A Fine-grained Pre-training Essentials

We evaluate the capabilities learned out of three classes of pre-training mechanisms, namely self-supervised contrastive, non-contrastive, and supervised methods. In particular, we focus on discriminative feature extraction and object localization ability. Without loss of generality, we select MoCo.v2, BYOL, Rot-Pred, and supervised classification for comparison. To explore the effects of object localization, we develop a simple binary classification as a pre-training task where the model is asked to classify images from CUB as foreground class and images from COCO as background class. (rather than design detection specific modules as InsLoc [48]).

A.1 Experimental Setup

Dataset. We evaluate the performance of baselines' representation pre-trained on the training set of 100% ImageNet-1k (IN-1k), 10% IN-1k, COCO, and CUB, details of datasets used are described in Sec. 3. We use the same fixed split for the 10% IN-1k where we randomly sample 10% of the total training set size from each class.

Pre-training details. To ensure impartial comparisons, MoCo.v2 data augmentations are adopted for all self-supervised methods and follow the exact setup described in the original papers. OpenMixup [23] is adopted as the codebase. All models are pre-trained 200 epochs on 100% IN-1k and 800 epochs on other datasets. For the binary classification, the model is pre-trained by SGD optimizer with an initial learning rate of 0.1 adjusted by a cosine annealing scheduler, the SGD momentum of 0.9, and the weight decay of 0.0001. In contrastive learning pre-training, the input resolution is 224×224 and the data augmentation strategy follows MoCo.v2 [5] as following: Geometric augmentation is *RandomResizedCrop* with the scale in $[0.2, 1.0]$ and *RandomHorizontalFlip*. Color augmentation is *ColorJitter* with {brightness, contrast, saturation, hue} strength of $\{0.4, 0.4, 0.4, 0.1\}$ and an applying probability of 0.8, and *RandomGrayscale* with an applying probability of 0.2. Blurring augmentation is using a square Gaussian kernel of size 23×23 with a standard deviation uniformly sampled in $[0.1, 2.0]$. During the evaluation, images are resized to 256 pixels along the shorter side and are center cropped to 224×224.

Evaluations. We evaluate the learned representation with a linear evaluation protocol, and a fully supervised fine-tune evaluation protocol. The **linear evaluation protocol** is a commonly adopted protocol detailed in [4][17], i.e., train a linear classifier on top of the frozen representation on the labeled training set. We use SGD optimizer with a cosine annealing scheduler, the SGD momentum of 0.9, and the weight decay of 0. Based on supervised fine-grained classification settings, we adopt the batch size of 16 with 50 training epochs for small-scale fine-grained datasets, while using the batch size of 32 with 80 training epochs for iNat2018. To avoid evaluation deviations caused by the learning rate, we report the best test performance achieved among the initial learning rate in $\{0.1, 0.01, 0.001\}$ for each comparing method. The fully supervised **fine-tune evaluation protocol**, as proposed in [4, 52], fine-tunes the entire network on the training set with labels. Since the original protocols are designed for coarse-grained datasets such as IN-1k, we adopt fine-grained training settings: we use SGD optimizer with a cosine annealing scheduler and a batch size of 16 training 50 epochs. To avoid evaluation deviations, we sweep over the initial learning rate in

Methods	100% IN-1k			10% IN-1k			COCO			CUB		
	Finetune	Linear	MaxBoxAcc	Finetune	Linear	MaxBoxAcc	Finetune	Linear	MaxBoxAcc	Finetune	Linear	MaxBoxAcc
Supervised	79.25	67.64	52.18	74.68	63.07	51.42	65.41	61.39	49.87	67.82	49.85	37.31
Rot-Pred	67.66	25.29	46.14	68.81	23.71	47.22	67.95	16.54	43.39	**66.67**	**15.46**	**34.69**
MoCo.v2	73.19	**33.34**	47.73	69.62	**25.32**	**48.23**	68.47	**18.69**	**46.74**	63.21	15.03	33.36
BYOL	**76.63**	29.17	**49.23**	**70.42**	20.38	46.52	**68.55**	16.36	45.60	64.85	15.24	33.81

Table A1: **Comparison of pre-training methods.** Top-1 fine-tune and linear accuracy and MaxBocAcc (%) are reported.

$\{0.1, 0.05, 0.01, 0.005, 0.001\}$ and the weight decay in $\{0.0005, 0.0001\}$, and select the hyperparameters achieving the best performance on the validation set. The linear test accuracy of the pre-trained model is referred to as the model's discriminative feature extraction ability [14]. We refer to the fine-tune evaluation as a pre-training representation quality metric for fine-grained classification problems in a practical sense. For each method, we report mean top-1 accuracy on the test set over 3 trials. To evaluate the localization ability of different approaches, we use a class activation mapping (CAM) based metric **MaxBoxAcc** [6]. A larger MaxBoxAcc indicates better localization ability.

A.2 Essential Requirement and Formulation

Where does the gap lie between self-supervised and supervised pre-training? As shown in Table A1, when pre-trained on 100% IN-1k and 10% IN-1k, the supervised method consistently outperforms all self-supervised pre-training methods. Compared to supervised pre-training, all self-supervised approaches yield lower MaxBoxAcc, indicating a lack of localization ability. Self-supervised methods are task-agnostic and could only learn low-level features, i.e., gradient and direction-dependent features for rotation, and invariant features across views to cluster different objects for contrastive methods. However, the supervised method discards task-irrelevant information and extracts related semantic features. Deep CNN, such as ResNet, has its natural ability in localization during the supervised pre-training process. However, such localization ability could hardly be acquired during self-supervised pre-training. Also, notice that supervised pre-training on CUB yields much lower linear and fine-tune accuracy than self-supervised methods, which states that discriminative feature extraction is largely affected by the size of the dataset.

Why doesn't the contrastive method look at the bird? We hypothesize that the lack of localization ability comes from the commonly adopted *RandomResizedCrop* (RRC) augmentation, where a random size patch at a random location is cut from the original image and then resized to the original size. We verify the hypothesis on STL-10 [7] (a subset of IN-1k) based on BYOL and MoCo.v2 using their training settings on IN-1k. Figure A1 left shows that performances of both methods drop drastically as the

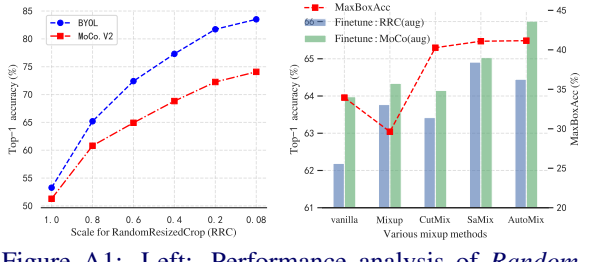

Figure A1: Left: Performance analysis of *RandomResizedCrop* (RRC) on STL-10 based on BYOL and MoCo.v2. The x axis indicates the scale factor for RRC and the Top-1 accuracy (%) of fine-tune evaluation is reported. Right: Performance analysis of the binary classification task with various mixup methods. The Top-1 accuracy (%) of fine-tune evaluation and MaxBoxAcc are reported.

cropped patch scale enlarges. The best accuracy is achieved with a scaling factor of 0.08, which is the default hyperparameter choice for current contrastive-based approaches. When asked to pull together two overly small patches cut from the same image, the model is forced to exploit low-level local texture features leading to poor localization ability. Different from contrastive-based methods, Rot-Pred takes in whole images rotated by four degrees as input. The authors claim that the model is required to understand the location and pose of the objects depicted in the image in order to predict the rotation angle. As can be seen from Table A1, Rot-Pred indeed yields better localization ability and overall performance than contrastive-based algorithms on CUB.

Then, we formulate current contrastive methods with a causal graph as illustrated in Figure A2 and will later use this concept to formalize our pre-training scheme. Let X be images with the content C composed of background prior B and foreground target prior T, generated with style prior S as augmentations like color jittering. Latent representation Z is learned and used to infer image labels Y. Contrastive methods assume image labels Y are an effect of whole image content C (both B and T) due to the local texture-biased nature. In this work, we propose to weaken the causality between B and Y to make Y a more direct effect of T.

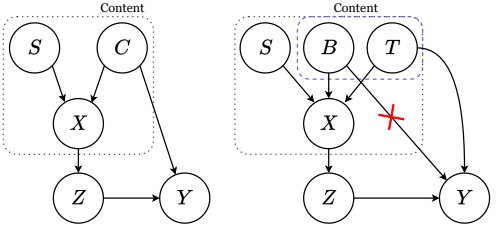

Figure A2: Causal interpretation of existing contrastive methods (left) vs. CVSA (right) from a causal perspective. The direct link denotes the causality from the cause to the effect.

Is localization all you need? We report linear and fine-tune test accuracy and MaxBoxAcc of the binary classification on the CUB test set pre-training with various mixup augmentations [26, 39, 49, 54] as well as image augmentations in Figure A1 right. From Table A1 and Figure A1, it is observed that the fine-tuned model using binary classification as pre-training yields comparable MaxBoxAcc as supervised pre-training, which indicates that a simple binary classification supervision signal empowers the network with localization ability. Yet, there still exists a vast fine-tuned accuracy gap compared to supervised pre-training on IN-1k. We assume that the gap mainly comes from an inferior feature extraction ability of the binary classification pre-training, as could be conducted from a much lower linear test accuracy. In other words, for better fine-grained recognition pre-training, discriminative feature extraction ability is as essential as localization ability.

Next, we investigate how different mixup augmentations affect the model's localization ability. We notice a simple interpolation between images as done by Mixup negatively impacts the model's localization ability. This negative impact is largely due to the unnatural characteristics of the mixed images. Cutmix mixes samples by replacing the image region with a patch from another training image, while SaliencyMix replaces the image region with the saliency region from another training image. We observe that both CutMix and SaliencyMix bring about better localization ability. However, directly applying these mixup-based augmentations to contrastive learning leads to degenerate solutions. Contrastive learning essentially expects positive pairs to share common semantic objects while keeping negative pairs as much dissimilar as possible. Due to the randomness introduced by such mixup algorithms, augmented images may contain multiple semantic objects or contain no semantic object at all. Without proper supervision, this easily causes the learned representation space to collapse during self-supervised contrastive pre-training. We address this problem

by proposing an image augmentation technique that swaps saliency regions of images which aims to introduce solely background variation.

Formulation. From the previous analysis, given a fine-grained classification problem, similar to [1], we assume $\mathcal{X}$ to be a set of all samples with an underlying set of discrete latent classes $\mathcal{C}$ that represent semantic content, we obtain the joint distribution between each sample x and its class c:

$$p(c,x) = p(c|x_{fore}) \cdot p(x_{fore}|x), \tag{8}$$

where x_{fore} stands for the foreground object. This factorization captures two important intuitions: (1) Given an image of a fine-grained object; the model should first localize the foreground object ($p(x_{fore}|x)$), namely, the localization ability of the model. (2) To further tell the species of the foreground object ($p(c|x_{fore})$), discriminative texture features should be extracted, namely, the texture extraction ability of the model. Following this formulation, a *dual-stage* pre-training pipeline is naturally proposed for self-supervised fine-grained recognition. In particular, we refer to previous contrastive learning methods such as MoCo.v2 and BYOL on large datasets such as ImageNet or COCO as the *first-stage* and the proposed CVSA as the *second-stage*. The model's discriminative texture extraction ability could be fulfilled by *first-stage* pre-training. In the *first-stage*, we regard the image content as a whole as the same assumption of current contrastive methods. For the *second-stage* pre-training, we propose a framework called cross-view saliency alignment (CVSA) to enhance the model's localization capability.

B More Ablation Experiments

We further study the impact of using saliency information provided by different saliency detection methods based on experiment settings in Sec. 3.3. We compare five well-recognized saliency detection methods (VSFs [29], GS [43], FST [42], RBD [59] and BSANet [31]) and the ground truth bounding box on CUB. As shown in Table B, the proposed SaliencySwap and alignment loss are robust to the quality of saliency bounding boxes because our approach helps the network to localize the object roughly and extract fine-grained semantic features. It is no need to provide accurate segmentation masks of the foreground objects as in object detection and segmentation [11, 35].

Method	VSFs	GS	FST	RBD	BSANet	Groundtruth
BYOL+SS	64.27	64.32	64.28	64.34	64.33	**64.35**
BYOL+SS+Align	64.89	64.94	64.93	64.97	**65.04**	65.02

Table A2: **Evaluation of different saliency detection methods for *second-stage only* pre-training.** Top-1 accuracy (%) under fine-tune evaluation is reported on CUB.

	Settings	V100	BYOL			BYOL+DiLo			BYOL+**CVSA**		
CUB-200	400 ep	1×	6.0h	30.3M	72.5	+0.5h	45.7M	+4.1	+0.5h	45.1M	**+4.6**
NAbirds	400 ep	4×	7.5h	30.3M	76.1	+2.0h	45.7M	+2.9	+1.5h	45.1M	**+3.5**

Table A3: **Comparison of computational overhead.** The total training time (hours), the number of parameters (M), and the performance of the *dual-stage* setting are reported.

We then compare the computation overhead and the performance gain in Table A3, demonstrating that the proposed CVSA significantly improves BYOL with limited extra computational overhead.

C Discussion

For fine-grained classification, our proposed CVSA aims to balance the abilities of target localization and discriminative feature extraction. As for contrastive-based methods designed for downstream tasks like object detection and segmentation, most frameworks perform object localization and classification in two network branches, focusing on improving localization ability. Compared to SaliencyMix [39] and DiLo [56], DiLo randomly places the masked foreground objects to raw background images such as texture backgrounds, while our proposed SaliencySwap swaps the saliency region in the randomly selected background image and the source image and regards the augmented view as a positive sample, as shown in Figure 1. Notice that SS and SaliencyMix only require coarse saliency information described by bounding boxes, while DiLo uses pixel-wise saliency masks. Compared to CASTing [35], it improves object localization by cropping views based on saliency regions (required mask-level supervision) and maximizing the similarity between learned saliency masks. However, for fine-grained classification, both discriminative feature extraction and target localization are crucial (performed in the same branch). Our alignment loss utilizes saliency maps of two images and aligns them with cross attention of the two views.