Exploring Set Similarity for Dense Self-supervised Representation Learning

Zhaoqing Wang[1] Qiang Li[2,*] Guoxin Zhang[2] Pengfei Wan[2]
Wen Zheng[2] Nannan Wang[3] Mingming Gong[4] Tongliang Liu[1]
[1]University of Sydney [2]Kuaishou Technology [3]Xidian University [4]University of Melbourne
derrickwang005@gmail.com; {liqiang03,zhangguoxin,wanpengfei,zhengwen}@kuaishou.com
nnwang@xidian.edu.cn; mingming.gong@unimelb.edu.au; tongliang.liu@sydney.edu.au

Abstract

By considering the spatial correspondence, dense self-supervised representation learning has achieved superior performance on various dense prediction tasks. However, the pixel-level correspondence tends to be noisy because of many similar misleading pixels, e.g., backgrounds. To address this issue, in this paper, we propose to explore ***set similarity*** *(**SetSim**) for dense self-supervised representation learning. We generalize pixel-wise similarity learning to set-wise one to improve the robustness because sets contain more semantic and structure information. Specifically, by resorting to attentional features of views, we establish the corresponding set, thus filtering out noisy backgrounds that may cause incorrect correspondences. Meanwhile, these attentional features can keep the coherence of the same image across different views to alleviate semantic inconsistency. We further search the cross-view nearest neighbours of sets and employ the structured neighbourhood information to enhance the robustness. Empirical evaluations demonstrate that SetSim surpasses or is on par with state-of-the-art methods on object detection, keypoint detection, instance segmentation, and semantic segmentation.*

1. Introduction

Pretraining has become a widely-used paradigm in various computer vision tasks. Generally, models are first pretrained on large-scale datasets (e.g., ImageNet [11]) and fine-tuned on the specific tasks. Recently, self-supervised pretraining has broken the dominance of the supervised ImageNet [11] pretraining on almost all downstream tasks including image classification [21], object detection [34], semantic segmentation [14, 15], etc. In particular, state-of-the-art self-supervised representation learning methods [3, 5, 8, 17] mainly adopt the *instance discrimination* formulation as their pretext task to obtain transfer learning ability for downstream tasks. The main idea of these methods is to maximize the similarity of two data-augmented views of the same image, while minimizing the similarity of views generated from different images.

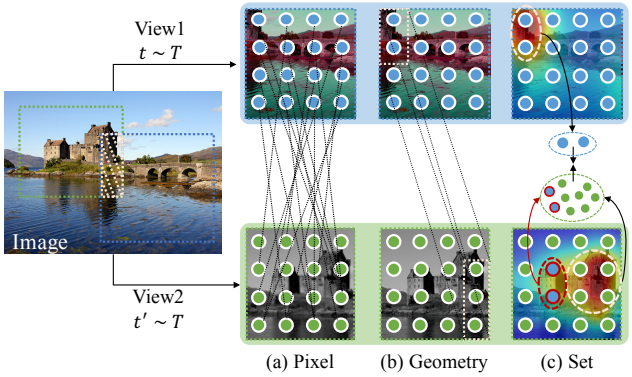

Figure 1. **The comparison of existing pixel-wise correspondence with our proposed method.** (a) Pixel-based: compare all combinations of pixel-wise features and maximize the most similar pairs. (b) Geometry-based: force features in overlapping regions to remain constant and distinguish features from different locations. (c) Set-based: considering misleading features and semantic inconsistency in the former methods, we propose to explore set similarity across two views for dense self-supervised representation learning. By resorting to attentional features, SetSim constructs corresponding sets across two views (blue point & green point), which can filter out misleading features and keep the coherence of the same image across views. Furthermore, SetSim searches the cross-view nearest neighbours (blue point with red circle) to enhance the structured neighbourhood information.

Since most of the self-supervised methods are designed for image-level tasks like image classification, they are sub-optimal for dense prediction tasks, e.g., object detection and semantic segmentation. To narrow this performance gap, dense self-supervised learning has been explored recently [32, 40, 46]. A popular way is to leverage intrinsic spatial information that a matching pair of pixel-wise features should remain constant over different viewing condi-

*Corresponding author.

tions. DenseCL [40] compares all combinations of pixel-wise features across two views and picks out the most similar pairs as spatial correspondences. Besides, VADeR [32] and PixPro [46] adopt geometry-based correspondence, which means that two views' pixel-wise features from the same location of the same image are treated as positive pairs, while features obtained from different locations are treated as negative pairs.

Nonetheless, in general, the pixel-wise correspondence based on similarity or geometry information is more likely to be noisy. Specifically, if there is a pixel-wise feature vector, there would be many misleading features similar to it, thus causing incorrect correspondences, which are illustrated in Figure 1(a). Besides, in Figure 1(b), the overlapping region of two views splits similar semantic features into two parts, which are pushed far in the embedding space by a learning objective, resulting in spatial semantic inconsistency. Although recently proposed methods have shown promising transfer performance improvements on dense prediction tasks, the issue of establishing robust correspondence remains unsolved, therefore, we are motivated to seek further exploration on this issue in order to apply it to our research.

In this paper, we propose to explore **set sim**ilarity (SetSim) across views for dense self-supervised representation learning. Considering that a set of pixel-wise features can represent more semantic and structure information than individual counterpart, we generalize pixel-wise similarity learning to set-wise one to improve the robustness. In particular, based on the attention map, we construct the corresponding set, which contains pixels of two different views with similar semantic information. As shown in Figure 1(c), attention maps are able to reveal the salient objects, *i.e.*, castle, in two views of the input image, which effectively keep the coherence of the input image. Since certain useful pixels are excluded from the set, we further search the cross-view nearest neighbours of one view's set and enhance the structured neighbourhood information. Finally, the model maps the pixels in the corresponding set to similar representations in the embedding space by a contrastive [18, 24] or cosine similarity [8] optimization function.

The proposed SetSim outperforms or is on par with the state-of-the-art methods on various dense prediction tasks, including object detection, keypoint detection, instance segmentation, and semantic segmentation. Compared to the MoCo-v2 baseline, our method can significantly improve the localization and classification abilities on dense prediction tasks: $+2.1AP^b$ on VOC object detection, $+1.3AP^b$ on COCO object detection, $+0.4AP^{kp}$ on COCO keypoint detection, $+1.0AP^m$, $+2.2AP^m$ on COCO and Cityscapes instance segmentation, $+2.6mIoU$ on VOC object segmentation, $+1.7mIoU$, $+1.6mIoU$ on ADE20K and Cityscapes semantic segmentation, respectively.

2. Related Work

Self-supervised representation learning. Self-supervised representation learning is a kind of unsupervised representation learning, which has received extensive attention in recent years. It leverages the intrinsic structure of data as a supervisory signal for training and learns informative and transferable representations for downstream tasks. Early self-supervised learning approaches consist of a wide range of pretext tasks, including denoising [39], inpainting [33], colorization [23, 47], egomotion prediction [1], and so on [12, 30, 48]. Besides, a series of high-level pretext tasks are under research, such as rotation [16], jigsaw puzzles [31] and temporal ordering [29]. However, these methods achieved minimal success in computer vision.

Recently, contrastive self-supervised learning has emerged as a promising approach to unsupervised visual representation learning. The breakthrough one is SimCLR [6], which adopts the *instance discrimination* formulation as its pretext task. It generates two views of each image by a diverse set of data augmentations and maximize the similarity of two augmented views from the same image, meanwhile, minimizing similarities with a large set of views from other images. Besides, MoCo [7, 18] introduces a momentum encoder to improve the consistency of a queue of negative samples and achieves remarkable performance. Shortly after that, another category of methods based on clustering [2–5] is proposed, which alternates between clustering feature representations and learning to predict the cluster assignment. More recently, BYOL [17] and SimSiam [8] are came up with directly predicting the output of one view from another view without consideration of negative samples. Nonetheless, image-level self-supervised pretraining can be sub-optimal for dense prediction tasks due to the discrepancy between image-level and pixel-level prediction.

Dense self-supervised representation learning. Image-level supervised and self-supervised pretraining has achieved encouraging results on a series of downstream tasks, including image classification, object detection, semantic segmentation and so on. Nonetheless, previous studies [19, 35, 36] demonstrate that there is a transfer gap between image-level pretraining and dense prediction tasks. Recently, several related approaches [32, 40, 46] are proposed to explore dense self-supervised representation learning. They generalize the instance discrimination from image-level to pixel-level. To be specific, the positive pairs of local features across views are defined by pixel-level correspondences and features excluded from the correspondence are treated as the negative pairs. In particular, DenseCL [40] compares all combinations of feature vectors and pulls the most similar pairs closer, which is similar to clustering. By utilizing parameters of the affine transformation, VADeR [32] and PixPro [46] map corresponding pixel-wise features in each view to their associated features. However, pixel-wise correspondence is

likely to be noisy because there is a wide range of highly misleading pixel-wise features [24, 43, 44], while the geometric correspondence is difficult to capture the coherence between two views of the input image. In this work, we introduce set similarity for dense self-supervised representation learning to improve the robustness.

3. Methodology

In this section, we first revisit the instance discrimination task and its general pipeline for self-supervised representation learning. Subsequently, we present a detailed description of our proposed SetSim, *i.e.*, a dense self-supervised learning framework based on set similarity.

3.1. Preliminaries

Instance discrimination is a widely-used pretext task for self-supervised visual representation learning [6, 7, 18, 42]. Given an unlabeled dataset, the input image I is augmented by a series of pre-defined data augmentations $T = [T_1, T_2, ..., T_n]$. By sampling $t \sim T$ and $t' \sim T$, we can generate the query view $I^q = t(I)$ and key view $I^k = t'(I)$. For each view, an encoder is adopted to extract image-level features p_{img}. The encoder consists of two main components, the backbone f and the projector g. Notice that only the backbone is transferred to downstream tasks after the pretraining process. Subsequently, a contrastive loss is adopted to pull each encoded query p^q_{img} close to its positive encoded key $p^{k_+}_{img}$ and away from its negative encoded keys $p^{k_-}_{img}$:

$$\mathcal{L}_{img} = -\log \frac{\exp\left(p^q_{img} \cdot p^{k_+}_{img}/\tau\right)}{\sum_{p^k_{img}} \exp(p^q_{img} \cdot p^k_{img}/\tau)}, \tag{1}$$

where encoded features p_{img} are L2-normalized, and τ is a temperature hyper-parameter [42].

3.2. Architecture Overview

As illustrated in figure 2, the proposed SetSim framework mainly consists of four parts: a backbone, two projectors, a matcher, and a queue. Firstly, SetSim augments an input image into two data-augmented views, I^q and I^k. Then, each view's deep feature are generated by the backbone network f (a ResNet-50 [21] is used by default), and are fed into two parallel projectors g_{img} and g_{set}. To maintain the basic architecture, we keep the design of the set-level projector as simple as the image-level one. Specifically, the image-level projector g_{img} is composed of two fully connected layers with a ReLU layer between them, and the set-level projector g_{set} is composed of two 1×1 convolution layers with a ReLU layer between them. Upon the attended transformed features, SetSim employs a matcher to establish two corresponding sets spatially across two views. Finally, we adopt a standard contrastive loss function [18] for image-level optimization and a modified contrastive loss function for set-level optimization.

3.3. Set Similarity Dense Representation Learning

Constructing Corresponding Set. With the help of image-level contrastive loss, attention maps of the top layers can reflect some salient regions (e.g., objects or stuff), which is crucial to alleviate both effects of misleading pixel-wise features and semantic inconsistency. In particular, for each data-augmented view from the same input image I, the backbone f extracts feature maps $z \in \mathbb{R}^{C \times HW}$ and the convolutional projector g_{set} generates feature maps $p \in \mathbb{R}^{C' \times HW}$, which is formulated as:

$$p = g_{set}(z), \quad z = f(I), \tag{2}$$

where the feature maps before and after projection have different channel dimension C and C', respectively. To construct the corresponding set, we first obtain the spatial attention map A by computing statistics of feature maps z across the channel dimension C, which is formulated as:

$$A = \sum_{i=1}^{C} |z_i|, \tag{3}$$

where $z_i = z(i, :)$ and the absolute value operation is element-wise. Then, we employ a relative selection strategy to append attentional vectors into the corresponding set, which uses Min-Max normalization for rescaling A and introduces a threshold δ for selecting vector p_j, which is defined as:

$$A' = \frac{A - \min(A)}{\max(A) - \min(A)}, \tag{4}$$

$$\Omega = \{j : A'(j) \geq \delta\}, \tag{5}$$

where A' is the resacled attention map, j is the spatial index of feature maps p thus $p_j \in \mathbb{R}^{C'}$. We conduct the discussion of δ in the following experiment section. Finally, attentional feature vectors p_j can be adaptively appended in the set Ω.

Set2Set-NN Matching Strategy. By resorting to attention maps, SetSim generates the corresponding set of attentional feature vectors from two views. For simple illustration, we assume that the numbers of attentional vectors of the query and key view are m and n, respectively. For each attentional query vector p^q_i, we first establish fully-connected correspondences s_i with each attentional key vectors p^k_i.

Due to the threshold δ selection, some useful vectors could be excluded from the corresponding set. So, we further search the nearest neighbour of p^q_i from the key view, which can enhance the structured neighbourhood information. Specifically, for each vectors in p^q_i, its associated nearest neighbour can be obtained by applying an `argmax` operation to the similarity of z^q and z^k, which is formulated as:

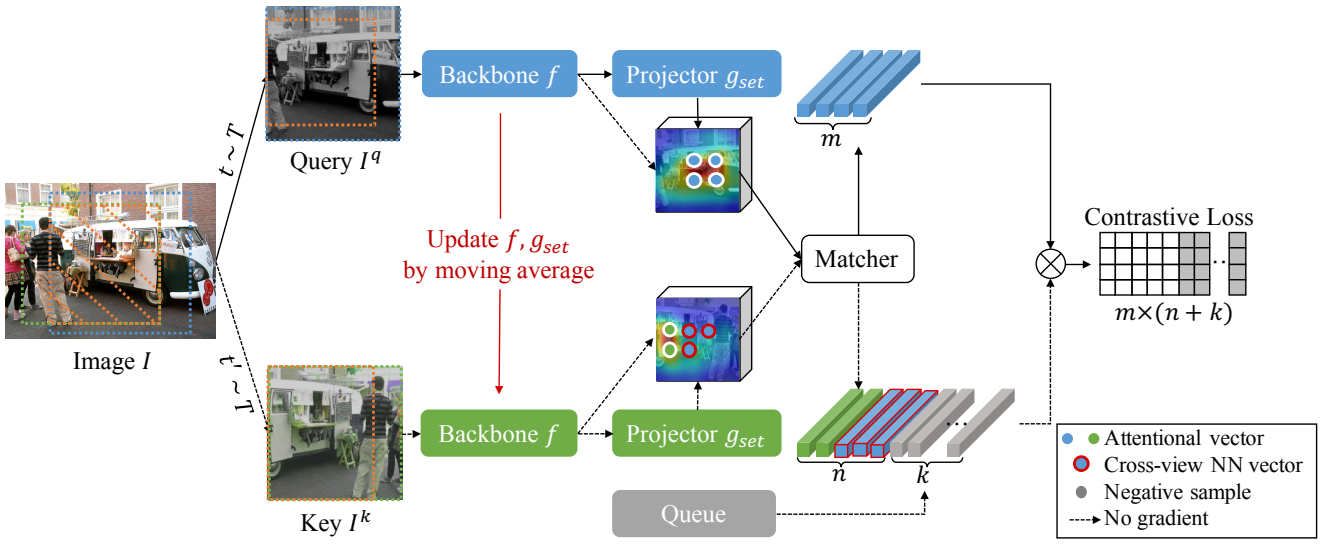

Figure 2. **An overview of the architecture of the proposed SetSim.** The SetSim architecture has two branches, including an encoder and its momentum-updated one. SetSim takes two augmented views of an image as inputs, I^q and I^k. For each view, the backbone f is adopted to extract visual features, and a convolutional projector g_{set} is adopted to generate transformed feature maps. SetSim introduces a matcher to construct corresponding sets and define the correspondence between two views. Finally, SetSim is optimized by an image-level and a set-level contrastive loss in an end-to-end manner. For brevity, the image-level branch [18] is not shown in the figure.

$$n_i = \arg\max_j sim(z_i^q, z_j^k),\ z_i^q \in z^q,\ z_j^k \in z^k, \quad (6)$$

where $sim(u, v) = u^\top v/\|u\|\|v\|$ denotes the cosine similarity, i is the spatial index of attentional query features p_i^q, j is the spatial index of key features z_j^k, and n_i means the obtained nearest neighbour of p_i^q.

Finally, we get the evolved corresponding set c_i for each p_i^q, which is formulated as:

$$c_i = s_i \cup n_i. \quad (7)$$

Similarity Learning Objectives. As illustrated in Figure 2, given attentional query vectors p_i^q and the corresponding set c_i, the positive pairs can be directly obtained and negatives k_- is provided by a queue of global average-pooled feature in the key view. The set-level contrastive loss is calculated as:

$$\mathcal{L}_{set} = \sum_i \frac{-1}{|c_i|} \sum_{j \in c_i} \log \frac{\exp(p_i^q \cdot p_j^k/\tau)}{\sum_{j \in c_i} \exp(p_i^q \cdot p_j^k/\tau) + \sum_{k_-} \exp(p_i^q \cdot k_-/\tau)}, \quad (8)$$

where the $\cdot$ symbol denotes the inner product, $|c_i|$ is a cardinality of c_i, and p_j^k is the attentional vector in the key view. τ is a temperature hyper-parameter. Following [7], we set $\tau = 0.2$ as default. Overall, the total loss in our framework can be formulated as follow,

$$\mathcal{L} = (1-\lambda)\mathcal{L}_{img} + \lambda \mathcal{L}_{set}. \quad (9)$$

where λ is a hyper-parameter to balance two terms, which is set to 0.5 [40].

4. Experiments

We conduct the experiments of self-supervised pretraining on two type of ImageNet dataset [11]: (a) IN-1K contains ~1.25M images, (b) IN-100 [37] is a subset of ImageNet-1K containing ~125K images. Subsequently, we evaluate the transfer performance on various dense prediction tasks. In particular, the pretrained model is fine-tuned on PASCAL VOC [13] for object detection and semantic segmentation, COCO [26] for object detection, instance segmentation and keypoint detection, Cityscapes [10] for semantic segmentation and instance segmentation, and ADE20K [49] for semantic segmentation. Considering the efficiency, the ablation study is conducted on the IN-100 dataset. Following the common protocol [7, 18], we report the 200-epoch pretrained model to compare with state-of-the-art methods. Note that all the comparing 200-epoch pretrained weights are downloaded from their official releases respectively except for PixPro, which didn't offer the 200-epoch pretrained weights, is re-trained by using their official code.

Pretraining Setting. Following the setting in [7, 8, 18], we utilize SGD as our optimizer with initial learning rates of 0.03 for MoCo-v2 and 0.1 for SimSiam. The learning rate is updated under the cosine decay scheduler [6, 28]. The weight decay and momentum in SGD are set as 0.0001 and 0.9, respectively. Shuffling BN is adopted for MoCo, and synchronized BN is adopted for SimSiam during pre-training. Each pre-trained model is trained on 8 Tesla V100 GPUs with the batch size of 256. Compared with the baselines,

Table 1. **Building SetSim on various self-supervised learning frameworks.** All methods are pretrained for 200 epochs on the IN-100 dataset and fine-tuned on PASCAL VOC object detection. The significant improvements indicate that our method is applicable to multiple frameworks. (Average over 5 trials)

method	AP^b	AP^b_{50}	AP^b_{75}
MoCo-v2	54.3	80.3	60.2
+ SetSim	**56.1** (+1.8)	**81.6** (+1.3)	**62.5** (+2.3)
SimSiam	54.5	80.4	60.5
+ SetSim	**55.8** (+1.3)	**81.2** (+0.8)	**62.0** (+1.5)

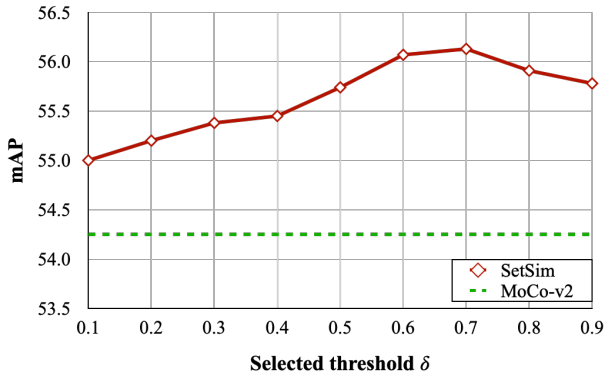

Figure 3. **Ablation study of selected threshold.** Each model is pretrained on the IN-100 dataset for 200 epochs and fine-tuned on PASCAL VOC object detection. (Average over 5 trials)

Table 2. **Ablation study of matching strategy.** All methods are pretrained for 200 epochs on the IN-100 dataset and fine-tuned on PASCAL VOC object detection. Note that the first three are pixel-to-pixel matching strategies, so the number of selected pixels across two views needs to remain the same. (Average over 5 trials)

strategy	AP^b	AP^b_{50}	AP^b_{75}
Random	54.9	80.6	60.6
Sort	55.4	81.0	61.1
Hungarian	55.6	81.1	61.5
Set2Set	56.0	81.4	61.9
Set2Set $-$ *NN*	**56.1**	**81.6**	**62.0**

Table 3. **Ablation study of the combination of correspondences.** "*Set*" and "*Geo*" denote the set-based and geometry-based correspondences, respectively. "*Sym*" denotes the symmetrized loss [8]. Each model is pretrained on the IN-100 dataset for 200 epochs and fine-tuned on PASCAL VOC object detection. The baseline is MoCo-v2.

Set	*Geo*	*Sym*	AP^b
			54.3
✓			55.3 (+1.0)
	✓		55.2 (+0.9)
✓	✓		56.0 (+1.7)
✓	✓	✓	**56.1** (+1.8)

our method slightly increases the training time. Note that our method does not cause any extra computational cost on following downstream tasks.

4.1. Ablation Study

We first build our method on various self-supervised learning baseline to evaluate the effectiveness and expansibility. To study each component in SetSim, we conduct extensive experiments with different setting and visualize the learned set-wsie correspondences for qualitative analysis. Due to the extra training time caused by SimSiam [8], we finally adopt the MoCo-v2 [7] as our base framework to compare with the state-of-the-art methods on various dense prediction tasks.

Experimental Setting. In this section, we fine-tune each pre-trained model on the widely-used PASCAL VOC object detection [13]. Following [18], we train a Fast R-CNN detector (C4-backbone) [34, 41] on `trainval07+12` set (~16.5k images) with standard 24k iterations, and evaluate on `test2007` set (~4.9k images). During the training process, the short-side length of input images is randomly selected from 480 to 800 pixels and fixed at 800 for inference. Similar to [7, 18], we fine-tune all Batch Normalization layers and adopt synchronized version during training. All results are averaged over five trials to overcome the randomness.

Comparisons with Baselines. Firstly, we compare our method with the MoCo-v2 baseline. Note that the overall architecture is similar to MoCo-v2, ensuring a fair comparison. As shown in Table 1, SetSim can effectively explore spatial representations across two views' convolutional features. The most significant performance gap occurs at AP^b_{75}, a high AP metric, which demonstrates that SetSim can improve the localization capability. Besides, we also build SetSim on a recently proposed self-supervised learning framework, SimSiam, without considering negative samples. Following [8], we keep SimSiam's design of the projector and predictor for image-level representation learning. And, the convolutional projector and predictor are parallel with the image-level part for dense representation learning. As illustrated in Table 1, SetSim surpasses the MoCO-v2 baseline by a large margin of 1.8, 1.3, 2.3 at AP^b, AP^b_{50}, and AP^b_{75}, which powerfully demonstrates the effectiveness and expansibility of our method. Meanwhile, we observe the similar phenomenon in the SimSiam experiment.

Matching Strategy. As illustrated in Table 2, we compare five different matching strategies. The first three are pixel-to-pixel matching strategies. (1) *Random*: features from two

Table 4. **Comparisons with the state-of-the-art approaches on PASCAL VOC object detection.** A Faster R-CNN (R50-C4) [34, 41] is trained on `trainval07+12`, evaluated on `test07`. Each model is pretrained for 200 epochs. (Average over 5 trials)

Method	AP^b	AP^b_{50}	AP^b_{75}
Random init.	33.8	60.2	33.1
IN-1K sup.	53.5	81.3	58.8
SimCLR [6]	51.5	79.4	55.6
MoCo-v2 [7]	57.0	82.3	63.3
BYOL [17]	55.3	81.4	61.1
SimSiam [8]	56.4	82.0	62.8
SwAV [5]	55.4	81.5	61.4
InfoMin [38]	57.5	82.5	64.0
DenseCL [40]	58.7	82.8	65.2
PixPro [46]	**59.4**	83.1	**66.9**
ReSim-C4 [45]	58.7	83.1	66.3
SetSim	59.1	**83.2**	66.1

Table 5. **Comparisons with the state-of-the-art approaches on Cityscapes, PASCAL VOC, and ADE20K semantic segmentation.** A FCN (R50) [9, 27] is adopted for all methods. Following the setting of DenseCL [40], we fine-tune all methods with their official pretrained weights. Each model is pretrained for 200 epochs. (Average over 5 trials)

Method	mIoU		
	Citys	VOC	ADE
Random init.	65.1	40.7	29.4
IN-1K sup.	74.0	67.5	35.9
MoCo-v1 [18]	74.5	66.2	37.0
MoCo-v2 [7]	75.4	67.3	36.9
SwAV [5]	73.2	65.2	36.7
DenseCL [40]	75.9	68.9	38.1
PixPro [46]	76.0	70.3	38.3
ReSim-C4 [45]	75.8	68.4	37.9
SetSim	**77.0**	**70.9**	**38.6**

views are randomly matched. (2) $Sort$: two views' features are sorted from large to small based on their attention values, then matched one by one. (3) $Hungarian$: features from two views are matched by the Hungarian algorithm [22], where we set the cosine distance as the cost matrix. $Random$ can obtain 0.6% AP^b gains compared to MoCo-v2, which can be explained as the convolutional projector is able to keep the spatial information and some random correspondences are correct. $Sort$ and $Hungarian$ bring more improvements than the former, demonstrating that these two strategies can effectively establish more adequate pixel-wise correspondences across two views. $Set2Set$ performs the better AP^b of 56.0% than three pixel-to-pixel strategies, because $Set2Set$ is able to learn more spatial-structured information by pulling closer two corresponding sets of pixels across views. Compared to pixel-wise similarity learning, sets-level representation learning facilitates exploring robust visual representation because the set contains more semantic and structure information. Finally, $Set2Set - NN$ can further improve AP^b and AP^b_{50} by 0.1% and 0.2%, because NN can recycle useful features excluded from sets and enhances the spatial-structured information.

Selected threshold δ. We explicitly control the selected threshold δ of attentional vectors to study the effects. Specifically, we raise δ from 0.1 to 0.9 to limit the number of selected attentional feature during pixel-level representation learning. As shown in Figure 3, we can observe a trend that increasing δ from 0.1 to 0.7 brings significant gains of 1.13% AP^b. It is explainable that the lower δ leads to the participation of more attentional vectors, which are not related to salient objects, in the corresponding set, thus establishing incorrect spatial correspondences. As the training progresses, the model gradually memorizes these incorrect correspondences, which causes the limitation of transfer ability on downstream tasks. Besides, the performance slightly drops 0.35% AP^b if we further increase δ to 0.9, because too few attentional vectors are adopted to learn sufficient visual representations.

Combination of Spatial Correspondence. As shown in Table 3, we further investigate the effect of different type of spatial correspondence. It can be seen that both "Set" and "Geo" [46] can achieve better transfer performance than MoCo-v2 (55.3 and 55.2 *v.s.* 54.3). When two types of spatial correspondence are adopted simultaneously, the model can obtain significant gains of 1.8% than the baseline, which indicates that the two types of spatial correspondence are complementary. Specifically, "Set" exploits sets' semantic and spatial-structured information, and "Geo" can force geometry-corresponding local representations to remain constant over different viewing conditions. Finally, we adopt the symmetrized loss [8] to improve the performance, and choose this as the default for SetSim.

Qualitative Analysis. As illustrated in Figure 4, we visualize the learned corresponding sets between two views of input images. The visualization shows that SetSim can accurately focus on salient objects or regions with similar semantic information across two data-augmented views, which keeps the coherence of input images. Besides, SetSim effectively filters out misleading local features, which avoids establishing noisy spatial correspondence. From a qualitative perspective, the visualization confirms our motivation for exploring set similarity across two views to improve the robustness.

Table 6. **Comparisons with the state-of-the-art approaches on COCO object detection, instance segmentation, and keypoint detection.** All methods are fine-tuned on `train2017` with 1× schedules and evaluated on `val2017`. A Mask-RCNN (R50) [20, 41] with FPN [25] is adopted for all methods. Average precision on bounding-boxes (AP^b), masks (AP^m) and keypoint (AP^{kp}) are used as benchmark metrics. Following [18], we fine-tune DenseCL and PixPro with their official pretrained weights, because they adopted a different COCO fine-tuning setting from the common approach [6, 7, 18]. Each model is pretrained for 200 epochs. (Average over 5 trials)

Method	Object Det.			Instance Seg.			Keypoint Det.		
	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}	AP^{kp}	AP^{kp}_{50}	AP^{kp}_{75}
Random init.	31.0	49.5	33.2	28.5	46.8	30.4	63.0	85.1	68.4
IN-1K sup.	38.9	59.6	42.7	35.4	56.5	38.1	65.3	87.0	71.3
MoCo-v1 [18]	38.5	58.9	42.0	35.1	55.9	37.7	66.1	86.7	72.4
MoCo-v2 [7]	38.9	59.2	42.4	35.4	56.2	37.8	66.3	87.1	72.2
VADeR [32]	39.2	59.7	42.7	35.6	56.7	38.2	66.1	87.3	72.1
DenseCL [40]	39.4	59.9	42.7	35.6	56.7	38.2	66.6	87.4	**72.6**
PixPro [46]	39.8	59.5	43.7	36.1	56.5	38.9	66.5	87.6	72.3
ReSim-C4 [45]	39.3	59.7	43.1	35.7	56.7	38.1	66.3	87.2	72.4
SetSim	**40.2**	**60.7**	**43.9**	**36.4**	**57.7**	**39.0**	**66.7**	**87.8**	72.4

4.2. Main Results

PASCAL VOC Object Detection. We utilize the Faster R-CNN (R50-C4) detector and keep the same setting as mentioned in Section 4.1. As illustrated in Table 4, we report the object detection result on PASCAL VOC and compare it with a series of state-of-the-art methods. Our method yields significant improvements than the MoCo-v2 baseline [18] at AP^b, AP^b_{50}, and AP^b_{75}. Furthermore, our method surpasses DenseCL [40] and ReSim [45] by 0.4% at AP^b, respectively.

PASCAL VOC & Cityscapes & ADE20K Semantic Segmentation. Following the setting of [40][1], we fine-tune an FCN [9, 27] on VOC `train_aug2012` set (~10k images) for 20k iterations and evaluate on `val2012` set. Besides, We fine-tune on Cityscapes `train_fine` set (2975 images) for 40k iterations and test on `val` set. Finally, following the standard scheduler [9], we fine-tune an FCN on ADE20K [49] `train` set (~20k images) for 80k iterations and evaluate on `val` set (~2k images). As shown in Table 5, SetSim obtains remarkable gains of 3.0%, 3.4%, and 2.7% mIoU on Cityscapes, VOC, and ADE20K than the supervised ImageNet pre-training, respectively. Moreover, SetSim outperforms DenseCL [40] and ReSim [45] by significant margins on three benchmarks, which strongly demonstrate our method is friendly for dense prediction task.

COCO Object Detection & Instance Segmentation & Keypoint Detection. Following [18], we fine-tune a Mask-RCNN (R50) [20, 41] with FPN [25] (Keypoint-RCNN [41]) under 1× scheduler and add new Batch Normalization layers before the FPN parameters. All Batch Normalization statistics are synchronized across GPUs. Training is conducted on `train2017` split with ~118k images, and testing is conducted on `val2017` split. The short-side length of input images is randomly selected from 640 to 800 pixels during training and fixed at 800 pixels for testing. Following [7, 8, 18], Average Precision on bounding-boxes (AP^b) and Average Precision on masks (AP^m) are adopted as our metrics. Table 6 demonstrates that SetSim improve over the state-of-the-art PixPro [46] under both box and mask metrics, where the gains are 0.4%, 0.3%, and 0.2% at AP^b, AP^m, and AP^{kp}.

[1] https://github.com/WXinlong/mmsegmentation

Table 7. **Results on Cityscapes instance segmentation.** A Mask-RCNN (R50) [20, 41] with FPN [25] is fine-tuned on `train_fine` set and evaluated on `val` set. Following [18], we fine-tune all methods with their official pretrained weights because the results of SwAV, DenseCL, and PixPro are not public. Each model is pretrained for 200 epochs. (Average over 5 trials)

Method	AP^m	AP^m_{50}
Random init.	25.6	51.5
IN-1K sup.	32.9	59.6
MoCo-v1 [18]	32.8	59.2
MoCo-v2 [7]	33.4	60.4
SwAV [5]	33.6	62.5
DenseCL [40]	34.9	62.5
PixPro [46]	34.0	62.0
ReSim-C4 [45]	35.4	63.1
SetSim	**35.6**	**63.4**

Figure 4. **Visualization of corresponding sets across two data-augmented views.** Each corresponding set is constructed by 200-epoch pretrained model. Red circle in Set 2 denotes the Set 1's nearest neighbours in the View 2.

Cityscapes Instance Segmentation. Following the setting of ReSim [45], we fine-tune a Mask-RCNN (R50) [20, 41] with FPN [25] and add Batch Normalization layers before the FPN, and synchronize all Batch Normalization during training. Table 7 illustrates the comparisons of SetSim versus the supervised pre-training counterparts and a series of state-of-the-art methods. SetSim remarkably improves over the baseline MoCo-v2 by 2.2% and 3.0% at AP^m and AP^m_{50} and surpass the state-of-the-art work, ReSim [45]. We present lots of qualitative analysis in the supplementary.

5. Conclusion

In this paper, we have proposed a simple but effective dense self-supervised representation learning framework, SetSim, by exploring set similarity across views for dense prediction tasks. By resorting to attentional features, SetSim constructs the corresponding set across two views, which alleviates the effect of misleading pixels and semantic inconsistency. Besides, since some useful features are neglected from sets, we further search the cross-view nearest neighbours of sets to enhance the structure neighbour information. Finally, a contrastive/similarity loss function is utilized to map two sets of pixel-wise feature vectors to similar representations in the embedding space, encouraging the model to learn adequate dense visual representations. Compared to the MoCo-v2 and SimSiam baseline, our SetSim significantly improves the localization and classification abilities on a series of dense prediction tasks, which narrow the transfer gap between the self-supervised pretraining and dense prediction tasks. Empirical evaluations have demonstrated that SetSim is comparable with or outperforms state-of-the-art approaches on various downstream tasks, including PASCAL VOC object detection, COCO object detection, COCO instance segmentation, COCO keypoint detection, Cityscapes instance segmentation, PASCAL VOC semantic segmentation, Cityscapes semantic segmentation, and ADE20K semantic segmentation. Besides, experimental results show that different types of spatial correspondence tend to be complementary. We argue that there should be a sweet point of various spatial correspondence combinations, and we are highle motivated to study that in the future work.

Acknowledgements We thank Wanjing Zong, Xunqiang Tao and Ziyu Chen for the helpful discussions on this work. Meanwhile, We appreciate Weiqiong Chen, Bin Long and Rui Sun for AWS technical support.

References

[1] Pulkit Agrawal, Joao Carreira, and Jitendra Malik. Learning to see by moving. In *Proceedings of the IEEE international conference on computer vision*, pages 37–45, 2015. 2

[2] Yuki Markus Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. *arXiv preprint arXiv:1911.05371*, 2019. 2

[3] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)*, pages 132–149, 2018. 1, 2

[4] Mathilde Caron, Piotr Bojanowski, Julien Mairal, and Armand Joulin. Unsupervised pre-training of image features on non-curated data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2959–2968, 2019. 2

[5] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, 33:9912–9924, 2020. 1, 2, 6, 7

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 2, 3, 4, 6, 7

[7] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 2, 3, 4, 5, 6, 7

[8] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021. 1, 2, 4, 5, 6, 7

[9] MMSegmentation Contributors. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. https://github.com/open-mmlab/mmsegmentation, 2020. 6, 7

[10] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 4

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 1, 4

[12] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016. 2

[13] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010. 4, 5

[14] Jun Fu, Jing Liu, Jie Jiang, Yong Li, Yongjun Bao, and Hanqing Lu. Scene segmentation with dual relation-aware attention network. *IEEE Transactions on Neural Networks and Learning Systems*, 32(6):2547–2560, 2020. 1

[15] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3146–3154, 2019. 1

[16] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018. 2

[17] 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. *Advances in Neural Information Processing Systems*, 33:21271–21284, 2020. 1, 2, 6

[18] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020. 2, 3, 4, 5, 6, 7

[19] Kaiming He, Ross Girshick, and Piotr Dollár. Rethinking imagenet pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4918–4927, 2019. 2

[20] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 7, 8

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1, 3

[22] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. 6

[23] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Colorization as a proxy task for visual understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6874–6883, 2017. 2

[24] Shikun Li, Xiaobo Xia, Shiming Ge, and Tongliang Liu. Selective-supervised contrastive learning with noisy labels. *arXiv preprint arXiv:2203.04181*, 2022. 2, 3

[25] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 7, 8

[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 *European conference on computer vision*, pages 740–755. Springer, 2014. 4

[27] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 6, 7

[28] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 4

[29] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *European Conference on Computer Vision*, pages 527–544. Springer, 2016. 2

[30] T Nathan Mundhenk, Daniel Ho, and Barry Y Chen. Improvements to context based self-supervised learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9339–9348, 2018. 2

[31] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pages 69–84. Springer, 2016. 2

[32] Pedro O O Pinheiro, Amjad Almahairi, Ryan Benmalek, Florian Golemo, and Aaron C Courville. Unsupervised learning of dense visual representations. *Advances in Neural Information Processing Systems*, 33:4489–4500, 2020. 1, 2, 7

[33] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016. 2

[34] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015. 1, 5, 6

[35] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5693–5703, 2019. 2

[36] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10781–10790, 2020. 2

[37] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *European conference on computer vision*, pages 776–794. Springer, 2020. 4

[38] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? *Advances in Neural Information Processing Systems*, 33:6827–6839, 2020. 6

[39] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008. 2

[40] Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li. Dense contrastive learning for self-supervised visual pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3024–3033, 2021. 1, 2, 4, 6, 7

[41] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 5, 6, 7, 8

[42] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3733–3742, 2018. 3

[43] Xiaobo Xia, Tongliang Liu, Bo Han, Nannan Wang, Mingming Gong, Haifeng Liu, Gang Niu, Dacheng Tao, and Masashi Sugiyama. Part-dependent label noise: Towards instance-dependent label noise. *Advances in Neural Information Processing Systems*, 33:7597–7610, 2020. 3

[44] Xiaobo Xia, Tongliang Liu, Nannan Wang, Bo Han, Chen Gong, Gang Niu, and Masashi Sugiyama. Are anchor points really indispensable in label-noise learning? *Advances in Neural Information Processing Systems*, 32, 2019. 3

[45] Tete Xiao, Colorado J Reed, Xiaolong Wang, Kurt Keutzer, and Trevor Darrell. Region similarity representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10539–10548, 2021. 6, 7, 8

[46] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16684–16693, 2021. 1, 2, 6, 7

[47] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer, 2016. 2

[48] Richard Zhang, Phillip Isola, and Alexei A Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1058–1067, 2017. 2

[49] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017. 4, 7