This ICCV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Semantics-Consistent Feature Search for Self-Supervised Visual Representation Learning

Kaiyou Song Shan Zhang Zimeng Luo Tong Wang Jin Xie
Megvii Technology
{songkaiyou, zhangshan, luozimeng, wangtong, xiejin}@megvii.com

Abstract

In contrastive self-supervised learning, the common way to learn discriminative representation is to pull different augmented "views" of the same image closer while pushing all other images further apart, which has been proven to be effective. However, it is unavoidable to construct undesirable views containing different semantic concepts during the augmentation procedure. It would damage the semantic consistency of representation to pull these augmentations closer in the feature space indiscriminately. In this study, we introduce feature-level augmentation and propose a novel semantics-consistent feature search (SCFS) method to mitigate this negative effect. The main idea of SCFS is to adaptively search semantics-consistent features to enhance the contrast between semantics-consistent regions in different augmentations. Thus, the trained model can learn to focus on meaningful object regions, improving the semantic representation ability. Extensive experiments conducted on different datasets and tasks demonstrate that SCFS effectively improves the performance of self-supervised learning and achieves state-of-the-art performance on different downstream tasks.[1]

1. Introduction

Due to the tremendous potential in learning discriminative feature representation without using data annotations, self-supervised learning, including contrastive learning [16, 5] and masked image modeling (MIM) [1, 15, 38] has received much attention in the representation learning field. Contrastive learning, as a type of discriminative self-supervised learning method, is heavily studied and has shown remarkable progress in the computer vision field in recent years. It aims at pulling different augmented "views" of the same image (positive pairs) closer while pushing diverse images (negative pairs) far from each other. To this end, a contrastive loss between the features of different views extracted from an encoder network is employed to train the encoder network end-to-end. According to whether the negative pairs are used, current contrastive learning can be generally divided into two categories.

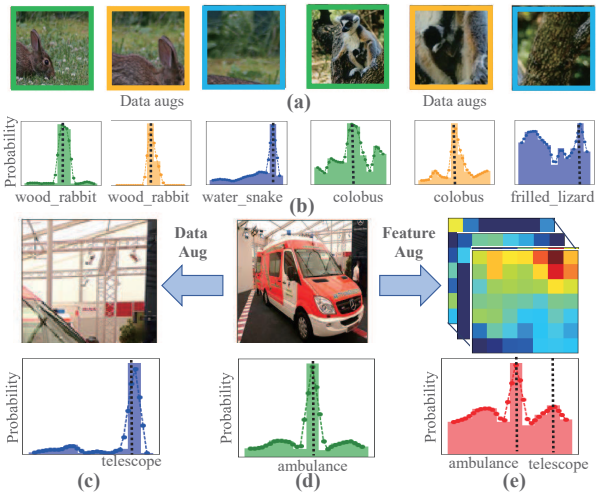

Figure 1. Semantic inconsistency of over-augmentation. (a) shows three augmentations of two images, in which the third augmentation is over-augmented and contains only background. (b) shows category probability distributions of the corresponding images in (a), which are obtained from a supervised pre-trained ResNet50 [18] model. (c)(d)(e) show three different samples of an image (the data-augmented image, the original image, and the semantics-consistent feature-augmented sample generated by Eq. (7) in this study) and their corresponding probability distributions, which point out that the over-augmented image generates different category with the original image, while the feature-augmented sample gets a balanced category probability.

The first category [16, 5] utilizes both positive pairs and negative pairs for contrast. MoCo [16, 7] uses a momentum update mechanism to maintain a memory bank of negative examples. SimCLR [5, 6] directly trains a single encoder network with a large batch size to ensure sufficient positive and negative samples for learning. Based on MoCo and Sim-

[1]Code: https://github.com/skyoux/scfs

CLR, some methods [24, 20, 36, 10, 49, 39, 19, 9, 48, 14] are proposed to improve the performance. For example, MSF [20], ISD [36] and NNCLR [10] aim to search semantics-consistent samples for contrast, solving the false negative problem. While some studies, such as Momentum2Teacher [24] and DCL [49], aim to solve the limitation that large batch size is necessary for satisfactory performance.

The second category of contrastive learning methods [13, 8, 3, 50, 2, 12, 4, 31, 35] only constructs positive pairs for contrast. Based on MoCo [16] and SimCLR [5], respectively, BYOL [13] and SimSiam [8] abandon the negative samples and use an asymmetric architecture to avoid model collapse. SwAV [3] uses online clustering to cluster samples and forces the consistency among cluster assignments of different augmentations. After that, some studies [4, 39, 10, 19] point out that enriching the augmented samples can improve the performance of contrastive learning. In addition, the study in [31] shows that improving the quality of positive augmented samples is important for self-supervised learning.

However, it is unavoidable to construct data augmentations containing different semantic concepts. Fig. 1(a) shows three augmentations of two images, in which the third augmentation is over-augmented and contains only the background. Fig. 1(b) shows category probability distributions of the corresponding images in (a), which are obtained from a supervised pre-trained ResNet50 [18]. We observed that the probability distribution of over-augmented images changes greatly compared with the first two augmentations, which indicates that the semantic information of the over-augmented images deviates from the normally-augmented images. Similar observation can be found in Fig. 1(c)(d). The original image in (d) shows a max probability for "ambulance", while the over-augmented image in (c) represents the different category "telescope". Due to such semantic inconsistency, conducting contrastive learning on these over-augmentations is harmful to representation learning. In this study, we found that semantics-consistent feature augmentation (Fig. 1(e), generated by Eq. (7)) can balance the original semantics "ambulance" and the over-augmented semantics "telescope", which can alleviate the influence of semantic inconsistency.

Motivated by this observation, we propose a novel semantics-consistent feature search (SCFS) method to alleviate the negative influence of semantic inconsistency in contrastive learning. SCFS utilizes the global feature of a view to adaptively search the semantics-consistent features of another view for contrast according to their similarity. It constructs informative feature augmentations and conducts contrast learning between feature augmentations and data augmentations. Thus, the pre-trained model can learn to focus on meaningful object regions to alleviate the negative influence of unmatched semantic alignment in current contrastive learning for better representation learning. In addition, the feature search is conducted on multiple layers of the backbone network, further enhancing the semantic alignment at different scales of features. Extensive experiments conducted on different datasets and tasks demonstrate that SCFS effectively improves the performance of self-supervised learning and achieves state-of-the-art performance on different downstream tasks. For example, it achieves state-of-the-art 75.7% ImageNet top-1 accuracy under the pre-training setting of 1024 batch size and 800 epochs for ResNet50.

The main contributions of this study are threefold:

- A novel contrastive learning method, i.e., SCFS, is proposed, and it can enhance semantic alignment in contrastive learning. To our knowledge, this is the first work that defines a feature search task in contrastive learning.
- We expand contrastive learning from a data-to-data manner to a feature-to-data manner, which enriches the diversity of augmentations.
- The proposed SCFS achieves state-of-the-art performance on different downstream tasks.

2. Related Works

Recently, some studies [33, 44, 42, 45, 46, 25, 47, 40, 23, 41] pointed out that the problem of semantic inconsistency is more serious for downstream dense prediction tasks, such as object detection and instance segmentation. Therefore, these methods utilize region-level and pixel-level features for contrast. In this study, the proposed SCFS construct feature-level augmentations using dense feature maps. Therefore, this section introduces related studies that conduct contrastive learning using region-level and pixel-level features.

Region-level contrastive learning. SCRL [33] minimizes the distance between two local features, which are cropped from two corresponding feature maps of two views. ReSim [44] aligns regional representations by sliding a fixed-sized window across the overlapping area between two views to improve the performance for localization-based tasks. SoCo [42], ORL [46], and UniVIP [25] extract object region proposals and use them to construct region-level features for contrastive learning. They achieve good performance for downstream dense prediction tasks.

Pixel-level contrastive learning. To obtain a more fine-grained representation, several studies [47, 40, 41] design pixel-level contrastive learning task, which assumes that features extracted from the same pixel of different views should be treated as positive pairs while pixels from others must be distinguished. PixPro [47] utilizes a pixel propagation module to select similar pixel features for contrast and encourages consistency between positive pixel pairs. DenseCL [40] proposes a dense projection head to generate dense

feature vectors for pixel-level contrastive learning. SetSim [41] is designed to realize pixel-wise similarity learning by filtering out noisy backgrounds.

As summarized above, data augmentations bring rich information while increasing uncertainty in contrastive learning. While methods that utilize region-level features expand the granularity of feature representation by alleviating the influence of noises. Recently, to solve the problem of semantic inconsistency in video contrastive learning, VITO [30] uses an MLP to learn attention masks for each view independently and fuses the attention pooling features from multi-layers for contrastive learning. Unlike VITO [30], SCFS uses a cross-view search to calculate attention and conducts contrastive learning at multi-layers independently. In addition, our method bridges the correlation between data and feature augmentations and extends the contrastive learning task to a semantics-consistent feature search task.

3. Methods

The overall architecture of SCFS is shown in Fig. 2. It consists of an encoder and a momentum encoder. The momentum encoder is an exponential-moving-average version of the encoder. SCFS consists of two contrastive learning tasks: the contrast between data augmentations ($\mathcal{L}_d$), which is designed based on the baseline DINO [4], and the contrast between data augmentations and feature augmentations ($\mathcal{L}_{fs}$). In this section, we first introduce the contrast between data augmentations in Sec. 3.1. Then, the key feature search module of SCFS is introduced in detail in Sec. 3.2. Next, we will introduce the overall loss of SCFS in Sec. 3.3. Finally, the implementation details are presented in Sec. 3.4.

3.1. Contrast Between Data Augmentations

Given a pair of global augmentations ($\boldsymbol{I}_1$ and $\boldsymbol{I}_2$) of an input image, the feature representations of the two augmentations are used to calculate the global contrastive loss. Specifically, $\boldsymbol{f}_1 = E_{\boldsymbol{\theta}}(\boldsymbol{I}_1)$ and $\boldsymbol{f'}_2 = E_{\boldsymbol{\theta}'}(\boldsymbol{I}_2)$, where $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$ are parameters of the encoder and the momentum encoder, respectively. $\boldsymbol{f}_1, \boldsymbol{f'}_2 \in R^K$, K is the output dimension. $\boldsymbol{f}_1$ is normalized with a softmax function:

$$P_1^i = \frac{\exp\left(f_1^i/\tau\right)}{\sum_{k=1}^{K}\exp\left(f_1^k/\tau\right)} \tag{1}$$

where $\tau > 0$ is a temperature parameter that controls the sharpness of the output distribution. Note that P_2' is obtained by normalizing $\boldsymbol{f'}_2$ with a similar softmax function with temperature τ'. $\boldsymbol{I}_1$ and $\boldsymbol{I}_2$ are fed to the momentum encoder and encoder symmetrically, and P_1' and P_2 are obtained respectively. Following DINO [4], the cross-entropy loss is employed as the contrastive loss between two global views:

$$\mathcal{L}_g = -\left(P_2' \log\left(P_1\right) + P_1' \log\left(P_2\right)\right) \tag{2}$$

To enrich augmentations, the multi-crop strategy [3] is employed. Multiple local augmentations $\boldsymbol{I}_l$ is also constructed and fed to the encoder: $\boldsymbol{f}_l = E_{\boldsymbol{\theta}}(\boldsymbol{I}_l)$. P_l is obtained by normalizing $\boldsymbol{f}_l$ with the softmax function with temperature τ. The contrast between local views and global views can be calculated:

$$\mathcal{L}_l = \sum_{n=1}^{N} -\left({P'}_1 \log\left(P_l^n\right) + {P'}_2 \log\left(P_l^n\right)\right) \tag{3}$$

where N denotes the number of local views. Thus, the overall loss is the sum of global loss and local loss:

$$\mathcal{L}_d = \mathcal{L}_g + \mathcal{L}_l \tag{4}$$

3.2. Semantics-Consistent Feature Search

As introduced in Sec. 1, it is unavoidable to construct augmentations that contain different semantic concepts during the augmentation procedure. It's harmful to pull these augmentations close indiscriminately in the feature space. Therefore, we propose the SCFS method to enhance the contrast between semantics-consistent regions in different augmentations.

The architecture of SCFS is shown in Fig. 2. By feeding the local augmentations $\boldsymbol{I}_l$ to the encoder, feature maps from different stages of the backbone ResNet50 [18] are extracted. Specifically, the output features from different stages, i.e., $Res2$, $Res3$ and $Res4$, are utilized to conduct SCFS, ensuring that each stage of the backbone produces discriminative features: $\left\{{}^2\boldsymbol{F}_l, {}^3\boldsymbol{F}_l, {}^4\boldsymbol{F}_l\right\} = E_{\boldsymbol{\theta}}(\boldsymbol{I}_l)$, where ${}^i\boldsymbol{F}_l \in R^{W_l^i \times H_l^i \times C^i}$, W_l^i, H_l^i, C^i denote the width, height and channel dimension, respectively. Next, the global average pooling operation is conducted on each ${}^i\boldsymbol{F}_l$ in the spatial dimensions:

$${}^i\boldsymbol{f}_l = \frac{1}{W_l^i \times H_l^i}\sum_{x=1}^{W_l^i}\sum_{y=1}^{H_l^i} {}^i\boldsymbol{F}_l(x,y,z) \tag{5}$$

where ${}^i\boldsymbol{f}_l \in R^{C^i}$. Meanwhile, the global augmentations $\boldsymbol{I}_g$ $(g = 1, 2)$ are fed to the momentum encoder to extract feature maps from different stages: $\left\{{}^2\boldsymbol{F}_g', {}^3\boldsymbol{F}_g', {}^4\boldsymbol{F}_g'\right\} = E_{\boldsymbol{\theta}'}(\boldsymbol{I}_g)$, ${}^i\boldsymbol{F}_g' \in R^{W_g^i \times H_g^i \times C^i}$, W_g^i, H_g^i, C^i denote width, height and channel dimension, respectively.

Then, based on ${}^i\boldsymbol{f}_l$ and ${}^i\boldsymbol{F}_g'$, SCFS aims to adaptively search the most semantics-consistent features in ${}^i\boldsymbol{F}_g'$ for contrast, while suppressing irrelevant features. In SCFS, each feature ${}^i\boldsymbol{f}_l$ of the local data augmentations is treated as query, and the features ${}^i\boldsymbol{F}_g'$ of the global augmentations are treated as keys. The similarity between ${}^i\boldsymbol{f}_l$ and ${}^i\boldsymbol{F}_g'$ is calculated:

$$\boldsymbol{A}(x,y) = \frac{{}^i\boldsymbol{f}_l \cdot {}^i\boldsymbol{F}_g'(x,y)}{\|{}^i\boldsymbol{f}_l\|_2 \left\|{}^i\boldsymbol{F}_g'(x,y)\right\|_2} \tag{6}$$

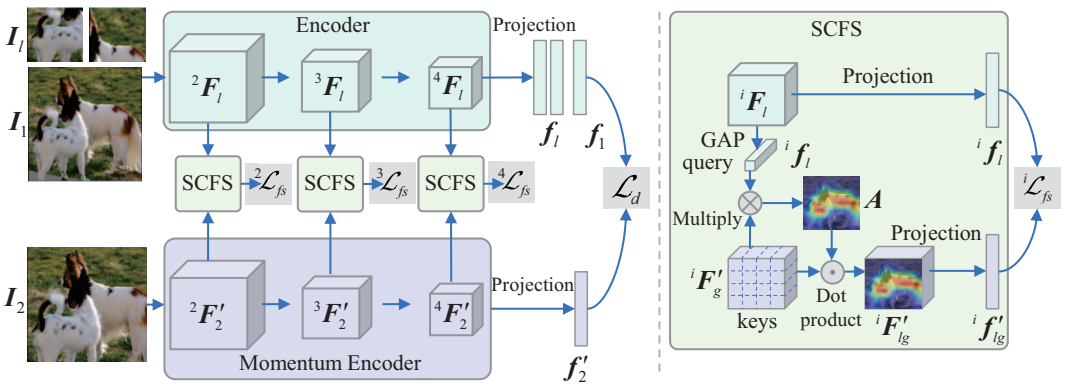

Figure 2. Overall architecture of the proposed semantics-consistent feature search (SCFS). It consists of an encoder and a momentum encoder. There are two contrastive learning tasks: the contrast between data augmentations ($\mathcal{L}_d$) in the final feature space and the feature search task conducted on multiple layers ($\mathcal{L}_{fs}$). The details of the feature search procedure is shown on the right.

where $\boldsymbol{A} \in R^{W_g^i \times H_g^i}$ is the attention map, and $x = 1, \ldots, W_g^i$, $y = 1, \ldots, H_g^i$, $\|\cdot\|_2$ is the L2 norm. The attention map $\boldsymbol{A}$ activates the semantics-consistent regions of the local augmentation on the global augmentation. Thus, the higher portion of local regions can be searched. To select semantic features and suppress irrelevant local features, we directly multiply the attention map $\boldsymbol{A}$ with ${}^i\boldsymbol{F}'_g$ to obtain the semantics-consistent feature augmentations:

$$ {}^i\boldsymbol{F}'_{lg} = \boldsymbol{A} \cdot {}^i\boldsymbol{F}'_g \tag{7} $$

This operation can be regarded as attention-weighted average pooling. Through feature search, N local data augmentations $\boldsymbol{I}_l$ can search N corresponding semantics-consistent features ${}^i\boldsymbol{F}'_{lg}$ from a global data augmentation $\boldsymbol{I}_g$. That is, in terms of the global data augmentation, N different features are constructed in the feature space through the feature search procedure. Therefore, we term ${}^i\boldsymbol{F}'_{lg}$ as feature-level augmentations.

Next, ${}^i\boldsymbol{F}_l$ and ${}^i\boldsymbol{F}'_{lg}$ are fed to corresponding projection heads to obtain their final representations for contrast:

$$ \begin{cases} {}^i\boldsymbol{f}_l = \mathrm{H}_i\left({}^i\boldsymbol{F}_l\right) \\ {}^i\boldsymbol{f}'_{lg} = H'_i\left({}^i\boldsymbol{F}'_{lg}\right) \end{cases} \tag{8} $$

where H_i and H'_i denote the projection heads on the i-th layer of the encoder and the momentum encoder, respectively. ${}^i\boldsymbol{f}_l$ and ${}^i\boldsymbol{f}'_{lg}$ are normalized with softmax function with temperature τ and τ', respectively, as the same formulation in Eq. (1). The corresponding output probability ${}^iP_l$ and ${}^iP'_{lg}$ are employed to calculate the contrast loss between local data augmentations and feature augmentations:

$$ {}^i\mathcal{L}_{fs} = \sum_{g=1}^{2} \sum_{n=1}^{N} - \left({}^iP'_{lg} \log\left({}^iP_l^n\right)\right) \tag{9} $$

The overall feature search loss is the sum of all layers:

$$ \mathcal{L}_{fs} = \sum_{i \in V_L} {}^i\mathcal{L}_{fs} \tag{10} $$

where V_L denotes the set of layers to conduct SCFS.

3.3. Overall Loss

The overall loss is the sum of the contrastive loss between data augmentations and the feature search loss:

$$ \mathcal{L} = \mathcal{L}_d + \mathcal{L}_{fs} \tag{11} $$

Through SCFS, the contrast between feature augmentations and data augmentations is bridged. The model can adaptively search the semantics-consistent features for contrast. Therefore, it can enhance the importance of semantics-consistent regions in different augmentations, alleviating the uncertainty in contrastive learning introduced by data augmentations that contain different semantic concepts.

3.4. Implementation Details

SCFS is based on DINO [4] and we follow the most hyper-parameter settings of DINO. For a fair comparison, the standard ResNet50 [18] is employed as the default backbone network.

For data augmentation, the global augmentations consist of random cropping, resizing to 224×224, random horizontal flip, gaussian blur, and color jittering. And the local augmentations consist of random cropping, resizing to 96×96, random horizontal flip, gaussian blur, and color jittering. For feature augmentations in SCFS, the $Res2$, $Res3$, and $Res4$ layers are used. Two global views with $N = 8$ local views are the default setting of augmentation.

The projection head for the contrast between data augmentations consists of a four-layer multi-layer-perceptron (MLP) with the same architecture as DINO [4]. The projection head for feature search consists of three convolutional layers and two FC layers.

Method	Batch Size	Epochs	LP	k-NN
Supervised	256	100	76.2	74.8
SimCLR [5]	4096	1000	69.3	-
BYOL [13]	4096	1000	74.3	66.9
BYOL [13]	4096	200	70.6	-
SwAV [3]	4096	800	75.3	-
SwAV [3]	256	200	72.7	-
MoCo-v2 [16]	256	200	67.5	54.3
SimSiam [8]	256	200	70.0	-
ISD [36]	256	200	69.8	62.0
MSF [20]	256	200	71.4	64.0
NNCLR [10]	4096	200	70.7	-
Barlow Twins [50]	2048	1000	73.2	-
VICReg [2]	2048	1000	73.2	-
OBoW [12]	256	200	73.8	-
DCL [49]	256	200	66.9	-
CLSA [39]	256	200	73.3	-
AdCo [19]	256	200	73.2	-
DetCo [45]	256	200	68.6	-
UniVIP [25]	4096	200	73.1	-
HCSC [14]	256	200	73.3	-
MoCo-v3 [9]	4096	300	72.8	-
MoCo-v3 [9]	4096	1000	74.6	-
DINO* [4]	256	200	73.0	64.0
DINO [4]	4080	800	75.3	67.5
SCFS	256	200	73.9	65.5
SCFS	1024	800	**75.7**	**68.5**

Table 1. Linear probing and k-NN accuracy (%) on ImageNet. The result with "*" is reproduced for fair comparison. LP denotes linear probing. Bold font and underline indicate the best results under the setting of 256 batch size and 200 epochs and the setting of 1024 batch size and 800 epochs, respectively.

Method	Batch Size	Epochs	Top-1		Top-5	
			1%	10%	1%	10%
Supervised [51]	256	90	25.4	56.4	48.4	80.4
SimCLR [5]	4096	1000	48.3	65.6	75.5	87.8
BYOL [13]	4096	1000	53.2	68.8	78.4	89.0
SwAV [3]	4096	800	53.9	70.2	78.5	89.9
DINO [4]	4080	800	50.2	69.3	74.0	89.1
SCFS	1024	800	**54.3**	**70.5**	**78.6**	**90.2**

Table 2. Evaluation on small labeled ImageNet. Bold font indicates the best result.

Method	Epochs	AP^b	AP^b_{50}	AP^b_{75}
Scratch	-	33.8	60.2	33.1
Supervised	90	53.5	81.3	58.8
SimCLR [5]	1000	56.3	81.9	62.5
BYOL [13]	300	51.9	81.0	56.5
SwAV [3]	400	45.1	77.4	46.5
DINO [4]	800	55.9	82.1	62.3
SCFS	800	**57.4**	**83.0**	**63.6**

Table 3. Results for PASCAL VOC object detection using Faster R-CNN [32] with ResNet50-C4. Bold font indicates the best result.

4. Experiments

In this section, comprehensive experiments are conducted to demonstrate the effectiveness of SCFS. We evaluate the performance on different downstream tasks, including ImageNet classification, object detection, instance segmentation, and other classification task on small datasets. In addition, we conduct ablation experiments to analyze the influence of each component in SCFS.

4.1. Comparing with SSL methods on ImageNet

k-NN and Linear Probing Accuracy on ImageNet. After pre-training on the ImageNet ILSVRC-2012 [34] training set, the pre-trained models are evaluated on the ImageNet ILSVRC-2012 validation set. For k-NN, it is evaluated as in study [43]. For linear probing, we train a linear classifier from scratch based on the feature extracted by a fixed backbone with 100 epochs [16]. The top-1 accuracy is adopted as the evaluation metric.

The results are reported in Tab. 1. With the standard ResNet50 [18] architecture and pre-trained with 256 batch size for 200 epoch, the proposed SCFS achieves the best k-NN top-1 accuracy 65.5% and the best linear probing top-1 accuracy 73.9%, outperforming its baseline DINO [4] by 1.5% and 0.9%, respectively. In addition, with 1024 batch size and 800 epoch, SCFS achieves the best k-NN accuracy (68.5%) and linear probing accuracy (75.7%), outperforming the accuracy of DINO [4] trained with 4080 batch size for 800 epoch. This result demonstrates that SCFS can improve the representation learning performance by searching semantics-consistent features for contrast.

Semi-Supervised Learning on ImageNet. In this part, we evaluate the performance of SCFS under the semi-supervised setting. Specifically, we use 1% and 10% of the labeled training data from ImageNet [34] for finetuning, which follows the semi-supervised protocol in SimCLR [5]. The same splits of 1% and 10% of ImageNet labeled training data in SimCLRv2 [6] are used. The results are reported in Tab. 2. After finetuning using 1% and 10% training data, SCFS outperforms all the compared methods. The results demonstrate that SCFS achieves the best feature representation quality.

4.2. Transfer Learning on Downstream Tasks

Object Detection and Instance Segmentation. In this part, we evaluate the representations of SCFS on dense prediction tasks, i.e., object detection and instance segmentation,

Method	Epochs	1×schedule						2×schedule					
		AP^b	AP^b_{50}	AP^b_{75}	AP^s	AP^s_{50}	AP^s_{75}	AP^b	AP^b_{50}	AP^b_{75}	AP^s	AP^s_{50}	AP^s_{75}
Scratch	-	31.0	49.5	33.2	28.5	46.8	30.4	38.4	57.5	42.0	34.7	54.8	37.2
Supervised	90	38.9	59.6	42.7	35.4	56.5	38.1	41.3	61.3	45.0	37.3	58.3	40.3
MoCo [16]	200	38.5	58.9	42.0	35.1	55.9	37.7	40.8	61.6	44.7	36.9	58.4	39.7
MoCo v2 [7]	200	40.4	60.2	44.2	36.4	57.2	38.9	41.7	61.6	45.6	37.6	58.7	40.5
BYOL [13]	300	40.4	61.6	44.1	**37.2**	58.8	**39.8**	42.3	62.6	46.2	**38.3**	59.6	**41.1**
SwAV [3]	400	-	-	-	-	-	-	42.3	62.8	**46.3**	38.2	60.0	41.0
ReSim-FPN^T [44]	200	39.8	60.2	43.5	36.0	57.1	38.6	41.4	61.9	45.4	37.5	59.1	40.3
SetSim [41]	200	40.2	60.7	43.9	36.4	57.7	39.0	41.6	62.4	45.9	37.7	59.4	40.6
DenseCL [40]	200	40.3	59.9	44.3	36.4	57.0	39.2	41.2	61.9	45.1	37.3	58.9	40.1
DSC [23]	200	39.4	58.9	43.2	35.7	56.1	38.3	-	-	-	-	-	-
HSA [48]	800	40.2	60.9	43.9	36.5	57.9	39.1	**42.2**	63.0	46.1	38.1	59.9	40.9
DetCo [45]	800	40.1	61.0	43.9	36.4	58.0	38.9	-	-	-	-	-	-
ORL* [46]	800	40.3	60.2	**44.4**	36.3	57.3	38.9	-	-	-	-	-	-
DINO [4]	800	40.0	61.6	43.4	36.5	58.6	39.1	41.9	62.6	46.0	37.8	59.7	40.6
SCFS	800	**40.5**	**61.8**	44.0	36.7	**58.8**	39.2	42.1	**63.4**	46.1	38.1	**60.2**	41.0

Table 4. Object detection and instance segmentation on COCO using Mask R-CNN [17] with ResNet50-FPN. Bold font indicates the best result. * indicates pre-training on COCO [26].

Method	CIFAR-10	CIFAR-100	CUB-Bird	Stanford-Cars	Aircraft	Oxford-Pets
Supervised	97.5	86.4	81.3	92.1	86.0	92.1
SimCLR[5]	97.7	85.9	–	91.3	88.1	89.2
BYOL[13]	97.8	86.1	–	91.6	88.1	91.7
DINO[4]*	97.7	86.6	81.0	91.1	87.4	91.5
SCFS	**97.8**	**86.7**	**82.7**	**91.6**	**88.5**	**91.9**

Table 5. Transfer learning results from ImageNet with the standard ResNet50 [18]. * denotes the results are reproduced in this study. Bold font indicates the best result.

on mainstream datasets PASCAL VOC [11] and MS COCO [26] datasets. On the PASCAL VOC dataset [11], the trainval07+12 set is used as the training set, and the test2007 set is used as the test set. Following [42], Faster R-CNN detector [32] with the ResNet50-C4 backbone initialized by the self-supervised pre-trained model is trained end-to-end. On the COCO dataset, the train2017 set is used for training and the val2017 set is used for evaluation. The Mask R-CNN [17] with R50-FPN is used. The AP^b, AP^b_{50} and AP^b_{75} metrics are used for object detection. While the AP^s, AP^s_{50} and AP^s_{75} metrics are used for instance segmentation.

The experimental results are shown in Tab. 3 and Tab. 4. SCFS achieves best performance on the two datasets. For example, on VOC, SCFS achieves 57.4% AP^b, 83.0% AP^b_{50} and 63.6% AP^b_{75}. The AP^b of SCFS outperforms its baseline DINO by 1.5%. These results shows that SCFS also has good transfer ability on dense prediction tasks.

Other Classification Tasks. In this part, we focus on the performance of self-supervised models when they are finetuned on small datasets, including CIFAR [22] and fine grained datasets [21, 37, 28, 29]. The results are shown in Tab. 5. The proposed SCFS shows the best performance on all the small datasets, which demonstrates that SCFS has good generalization ability.

4.3. Pre-training on Uncurated Dataset

The proposed SCFS can solve the problem of semantic inconsistency during pre-training, which is important when pre-training on uncurated datasets since this problem is more serious. To verify this, we pre-train SCFS and DINO on COCO [26], which is much more Uncurated than ImageNet. The same hyper-parameters used on ImageNet are applied to train the models with 512 batch size for 500 epochs. After pre-training, we fine-tune the pre-trained models on COCO for object detection and instance segmentation. The Mask R-CNN [17] with R50-FPN is used. As shown in Tab. 6, SCFS improves the performance significantly compared to its baseline DINO. In addition, compared to other dense pixel-level and region-level methods, such as DenseCL [40] and ORL [46], SCFS also achieves the best performance. This experiment verifies that SCFS can effectively solve the problem of semantic inconsistency during pre-training.

Method	k-NN	AP^b	AP^b_{50}	AP^b_{75}	AP^s	AP^s_{50}	AP^s_{75}
Scratch	-	31.0	49.5	33.2	28.5	46.8	30.4
Supervised*	-	38.9	59.6	42.7	35.4	56.5	38.1
SimCLR [5]	35.3	37.0	56.8	40.3	33.7	53.8	36.1
MoCov2 [7]	39.4	38.5	58.1	42.1	34.8	55.3	37.3
BYOL [13]	42.6	39.5	59.3	43.2	35.6	56.5	38.2
DenseCL [40]	32.5	39.6	59.3	43.3	35.7	56.5	38.4
ORL [46]	43.6	40.3	60.2	**44.4**	36.3	57.3	38.9
UniVIP [25]	-	40.8	-	-	36.8	-	-
DINO [4]	42.6	39.0	59.6	42.9	35.6	56.8	38.0
SCFS	**44.6**	**40.9**	**61.6**	**44.4**	**36.9**	**58.4**	**39.5**

Table 6. Pre-training and than Fine-tuning on COCO [26] using Mask R-CNN [17] with ResNet50-FPN and $1\times$ schedule. All models pre-trained on COCO are pre-trained with 512 batch size for 800 epochs. Bold font indicates the best result. * indicates pre-training on ImageNet [34]

4.4. Ablation Studies

We analyze the influence of each component in SCFS. To speed up the training time, the ImageNet100 dataset, which contains 100 randomly selected categories from ImageNet [34], is adopted. All the models are pre-trained on the ImageNet100 training set with 256 batch size for 200 epoch, and tested on the validation set. The k-NN and linear probing top-1 accuracy are used as the evaluation metrics.

Influence of Different Contrast Modes. The contrast mode can be divided into three types: contrast between two global data augmentations used in all contrastive learning methods (G_d2G_d); contrast between local data augmentations and global data augmentations used in multi-crop strategy (L_d2G_d); and contrast between local data augmentations and local feature augmentations used in SCFS (L_d2L_f).

The results are shown in Tab. 7. With multi-crop, DINO [4] (81.1%) improves accuracy by 3.0% compared to DINO without multi-crop baseline. SCFS (84.8%) further improves accuracy by 3.7% by introducing a contrast between local data augmentation and local feature augmentation. Some attention maps of SCFS and DINO are shown in Fig. 3. SCFS can more accurately focus on semantics-consistent regions between global view and local views, while DINO is easily influenced by background.

We also add multi-layer feature contrastive learning on DINO. The result in Tab. 7 (the "DINO w ML" row) verifies the improvements of SCFS are not totally owed to multi-layer contrast.

In addition, we directly crop the corresponding region of local augmentation on the feature map of global augmentation for contrastive learning. As shown in Tab. 7, this variant (ROIAlign) of SCFS also outperforms the DINO baseline, which shows that the directly cropped features are also beneficial for contrastive learning. And the ROIAlign variant

Contrast Mode	G_d2G_d	L_d2G_d	L_d2L_f	G_d2G_f	k-NN	LP
DINO w/o MC	✓				78.1	83.7
DINO	✓	✓			81.1	87.0
DINO w ML	✓	✓			82.2	87.4
SCFS	✓	✓	✓		84.8	89.2
ROI Crop	✓	✓	✓		83.9	88.1
SCFS w/o MC				✓	79.7	86.3

Table 7. Influence of different contrast modes. MC, ML, and LP denote multi-crop, multi-layer, and linear probing, respectively.

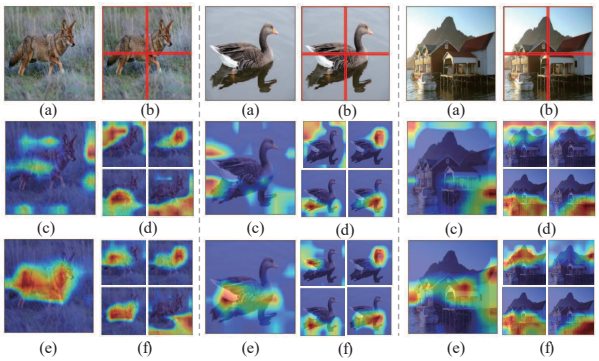

Figure 3. Attention maps of SCFS (the third row) compared with DINO [4] (the second row). In each example, (a) shows a global image, and its four local images in (b) are constructed by 2×2 jigsaw. (d) and (f) show the attention maps that highlight the semantics-consistent regions between the local images in (b) and the global image in (a). They are obtained by multiplying the globally average pooled feature maps from the encoder (Res4) of the local images in (b) with the feature map (Res4) of the global image in (a). And the encoder is the trained DINO ResNet50 model and SCFS ResNet50 model in (d) and (f), respectively. (c) and (e) show the mean attention maps of DINO and SCFS respectively, which are obtained by multiplying the mean globally average pooled feature map of the four local images in (b) with feature map of the global image in (a).

of SCFS achieves lower accuracy than SCFS, demonstrating that the soft feature search in SCFS is better than the hard ROIAlign since ROIAlign may damage the continuous semantic context of the feature map.

Further, we also test the performance of SCFS under the setting without multi-crop. That is, the feature search is conducted between two global data augmentations. We term this contrast mode as G_d2G_f. As shown in the "SCFS w/o MC" row, SCFS also improves the performance compared to its baseline (the "DINO w/o MC" row), which proves that SCFS is also helpful in solving the semantic inconsistency caused by other augmentations, not only the multi-crop augmentation strategy.

Influence of Multi-Layer Contrast. The influence of the feature layer that is used for feature search is analyzed. The $Res2$, $Res3$ and $Res4$ in the ResNet50 [18] backbone are evaluated. As shown in Tab. 8, the performance improves

Res2	Res3	Res4	k-NN	Linear Probing
✓			82.0	86.1
✓	✓		84.3	87.5
✓	✓	✓	84.8	89.2

Table 8. Influence of different feature augmentation layer.

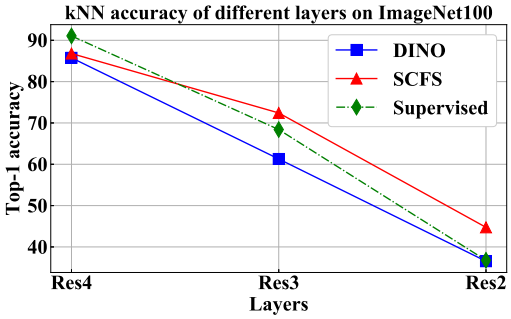

Figure 4. The k-NN accuracy of features from different layers.

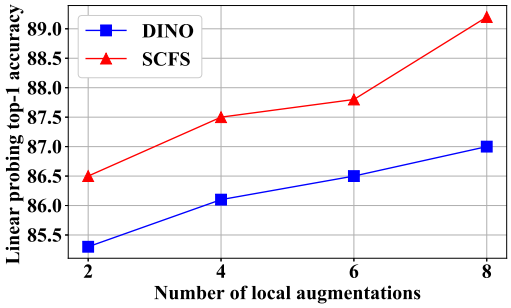

Figure 5. Influence of local augmentation number.

with the increase of feature layer numbers, which demonstrates that conducting feature search on more layers is helpful for representation learning.

Further, we evaluate the k-NN accuracy using feature maps from different layers to observe the influence of feature search on the representation of middle layers. We also choose the features extracted by the $Res2$, $Res3$ and $Res4$ layer of ResNet50. The results are shown in Fig. 4. Compared with DINO [4], SCFS achieves better performance with features from all middle layers on ImageNet100, which verifies that enhancing the semantic consistency can improve the semantic representation of shallow layers. Compared with supervised learning, SCFS has higher performance on res2 and res3 layer, which shows that SCFS is more advantageous in the shallow layer feature representation.

Influence of Local Augmentation Number. In this part, we analyze the performance difference with the change of local augmentation numbers. The results are shown in Fig. 5. The performance of DINO and SCFS is steadily improved when adding more local augmentations for contrast. In addition, SCFS improves the performance under different local augmentation numbers, which demonstrates that semantics-consistent feature search is helpful to alleviate the influence of semantics inconsistent data augmentations.

Cropping Scale	0.05-0.14	0.14-0.25	0.25-0.4	0.4-0.6
DINO [4]	87.0	87.2	85.6	82.6
SCFS	89.2	88.7	87.0	83.1

Table 9. Influence of random cropping scale. We report the linear probing top-1 accuracy.

Method	Backbone	Batch Size	Epochs	k-NN	LP
DINO [4]	R101	256	200	81.0	86.3
SCFS	R101	256	200	**85.1**	**88.3**
DINO [4]	ViT-S-16	256	200	75.0	80.4
SCFS	ViT-S-16	256	200	**76.3**	**81.0**
DINO [4]	ViT-B-16	256	200	76.2	80.7
SCFS	ViT-B-16	256	200	**77.2**	**82.3**

Table 10. Experiments on other backbones. LP denotes linear probing.

Influence of Random Cropping Scale. The hyperparameter of the random cropping scale influences the degree of semantic inconsistency. In this experiment, we analyze the performance of SCFS and DINO [4] under different random cropping scales. As shown in Tab. 9, owing to the feature search, SCFS is more adaptive to different cropping scales than the baseline DINO [4].

Experiments on Other Backbones. In this part, we conduct experiments on other backbones to further evaluate the effectiveness of SCFS. Apart for the default Reset50 used in other experiments, ResNet101 and Vision Transformer (ViT-S and ViT-B) are tested. The results are shown in Tab. 10. SCFS achieves significant improvement on different backbones compared to its baseline DINO, which demonstrates that SCFS is applicable to different backbones.

5. Conclusions

In this study, we aim to alleviate the problem of unmatched semantic alignment in current contrastive learning by expanding the augmentations from data space to feature space. The proposed semantics-consistent feature search (SCFS) adaptively searches semantics-consistent local features between different views for contrast, while suppressing irrelevant local features during pre-training. It conducts contrast learning between feature augmentation and data augmentation. The experimental results demonstrate that SCFS can learn to focus on meaningful object regions and effectively improve the performance of self-supervised learning. The feature search procedure in SCFS is learnable parameter-free. We will utilize the self-attention mechanism in Transformer to perform the feature search procedure to further boost its performance in future work.

References

[1] Hangbo Bao, Li Dong, and Furu Wei. Beit: Bert pre-training of image transformers. In *ICLR*, 2022. 1

[2] Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *arXiv preprint arXiv:2105.04906*, 2021. 2, 5

[3] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *NeurIPS*, 33:9912–9924, 2020. 2, 3, 5, 6, 12

[4] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, pages 9650–9660, 2021. 2, 3, 4, 5, 6, 7, 8, 11, 12

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, pages 1597–1607. PMLR, 2020. 1, 2, 5, 6, 7, 12

[6] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. *NeurIPS*, 33:22243–22255, 2020. 1, 5

[7] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 1, 6, 7

[8] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *CVPR*, pages 15750–15758, 2021. 2, 5

[9] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *ICCV*, pages 9640–9649, 2021. 2, 5, 12

[10] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. In *ICCV*, pages 9588–9597, 2021. 2, 5

[11] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, 2010. 6

[12] Spyros Gidaris, Andrei Bursuc, Gilles Puy, Nikos Komodakis, Matthieu Cord, and Patrick Perez. Obow: Online bag-of-visual-words generation for self-supervised learning. In *CVPR*, pages 6830–6840, 2021. 2, 5

[13] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *NeurIPS*, 33:21271–21284, 2020. 2, 5, 6, 7, 12

[14] Yuanfan Guo, Minghao Xu, Jiawen Li, Bingbing Ni, Xuanyu Zhu, Zhenbang Sun, and Yi Xu. Hcsc: Hierarchical contrastive selective coding. In *CVPR*, pages 9706–9715, 2022. 2, 5

[15] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, pages 16000–16009, 2022. 1

[16] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, pages 9729–9738, 2020. 1, 2, 5, 6

[17] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017. 6, 7

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 1, 2, 3, 4, 5, 6, 7

[19] Qianjiang Hu, Xiao Wang, Wei Hu, and Guo-Jun Qi. Adco: Adversarial contrast for efficient learning of unsupervised representations from self-trained negative adversaries. In *CVPR*, pages 1074–1083, 2021. 2, 5

[20] Soroush Abbasi Koohpayegani, Ajinkya Tejankar, and Hamed Pirsiavash. Mean shift for self-supervised learning. In *ICCV*, pages 10326–10335, 2021. 2, 5

[21] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013. 6

[22] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 6

[23] Xiaoni Li, Yu Zhou, Yifei Zhang, Aoting Zhang, Wei Wang, Ning Jiang, Haiying Wu, and Weiping Wang. Dense semantic contrast for self-supervised visual representation learning. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 1368–1376, 2021. 2, 6

[24] Zeming Li, Songtao Liu, and Jian Sun. Momentumˆ 2 teacher: Momentum teacher with momentum statistics for self-supervised learning. *arXiv preprint arXiv:2101.07525*, 2021. 2

[25] Zhaowen Li, Yousong Zhu, Fan Yang, Wei Li, Chaoyang Zhao, Yingying Chen, Zhiyang Chen, Jiahao Xie, Liwei Wu, Rui Zhao, et al. Univip: A unified framework for self-supervised visual pre-training. In *CVPR*, pages 14627–14636, 2022. 2, 5, 7

[26] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755. Springer, 2014. 6, 7

[27] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 11

[28] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 6

[29] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *CVPR*, pages 3498–3505. IEEE, 2012. 6

[30] Nikhil Parthasarathy, SM Eslami, João Carreira, and Olivier J Hénaff. Self-supervised video pretraining yields strong image representations. *arXiv preprint arXiv:2210.06433*, 2022. 3, 12

[31] Xiangyu Peng, Kai Wang, Zheng Zhu, Mang Wang, and Yang You. Crafting better contrastive views for siamese representation learning. In *CVPR*, pages 16031–16040, 2022. 2

[32] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *NeurIPS*, 28, 2015. 5, 6

[33] Byungseok Roh, Wuhyun Shin, Ildoo Kim, and Sungwoong Kim. Spatially consistent representation learning. In *CVPR*, pages 1144–1153, 2021. 2

[34] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015. 5, 7

[35] Kaiyou Song, Jin Xie, Shan Zhang, and Zimeng Luo. Multi-mode online knowledge distillation for self-supervised visual representation learning. In *CVPR*, 2023. 2

[36] Ajinkya Tejankar, Soroush Abbasi Koohpayegani, Vipin Pillai, Paolo Favaro, and Hamed Pirsiavash. Isd: Self-supervised learning by iterative similarity distillation. In *ICCV*, pages 9609–9618, 2021. 2, 5

[37] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 6

[38] Haochen Wang, Kaiyou Song, Junsong Fan, Yuxi Wang, Jin Xie, and Zhaoxiang Zhang. Hard patches mining for masked image modeling. In *CVPR*, 2023. 1

[39] Xiao Wang and Guo-Jun Qi. Contrastive learning with stronger augmentations. *IEEE TPAMI*, 2022. 2, 5

[40] Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li. Dense contrastive learning for self-supervised visual pre-training. In *CVPR*, pages 3024–3033, 2021. 2, 6, 7

[41] Zhaoqing Wang, Qiang Li, Guoxin Zhang, Pengfei Wan, Wen Zheng, Nannan Wang, Mingming Gong, and Tongliang Liu. Exploring set similarity for dense self-supervised representation learning. In *CVPR*, pages 16590–16599, 2022. 2, 3, 6

[42] Fangyun Wei, Yue Gao, Zhirong Wu, Han Hu, and Stephen Lin. Aligning pretraining for detection via object-level contrastive learning. *NeurIPS*, 34, 2021. 2, 6

[43] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, pages 3733–3742, 2018. 5

[44] Tete Xiao, Colorado J Reed, Xiaolong Wang, Kurt Keutzer, and Trevor Darrell. Region similarity representation learning. In *ICCV*, pages 10539–10548, 2021. 2, 6

[45] Enze Xie, Jian Ding, Wenhai Wang, Xiaohang Zhan, Hang Xu, Peize Sun, Zhenguo Li, and Ping Luo. Detco: Unsupervised contrastive learning for object detection. In *ICCV*, pages 8392–8401, 2021. 2, 5, 6

[46] Jiahao Xie, Xiaohang Zhan, Ziwei Liu, Yew Soon Ong, and Chen Change Loy. Unsupervised object-level representation learning from scene images. *NeurIPS*, 34:28864–28876, 2021. 2, 6, 7

[47] Zhenda Xie, Yutong Lin, Zheng Zhang, Yue Cao, Stephen Lin, and Han Hu. Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning. In *CVPR*, pages 16684–16693, 2021. 2

[48] Haohang Xu, Xiaopeng Zhang, Hao Li, Lingxi Xie, Wenrui Dai, Hongkai Xiong, and Qi Tian. Seed the views: Hierarchical semantic alignment for contrastive representation learning. *IEEE TPAMI*, 2022. 2, 6

[49] Chun-Hsiao Yeh, Cheng-Yao Hong, Yen-Chi Hsu, Tyng-Luh Liu, Yubei Chen, and Yann LeCun. Decoupled contrastive learning. In *ECCV*, pages 668–684. Springer, 2022. 2, 5

[50] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *ICML*, pages 12310–12320. PMLR, 2021. 2, 5

[51] Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov, and Lucas Beyer. S4l: Self-supervised semi-supervised learning. In *ICCV*, pages 1476–1485, 2019. 5