batch noise +Gaussian noise	512 512 512 512 512 512 512	- 1/4 1/4 1/4 1/4 1/4	69.1 69.5/70.0 71.4/70.9 70.0/70.8 69.8/70.2 69.9/70.1

TABLE VI
TRANSFERRING TO OBJECT DETECTION ON PASCAL VOC (IN-1K STANDS FOR IMAGENET-1K).

Method	Epochs	IN-1k	AP	AP ₅₀	AP ₇₅	Epochs	IN-1k	AP	AP ₅₀	AP ₇₅
Baseline	100	66.0	52.6	80.7	57.5	200	68.4	53.7	81.2	59.5
+ Mask	100	67.2	54.4	81.6	60.4	200	69.4	55.1	82.0	61.9
+ NN	100	68.6	52.5	80.7	57.1	200	70.9	53.3	81.2	58.5
+ NN noise	100	67.3	54.6	81.6	60.2	200	69.3	55.3	82.3	61.9
+ Batch noise	100	67.4	54.4	81.4	60.3	200	69.5	55.3	82.1	60.6
+ Gaussian noise	100	67.4	54.8	81.3	60.5	200	69.4	55.3	81.8	61.4

from 0.99 to 1.0. Other settings keep the same with Instance Discrimination.

B. Evaluation on ImageNet-1k

We evaluate the pre-trained representation by training a linear layer on top of the frozen backbone. It is trained with the SGD optimizer (epochs=60, batchsize=256, weight decay=0, momentum=0.9, lr=1.0 / 0.2 for instance discrimination / similarity, decaying at [40, 50, 55]th epochs). The results on ImageNet-1k are shown in Table IV.

For instance discrimination, FA methods bring consistent improvements over the baseline model, no matter in 100/200 training epochs as shown in Table IV. These improvements indicate that feature augmentation is also effective when pretraining on the large-scale dataset. Among the FA methods, NN brings the biggest improvement, about 2.5% over the baseline model on 100/200 epochs setting. We also compare the baseline+NN model with NNCLR which is the stateof-the-art self-contrastive model. The baseline+NN cannot catch up with NNCLR in the 100-epoch setting, which can be attributed to our smaller batch size. Note that due to computation limitations, we only use 512 batch-size with gradient accumulation. Because the instance discrimination method uses the mini-batch as negative samples for contrastive learning, a smaller batch size means less data being explored. However, when training longer to 200 epochs, the performance of baseline+NN catches up with NNCLR (70.9% v.s. 70.8%). This indicates that feature augmentation can provide sample diversity and mitigate the limitation of data lack during pretraining.

For instance similarity, FA methods produce various degrees of performance boosts over the BYOL model as Table IV shows. When augmenting 1 sample in the feature space, the random Mask only improves marginally $(69.5\% \ v.s. \ 69.1\%, 0.4\% \ improved)$ and the single NN significantly enhances the BYOL $(71.4\% \ v.s. \ 69.1\%, 2.3\% \ improved)$. When the augment

number increases to 4, the performance of FA-NN degrades slightly while other FA methods get higher accuracy than augmenting only one sample. The performance drop of NN is consistent with NNCLR [20], the best result usually comes from the most similar feature. Overall, FA indeed boosts the BYOL regardless of 1 or 4 samples in feature space. We use free loss in this experiment.

C. Transfer Learning on Object Detection

To evaluate the generalization of FA, we apply Faster R-CNN [52] for downstream object detection. We follow the procedure and default parameters adopted in [21] based on Detectron2 [53]. The pre-training parameters are finetuned on PASCAL VOC [54] trainval07+12 and evaluated at test07 set. The results are reported in Table IV. As we can see, FA methods (Mask and three Mixup interpolated noise) consistently improve over the baseline when transferring to the detection task. Also, we can conclude that these four FA methods can help the baseline model generalize better on the downstream detection task. However, the NN methods slightly hurt the baseline, even if NN gets the highest results on ImageNet-1k linear evaluation. We argue that the NN methods concentrate on finding the most semantically similar feature, which could overfit during the pre-training process on the ImageNet-1k dataset. Hence, NN does not improve the generalization although achieves a good performance on ImageNet-1k linear evaluation. In summary, FA can tackle the task-bias problem and generalize well thanks to less data bias.

V. Conclusion

We introduce and empirically evaluate the efficacy of feature augmentation for self-supervised contrastive learning. We reconsider feature augmentation as a basic strategy to improve sample diversity, and we solve three open problems ignored by previous feature manipulation works. By integrating the proposed FA methods, we verify the proper architecture, ingredient component (stop-gradient) and the relationship with data augmentation, for this new strategy. Experiments on baselines of the instance discrimination and instance similarity paradigms show that FA consistently brings performance improvement and good generalization on downstream tasks.

REFERENCES

- M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," *NeurIPS*, 2020.
- [2] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *ICML*, 2020.
- [3] Y. Tian, C. Sun, B. Poole, D. Krishnan, C. Schmid, and P. Isola, "What makes for good views for contrastive learning," arXiv preprint arXiv:2005.10243, 2020.
- [4] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar et al., "Bootstrap your own latent: A new approach to self-supervised learning," NeurIPS, 2020.
- [5] W. Van Gansbeke, S. Vandenhende, S. Georgoulis, and L. V. Gool, "Revisiting contrastive methods for unsupervised learning of visual representations," *NeurIPS*, vol. 34, pp. 16238–16250, 2021.
- [6] E. Cole, X. Yang, K. Wilber, O. Mac Aodha, and S. Belongie, "When does contrastive visual representation learning work?" in CVPR, 2022, pp. 14755–14764.
- [7] C. Xu, D. Tao, and C. Xu, "A survey on multi-view learning," arXiv preprint arXiv:1304.5634, 2013.
- [8] R. D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, and Y. Bengio, "Learning deep representations by mutual information estimation and maximization," *ICLR*, 2019.
- [9] K. Lee, Y. Zhu, K. Sohn, C.-L. Li, J. Shin, and H. Lee, "i-mix: A domain-agnostic strategy for contrastive representation learning," in ICLR, 2021.
- [10] V. Verma, T. Luong, K. Kawaguchi, H. Pham, and Q. Le, "Towards domain-agnostic contrastive learning," in *ICML*. PMLR, 2021.
- [11] X. Wang, R. Zhang, C. Shen, T. Kong, and L. Li, "Dense contrastive learning for self-supervised visual pre-training," *CVPR*, 2021.
- [12] E. Xie, J. Ding, W. Wang, X. Zhan, H. Xu, Z. Li, and P. Luo, "Detco: Unsupervised contrastive learning for object detection," arXiv preprint arXiv:2102.04803, 2021.
- [13] Z. Xie, Y. Lin, Z. Zhang, Y. Cao, S. Lin, and H. Hu, "Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning," CVPR, 2021.
- [14] N. Zhao, Z. Wu, R. W. Lau, and S. Lin, "What makes instance discrimination good for transfer learning?" arXiv preprint arXiv:2006.06606, 2020
- [15] T. DeVries and G. W. Taylor, "Dataset augmentation in feature space," arXiv preprint arXiv:1702.05538, 2017.
- [16] R. Balestriero, M. Ibrahim, V. Sobal, A. Morcos, S. Shekhar, T. Goldstein, F. Bordes, A. Bardes, G. Mialon, Y. Tian et al., "A cookbook of self-supervised learning," arXiv preprint arXiv:2304.12210, 2023.
- [17] V. Verma, A. Lamb, C. Beckham, A. Najafi, I. Mitliagkas, D. Lopez-Paz, and Y. Bengio, "Manifold mixup: Better representations by interpolating hidden states," in *ICML*. PMLR, 2019, pp. 6438–6447.
- [18] R. Zhu, B. Zhao, J. Liu, Z. Sun, and C. W. Chen, "Improving contrastive learning by visualizing feature transformation," in *ICCV*, 2021.
- [19] Y. Kalantidis, M. B. Sariyildiz, N. Pion, P. Weinzaepfel, and D. Larlus, "Hard negative mixing for contrastive learning," *NeurIPS*, vol. 33, 2020.
- [20] D. Dwibedi, Y. Aytar, J. Tompson, P. Sermanet, and A. Zisserman, "With a little help from my friends: Nearest-neighbor contrastive learning of visual representations," arXiv preprint arXiv:2104.14548, 2021.
- [21] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in CVPR, 2020, pp. 9729–9738.
- [22] S. A. Koohpayegani, A. Tejankar, and H. Pirsiavash, "Mean shift for self-supervised learning," ICCV, 2021.
- [23] S. Yang, W. Xiao, M. Zhang, S. Guo, J. Zhao, and F. Shen, "Image data augmentation for deep learning: A survey," arXiv preprint arXiv:2204.08610, 2022.
- [24] P. H. Le-Khac, G. Healy, and A. F. Smeaton, "Contrastive representation learning: A framework and review," *IEEE Access*, vol. 8, pp. 193 907– 103 934, 2020
- [25] P. Bachman, R. D. Hjelm, and W. Buchwalter, "Learning representations by maximizing mutual information across views," in *NeurIPS*, 2019, pp. 15 509–15 519.
- [26] C.-Y. Chuang, J. Robinson, L. Yen-Chen, A. Torralba, and S. Jegelka, "Debiased contrastive learning," *NeurIPS*, 2020.

- [27] M. Zheng, S. You, F. Wang, C. Qian, C. Zhang, X. Wang, and C. Xu, "Ressl: Relational self-supervised learning with weak augmentation," *NeurIPS*, vol. 34, 2021.
- [28] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *ICLR*, 2018.
- [29] K. Lee, Y. Zhu, K. Sohn, C.-L. Li, J. Shin, and H. Lee, "i-mix: A strategy for regularizing contrastive representation learning," in *ICLR*, 2021.
- [30] H.-Y. Zhou, S. Yu, C. Bian, Y. Hu, K. Ma, and Y. Zheng, "Comparing to learn: Surpassing imagenet pretraining on radiographs by comparing image representations," in *MICCAI*. Springer, 2020.
- [31] Z. Shen, Z. Liu, Z. Liu, M. Savvides, T. Darrell, and E. Xing, "Unmix: Rethinking image mixtures for unsupervised visual representation learning," arXiv preprint arXiv:2003.05438, 2020.
- [32] J. Wu, J. Hobbs, and N. Hovakimyan, "Hallucination improves the performance of unsupervised visual representation learning," in *ICCV*, 2023.
- [33] J. Li, W. Qiang, C. Zheng, B. Su, and H. Xiong, "Metaug: Contrastive learning via meta feature augmentation," in *ICML*, 2022.
- [34] B. Kim and J. C. Ye, "Energy-based contrastive learning of visual representations," in *NeurIPS*, 2022.
- [35] T. Du, Y. Wang, W. Huang, and Y. Wang, "Variational energy-based models: A probabilistic framework for contrastive self-supervised learning," in *NeurIPS workshop*, 2022.
- [36] M. Assran, Q. Duval, I. Misra, P. Bojanowski, P. Vincent, M. Rabbat, Y. LeCun, and N. Ballas, "Self-supervised learning from images with a joint-embedding predictive architecture," in CVPR, 2023.
- [37] V. Kumar, H. Glaude, C. de Lichy, and W. Campbell, "A closer look at feature space data augmentation for few-shot intent classification," arXiv preprint arXiv:1910.04176, 2019.
- [38] B. Hariharan and R. Girshick, "Low-shot visual recognition by shrinking and hallucinating features," in ICCV, 2017, pp. 3018–3027.
- [39] B. Ko and G. Gu, "Embedding expansion: Augmentation in embedding space for deep metric learning," in CVPR, 2020, pp. 7255–7264.
- [40] X. Lin, Y. Duan, Q. Dong, J. Lu, and J. Zhou, "Deep variational metric learning," in ECCV, 2018, pp. 689–704.
- [41] Y. Duan, W. Zheng, X. Lin, J. Lu, and J. Zhou, "Deep adversarial metric learning," in CVPR, 2018, pp. 2780–2789.
- [42] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance discrimination," in CVPR, 2018, pp. 3733– 3742
- [43] T. Gao, X. Yao, and D. Chen, "Simcse: Simple contrastive learning of sentence embeddings," arXiv preprint arXiv:2104.08821, 2021.
- [44] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *ICML*. PMLR, 2015, pp. 448–456.
- [45] S. Gidaris, A. Bursuc, G. Puy, N. Komodakis, M. Cord, and P. Perez, "Obow: Online bag-of-visual-words generation for self-supervised learning," in CVPR, 2021, pp. 6830–6840.
- [46] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," in *ICCV*, 2021, pp. 9650–9660.
- [47] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *IJCV*, 2015.
- [48] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, pp. 303–338, 2010.
- [49] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *NeurIPS*, vol. 32, 2019.
- [50] W. Falcon and The PyTorch Lightning team, "PyTorch Lightning," Mar. 2019. [Online]. Available: https://github.com/Lightning-Al/lightning
- [51] Y. You, I. Gitman, and B. Ginsburg, "Large batch training of convolutional networks," arXiv preprint arXiv:1708.03888, 2017.
- [52] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *NeurIPS*, vol. 28, 2015.
- [53] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2," https://github.com/facebookresearch/detectron2, 2019.
- [54] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.