Patch-level Contrastive Learning via Positional Query for Visual Pretraining

Shaofeng Zhang [1] **Qiang Zhou** [2] **Zhibin Wang** [2] **Fan Wang** [2] **Junchi Yan** [1]

Abstract

Dense contrastive learning (DCL) has been recently explored for learning localized information for dense prediction tasks (*e.g.*, detection and segmentation). It still suffers the difficulty of mining pixels/patches correspondence between two views. A simple way is inputting the same view twice and aligning the pixel/patch representation. However, it would reduce the variance of inputs, and hurts the performance. We propose a plug-in method PQCL (Positional Query for patch-level Contrastive Learning), which allows performing patch-level contrasts between two views with exact patch correspondence. Besides, by using positional queries, PQCL increases the variance of inputs, to enhance training. We apply PQCL to popular transformer-based CL frameworks (DINO and iBOT, and evaluate them on classification, detection and segmentation tasks, where our method obtains stable improvements, especially for dense tasks. It achieves new state-of-the-art in most settings. Code is available at https://github.com/Sherrylone/Query_Contrastive.

1. Introduction

Self-supervised learning (SSL) has achieved promising results across variant tasks due to it's strong transferability. Besides, unlike supervised learning, SSL does not rely on heavily labeled data in pre-training stage, which reduces the cost of data annotations. Existing SSL methods mainly fall into three categories: **1) Generative** approaches (Goodfellow et al., 2014) learn to estimate the distribution of input data. However, generation can be computationally expensive and pixel-wise information may not be necessary for representation learning. **2) Contextual** methods (Zhang et al., 2016; Gidaris et al., 2018) design pretext tasks (denoising auto-encoders (Vincent et al., 2008), context auto encoders (Zhang et al., 2016), etc). **3) Contrastive** methods (Chen et al., 2020a;b) take augmented views of the same image as positive pairs and others as negative pairs. Contrastive-based methods have shown great promise *e.g.*, in image classification/detection, video classification (Caron et al., 2021), multi-modal learning (Jin et al., 2022; 2023a) and others (Jin et al., 2023b).

Existing contrastive learning in general (Chen et al., 2020a; Caron et al., 2021) aims to learn global-discriminative features, which may lack spatial sensitivity (Yun et al., 2022), limiting their ability on dense vision tasks like detection and segmentation. Consequently, pixel-wise (Xie et al., 2021c; Wang et al., 2021) and patch-wise (Yun et al., 2022) contrastive objectives and frameworks are proposed. However, one main shortcoming of these DCL methods is establishing the correspondence among pixels/patches usually requires bilinear interpolation, which is complex and heavily sensitive to random crop augmentation (in an extreme case, if two views have no intersection parts, there are no correspondence relation). To overcome this issue, patch-level masked augmentation is proposed within the same view is proposed in iBOT (Zhou et al., 2022). However, the variance of the inputs (masked and unmasked views) is much lower than inputting two different views, where variance has been proven to be the key to success in contrastive learning (Wang et al., 2022a). To address this issue, inspired by query crop and cross attention mechanism proposed in (Zhang et al., 2023), we propose positional-query-based patch-level contrasting, which only inputs relative positional embedding (without pixel information) to student branch, and feeds the query crop (with pixel information) to teacher branch to guide the student. Such that, PQCL allows to perform patch-level contrasts with larger variance, and learns more spatial-sensitive information. **The main contributions are:**

1) We propose PQCL, a positional-query-based patch-level contrastive learning method, which inputs relative positional embedding (without pixel information) to the student branch, and feeds the query crop (with pixel information) to the teacher branch to guide the student. PQCL could further increase variance between inputs, resulting in better performance in downstream tasks. Besides, PQCL can serve as a plug-in tool, and can be easily integrated into recent advanced transformer-based contrastive learning architecture (*e.g.*, DINO (Caron et al., 2021), iBOT (Zhou et al., 2022)).

[1]Department of CSE, and MoE Key Lab of Artificial Intelligence, Shanghai Jiao Tong University [2]Alibaba Group. Correspondence to: Junchi Yan <yanjunchi@sjtu.edu.cn>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

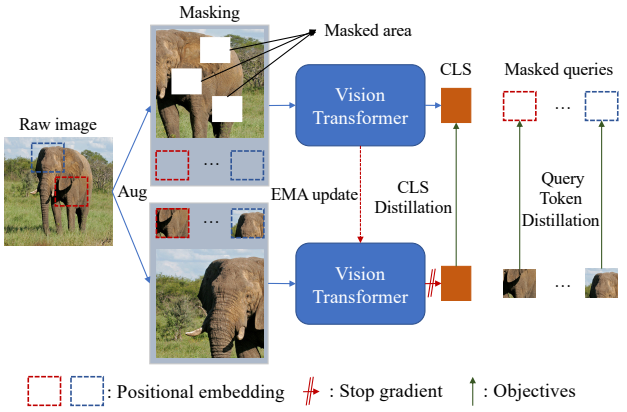

Figure 1. Framework of PQCL. Images are first augmented twice to generate two views. Then, we randomly crop several query views and calculate the relative position between query crops and the two augmented views. Then, for student branch (top), we randomly masked some patches of the augmented views and jointly input the masked views and relative positional embedding of query views (**without pixel information**). For teacher branch (bottom), we individually input the augmented views (without masking) and query views (**with pixel information**) to guide the student branch. Then, objectives are added on the *[CLS]* token and query patches.

2) Instead of directly using the self attention (Vaswani et al., 2017) (would hurt the downstream performance) or regressor module (extra parameters) recently proposed in the previous method (CAE (Chen et al., 2022), SIM (Tao et al., 2022)), we propose cross attention in each block between positional and patches embeddings to learn semantic information of positional queries, and the cross attention mechanism does not incur extra parameters on the basis of iBOT.

3) We conduct comprehensive experiments on standard visual benchmarks, including linear probing, finetuning on classification, detection and segmentation, where the proposed PQCL can stably improve the baseline DINO and iBOT a lot. Specifically, For ViT/B, PQCL outperforms iBOT 0.9% top-1 accuracy by linear probing on ImageNet. For ViT/S, PQCL outperforms baseline iBOT by 2.4% mAP^{bb} and 1.5% mAP^{mk} on detection and segmentation on MS-COCO dataset, respectively. For semantic segmentation, PQCL outperforms iBOT 0.8% mIoU on ADE20K.

2. Related Work

Dense contrastive SSL. A prominent line of SSL, often referred to as "contrastive" or "siamese" approaches, trains networks by matching the representation of different views obtained from the same image by means of data augmentation (Chen et al., 2020a; He et al., 2020; Caron et al., 2021; Wang et al., 2021). These approaches have primarily been developed with global (image-level) objectives but several recent works have adapted them to learn local features (Wang et al., 2021; Yang et al., 2021; Ziegler & Asano, 2022; Ge et al., 2021) by different pixel-/patch- wise correspondence mining techniques. DenseCL (Wang et al., 2021) exploits the correspondence by sorting the similarities of pixels in the deep feature map, while PixPro (Xie et al., 2021c) utilizes the augmentation wrapper to get the spatial correspondence of the pixel intersection between two views. Furthermore, Detco (Xie et al., 2021a) tries to improve the performance of general contrastive learning approaches by augmenting multiple global and local views simultaneously. Inspired by PixPro, Resim (Xiao et al., 2021) uses RoI Pooling (Jiang et al., 2018) to extract a feature vector from the associated feature map region for both views. On the basis of DenseCL, SetSim (Wang et al., 2022b) employs a threshold selection to filter out noisy backgrounds. With the development of ViT in SSL (Dosovitskiy et al., 2020), SelfPatch (Yun et al., 2022) treats the spatial neighbors of the patch as positive examples for learning semantically meaningful relations among patches. On the basis of SelfPatch, ADCLR (Zhang et al., 2023) proposes patch-level contrasting via query crop and cross attention mechanism.

Masked Image Modeling (MIM). These methods learn vision representation by reconstructing the masked patches from the partial observations. Based on the reconstruction objective, they can be divided into: pixel-wise reconstruction (He et al., 2021) and auxiliary feature/tokens prediction (Dong et al., 2021; Zhou et al., 2022). SimMIM (Xie et al., 2021d) and MAE (He et al., 2021) are the first two methods applying mask modeling in the visual domain. They propose to reconstruct the raw pixel values from either the full set of image patches (mask tokens and visible patches for SimMIM) or partially observed patches (visible patches for MAE). Compared with SimMIM, MAE is more efficient by dropping out a large portion of input patches. To learn richer semantic features, MaskFeat (Wei et al., 2021) introduces the HOG features (Dalal & Triggs, 2005) as supervision and l2 loss is added to each pixel on HOG feature and predicted features. Inspired by adversarial training (Goodfellow et al., 2014), CIM (Fang et al., 2022) adds perturbations to raw images to enhance robustness for reconstruction. Another line of MIM is predicting token prediction in teacher-student architecture. iBOT (Zhou et al., 2022) is the first to perform MIM objective by inputting the same view to the teacher (original) and student (masked) branches. On the basis of iBOT, SIM (Tao et al., 2022) adds a transformer decoder module to predict masked patches information of the other view.

3. Methodology

3.1. Preliminaries

Vision Transformers. Denote an image by $\mathbf{x} \in \mathbb{R}^{C \times H \times W}$, where $H \times W$ is the resolution of the image and C is

the number of channels. Plain ViT (Dosovitskiy et al., 2020) treats the image $\mathbf{x}$ as a sequence composed of non-overlapping patches $\{\mathbf{x}^{(i)} \in \mathbb{R}^{CP^2}\}_{i=1}^{N}$, where each patch has a fixed $P \times P$ resolution. Then, the patches are linearly transformed to D-dimensional patch embeddings $\mathbf{z}^{(i)} = \mathbf{E}\mathbf{x}^{(i)} + \mathbf{W}_{pos}^{i} \in \mathbb{R}^{D}$, where $\mathbf{E} \in \mathbb{R}^{D \times CP^2}$ is the linear projection and $\mathbf{W}_{pos} \in \mathbb{R}^{D}$ is the positional embedding for the i-th patch. A $[CLS]$ token $\mathbf{z}^{[CLS]} \in \mathbb{R}^{D}$ is subsequently prepended to the patch sequence to extract global information, so the resulting input sequence is represented as $\mathbf{z} = [\mathbf{z}^{[CLS]}, \mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \cdots, \mathbf{z}^{(N)}]$. Then, ViT uses a Transformer encoder (Vaswani et al., 2017) to generate both image-level ($[CLS]$ token) and patch-level (other tokens). In line with SelfPatch (Yun et al., 2022), we use f_θ to denote the whole process of a ViT parameterized by θ:

$$
\begin{aligned}
f_\theta(\mathbf{x}) &= f_\theta\left(\left[\mathbf{z}^{[CLS]}, \mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \cdots, \mathbf{z}^{(N)}\right]\right) \\
&= \left[f_\theta^{[CLS]}(\mathbf{x}), f_\theta^{(1)}(\mathbf{x}), f_\theta^{(2)}(\mathbf{x}), \cdots, f_\theta^{(N)}(\mathbf{x})\right],
\end{aligned} \tag{1}
$$

where $f_\theta^{[CLS]}(x)$ and $f_\theta^{(i)}(\mathbf{x})$ are the representations of the whole image and i-th patch, respectively.

Self-supervised learning with ViTs. Since our PQCL is mainly built on top of iBOT (Zhou et al., 2022), we shortly review the framework and objective of iBOT. Given the image $\mathbf{x}$, iBOT constructs a positive pair $(\mathbf{X}_A, \mathbf{X}_B)$ through random augmentation. Then, iBOT randomly masks some patches to generate the masked versions $\mathbf{X}_{Am}$ and $\mathbf{X}_{Bm}$. The overall objectives are added on $[CLS]$ tokens across two views and masked tokens within the same view (masked and original versions), which is formulated as:

$$
\begin{aligned}
\mathcal{L}_{iBOT} = &\mathcal{H}\left(g_\gamma\left(f_\theta^{[CLS]}(\mathbf{x}_A)\right), sg\left(g_{\gamma'}(f_{\theta'}^{[CLS]}(\mathbf{x}_B))\right)\right) \\
&+ \mathcal{H}\left(g_\gamma\left(f_\theta^{[CLS]}(\mathbf{x}_B)\right), sg\left(g_{\gamma'}(f_{\theta'}^{[CLS]}(\mathbf{x}_A))\right)\right) \\
&+ \lambda \cdot \mathcal{H}\left(g_\gamma\left(f_\theta^{[mk]}(\mathbf{x}_{Am})\right), sg\left(g_{\gamma'}(f_{\theta'}^{[mk]}(\mathbf{x}_A))\right)\right) \\
&+ \lambda \cdot \mathcal{H}\left(g_\gamma\left(f_\theta^{[mk]}(\mathbf{x}_{Bm})\right), sg\left(g_{\gamma'}(f_{\theta'}^{[mk]}(\mathbf{x}_B))\right)\right)
\end{aligned} \tag{2}
$$

where $\mathcal{H}(a, b) = -a \log b$ is the cross entropy loss. $sg(\cdot)$ means stop-gradient operation. g_γ is the MLP projector, which is commonly used in previous SSL methods (Chen et al., 2020a; Grill et al., 2020). γ' and θ' denote the exponential moving averages updated parameters in the teacher branch. $f_\theta^{[mk]}$ means masked patches. The first two terms in Eq. 2 are the global loss added on $[CLS]$ token across the two different views, and the last two terms are the patch-level loss on the masked patches. We can glance the patch-level loss is added on the same view (*i.e.*, $\mathbf{X}_{A/B}$ and $\mathbf{X}_{Am/Bm}$), which limits the variance of two branches inputs, and could influence the downstream performance.

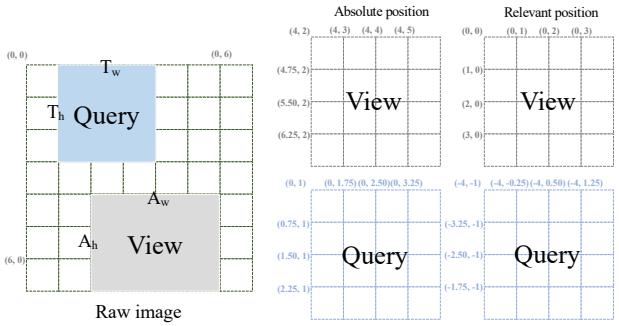

Figure 2. Illustration of relative positional encoding: absolute global positional encoding and relative positional encoding. We first randomly crop global views and query views, where query view is usually much smaller than global views (for better illustration, we increase the region of query crop). Then, we use the positional encoding of the global view to represent the relative positional encoding of the query view.

3.2. Contrastive Learning via Positional Query

Query crops. To increase the variance of patch-level inputs, we randomly crop each image to generate Q query crops, where Q is a pre-defined hyper-parameter. Then, we resize each query crop to the low resolutions (*e.g.*, 32 $\times$ 32, 96 $\times$ 96) and divide them into several query patches with the exact resolution of raw patches (e.g., 16 $\times$ 16). Denote the i-th query patches as $\mathbf{x}^{q_i}$. Then, we add the relative positional embedding $\mathbf{w}^q$ on each query patch and feed the additions into a linear projector to get its embedding $\mathbf{z}^{(q_i)}$. Then, the embedding sequence can be formulated as:

$$
\mathbf{z} = \left[\mathbf{z}^{[CLS]}, \underbrace{\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \cdots, \mathbf{z}^{(N)}}_{\text{raw patches}}, \underbrace{\mathbf{z}^{(q_1)}, \mathbf{z}^{(q_2)}, \cdots, \mathbf{z}^{(q_Q)}}_{\text{query patches}}\right]. \tag{3}
$$

Note that both **raw patches** and **query patches** will add the positional embedding.

Relative positional encoding. Instead of simply using learnable positional encoding in baseline iBOT (Zhou et al., 2022), which is difficult to represent the relation between global views and query view, we use fixed positional encoding of global views to represent the position of query views, which is illustrated in Fig. 2. Specifically, for each position, we use the following popular form in transformers (Vaswani et al., 2017) to generate positional embedding:

$$
\begin{aligned}
\mathbf{W}_{h,w} = \Big[&\sin\left(\frac{w}{e^{2*1/d}}\right), \cos\left(\frac{w}{e^{2*2/d}}\right), \cdots, \sin\left(\frac{w}{e}\right), \\
&\sin\left(\frac{h}{e^{2*1/d}}\right), \cos\left(\frac{h}{e^{2*2/d}}\right), \cdots, \sin\left(\frac{h}{e}\right)\Big]
\end{aligned} \tag{4}
$$

where $e = 10000$ is the pre-defined parameter, which is also commonly used in MAE (He et al., 2021). (w, h) means the position of top left patch position (illustrated in Fig. 2).

Extracting semantic information from query views. For the student branch, we input raw patches and positional embeddings of query views. In other words, query views are completely masked. To extract the semantic information of query views, we design **cross attention** between positional embedding and raw views. Specifically, the input of the student self-attention block can be written as:

$$\mathbf{Z} = \Big[\mathbf{z}^{[CLS]}, \underbrace{\mathbf{z}^{(1)}, \mathbf{m}^{(2)}, \cdots, \mathbf{z}^{(N)}}_{\text{raw patches}}, \underbrace{\mathbf{m}^{(q_1)}, \mathbf{m}^{(q_2)}, \cdots, \mathbf{m}^{(q_Q)}}_{\text{query positions}}\Big] \tag{5}$$

where $\mathbf{m}^{(\cdot)} = \mathbf{m} + \mathbf{w}^{(\cdot)}$ means the placeholder of masked patches in position $(\cdot)$, and $\mathbf{m}$, $\mathbf{w}$ are the learnable vector to represent the general masked information and positional embedding, respectively. Q is the number of patches in the query view. For each attention block, we formulate the cross-attention mechanism as:

$$\mathrm{Attn}(\mathbf{Q}_{\mathbf{z}^{(q_i)}}, \mathbf{K}_{\mathbf{z}'}, \mathbf{V}_{\mathbf{z}'}) = \mathrm{Softmax}\left(\frac{\mathbf{Q}_{\mathbf{z}^{(q_i)}}\mathbf{K}_{\mathbf{z}'}{}^{\top}}{\sqrt{d_k}}\right)\mathbf{V}_{\mathbf{z}'}, \tag{6}$$

where $1 \le i \le Q$. Eq. 6 formulates how to extract semantic information of completely masked query views. For the two global views in each attention block, we perform self attention, which is the same with vallina ViTs (Dosovitskiy et al., 2020). For teacher backbone, we feed the two global views and query view (with pixel information) one by one:

$$\begin{aligned}\mathbf{Z}^{raw} &= \left[\mathbf{z}^{[CLS]}, \mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \cdots, \mathbf{z}^{(N)}\right] \\ \mathbf{Z}^{query} &= \left[\mathbf{z}^{[query]}, \mathbf{z}^{(q_1)}, \mathbf{z}^{(q_2)}, \cdots, \mathbf{z}^{(q_Q)}\right].\end{aligned} \tag{7}$$

Objective functions. The objective function of PQCL includes three parts: **i)** global objective function across two global views to learn global-discriminative information, **ii)** patch-level contrastive objective within the same view to learn local information and **iii)** patch-level contrastive objective of completely masked query views. The fist two objectives are written in Eq. 2, and the proposed objective (last one) can be formulated as:

$$\begin{aligned}\mathcal{L}_{query} = &\mathcal{H}\left(g_\gamma\left(f_\theta(\mathbf{Z}_A, \mathbf{M}^{query})\right), sg\left(g_{\gamma'}(f_{\theta'}(\mathbf{Z}^{query}))\right)\right) \\ &+ \mathcal{H}\left(g_\gamma\left(f_\theta(\mathbf{Z}_B, \mathbf{M}^{query})\right), sg\left(g_{\gamma'}(f_{\theta'}(\mathbf{Z}^{query}))\right)\right)\end{aligned} \tag{8}$$

where $\mathbf{M}_{query} = [\mathbf{m}^{(q_1)}, \mathbf{m}^{(q_2)}, \cdots, \mathbf{m}^{(q_Q)}]$ is the sequence of tokenized query view. The backbone f receives two types of inputs, $f(\cdot, \cdot)$ means using cross attention to represent the completely masked query views in student branch and $f(\cdot)$ means only using self attention to extract the semantic information of query views. The overall objective function of PQCL is:

$$\mathcal{L}_{PQCL} = \begin{cases} \mathcal{L}_{iBOT} + \lambda \cdot \mathcal{L}_{query}, & \textit{for iBOT baseline} \\ \mathcal{L}_{DINO} + \lambda \cdot \mathcal{L}_{query}, & \textit{for DINO baseline} \end{cases} \tag{9}$$

where λ is the hyper-parameter to balance the query loss and loss of baselines. We use iBOT as the baseline by default.

4. Experiments

4.1. Experiment Setup

Platform. The experiments are performed on a work station with 16 V100 GPUs by default (if not otherwise specified).

Datasets. We conduct self-supervised pre-training on the ImageNet-1K (Deng et al., 2009) training set with 1,000 classes, as used in SSL for both MIM (He et al., 2021) and contrastive learning (Chen et al., 2020a). We also transfer the encoder pre-trained by PQCL on MS-COCO (Lin et al., 2014), ADE20K (Zhou et al., 2017), and video segmentation dataset DAVIS 2017 (Pont-Tuset et al., 2017).

Baselines. We consider recent advanced self-supervised methods based on the ResNets (He et al., 2016) and ViTs (Dosovitskiy et al., 2020) architectures: (a) self-supervised ResNets: SimCLR (Chen et al., 2020a), MoCo-v2 (Chen et al., 2020b), SwAV (Caron et al., 2020), Barlow Twins (Zbontar et al., 2021), ZeroCL (Zhang et al., 2021), ARB (Zhang et al., 2022), DenseCL (Wang et al., 2021), ReSim (Xiao et al., 2021), and DetCo (Xie et al., 2021a); and (b) self-supervised ViTs: DINO (Caron et al., 2021), MoCo-v3 (Chen et al., 2021), MoBY (Xie et al., 2021b), iBOT (Zhou et al., 2022) and SelfPatch (Yun et al., 2022).

Pre-training hyper-parameters. In line with iBOT, we train with Adamw (Loshchilov & Hutter, 2018) and a batch size of 1024, distributed over 16 GPUs using ViT-S/16 (batch size per GPU is 64). The learning rate is linearly ramped up during the first 30 epochs to its base value determined with the following linear scaling rule (Chen et al., 2020a): lr = 0.0005, batchsize=256. After warmup, we decay the learning rate with a cosine schedule (Loshchilov & Hutter, 2016). The weight decay also follows a cosine scheduled from 0.04 to 0.4. The temperature τ is set to 0.04 while we use a linear warm-up for τ_t from 0.04 to 0.07 during the first 30 epochs (Although the larger temperature could result in better performance, it will be also **unsafe** for small architectures, *e.g.*, ViT-S.). We follow the data augmentations of BYOL (Grill et al., 2020) (color jittering, Gaussian blur, and solarization) and multi-crop (Caron et al., 2020) with a bicubic interpolation to adapt the position embeddings to the scales. For both two baselines DINO and iBOT, the query crop ratio is randomly sampled from 0.05$\sim$0.25 and we only use two global views without local views due to the limitation of GPU storage. For baseline iBOT, we set the masked ratio of global views as 0.3.

Evaluation protocols. We use standard self-supervised learning protocols, including learning a linear classifier on frozen features (Chen et al., 2020a; He et al., 2020) and

Table 1. Linear probing on ImageNet-1K. All the methods only use two global views without the multi-crop strategy. * means our reproduction (with main difference as the positional embedding).

Method	Arch	Epochs	Top-1	Top-5
SimCLR (Chen et al., 2020a)	ResNet-50	1000	70.0	89.0
Moco V2 (Chen et al., 2020b)	ResNet-50	100	60.6	∼
Moco V2 (Chen et al., 2020b)	ResNet-50	800	71.1	∼
SwAV (Caron et al., 2020)	ResNet-50	400	70.1	∼
Zero-CL (Zhang et al., 2021)	ResNet-50	400	72.6	90.5
Barlow Twins (Zbontar et al., 2021)	ResNet-50	1000	73.2	91.0
ARB (Zhang et al., 2022)	ResNet-50	1000	73.0	91.5
BYOL (Grill et al., 2020)	ResNet-50	1000	74.3	∼
Moco V3 (Chen et al., 2021)	ViT/S	100	68.9	∼
Moco V3 (Chen et al., 2021)	ViT/S	300	72.8	∼
DINO (Caron et al., 2021)	ViT/S	100	67.8	∼
DINO (Caron et al., 2021)	ViT/S	300	72.5	∼
iBOT (Zhou et al., 2022)	ViT/S	100	68.8	88.7
iBOT (Zhou et al., 2022)	ViT/S	400	73.5	91.3
iBOT (Zhou et al., 2022)	ViT/S	800	74.0	91.6
PQCL (Ours)	ViT/S	100	69.7	89.1
PQCL (Ours)	ViT/S	400	73.8	91.4
PQCL (Ours)	ViT/S	800	**74.4**	**91.9**
iBOT (Zhou et al., 2022)	ViT/B	400	76.0	92.6
iBOT* (Zhou et al., 2022)	ViT/B	400	75.8	92.5
PQCL (Ours)	ViT/B	400	**76.9**	**93.0**

finetune on downstream tasks (He et al., 2021; Chen et al., 2022). For linear evaluations, we apply random resize crops and horizontal flips augmentation for training, and report accuracy on a central crop. For finetuning evaluations (detection and segmentation on MS-COCO (Lin et al., 2014), segmentation on ADE20K (Zhou et al., 2017)), we initialize networks with the pre-trained weights to adapt with further training. In line with (Zbontar et al., 2021), we also evaluate our method's transfer ability on small-scale and fine-grained classification dataset (Van Horn et al., 2018).

4.2. Main Results

We choose ViT/S and ViT/B as backbones, and report the linear probing results with different pretraining epochs in Table 1. For 100 and 800 epochs pretraining with ViT/S, PQCL outperforms iBOT by 0.6% and 0.4%, respectively. For ViT/B, PQCL outperforms iBOT by 1.1% top-1 accuracy with 400 epochs pretraining.

4.3. Transfer Learning Tests

COCO object detection and segmentation. Setups. We evaluate pre-trained models on the COCO object detection and instance segmentation tasks (Lin et al., 2014). We test our model under two popular frameworks Mask R-CNN (He et al., 2017) and Cascade R-CNN (Cai & Vasconcelos, 2018) with the standard 1x schedule. **Results.** Table 2 shows the proposed PQCL can consistently outperform iBOT in both detection and segmentation tasks. We evaluate PQCL with both 200 and 300 epochs pretraining. For 300 epochs pretraining without local views, PQCL surpasses DINO with 800 epochs and 10 local views pretraining 0.9% point mAP^{bb} and 0.5% point mAP^{mk}, respectively. For 200 epochs pretraining, PQCL outperforms DINO and iBOT with 300 epochs pretraining without local views. With 300 epochs pretraining, PQCL outperforms iBOT by 2.3% point mAP^{bb} and 1.5% mAP^{mk}, respectively.

ADE20K semantic segmentation. Setup. We evaluate semantic segmentation performances of pre-trained models on ADE20k (Zhou et al., 2017), which contains 150 fine-grained semantic categories and 25k training data. We finetune the pretrained models on Semantic FPN (Lin et al., 2017) and UperNet (Xiao et al., 2018) with 40k and 160k iteration, respectively. Following SelfPatch (Yun et al., 2022), we report three metrics: (a) mean intersection of union (mIoU) averaged over all semantic categories, (b) all pixel accuracy (aAcc), and (c) mean class accuracy (mAcc). **Results.** As shown in Table 3, PQCL can outperform previous all methods under the same setting. Besides, we further evaluate DINO with 10 local views and 800 epochs pretraining (checkpoint is downloaded in their official repository [1]), where PQCL gets 0.8 point improvements with only 200 epochs pretraining and only two global views. We guess the big improvements are because PQCL is a patch-level contrastive learning method, which is more sensitive to spatial information, resulting the higher performance in dense prediction tasks. Besides, compared with the baseline iBOT, our PQCL also gets 1.1 higher points, since PQCL increases the difficulty of patch-level objectives on the basis of iBOT.

DAVIS 2017 (Pont-Tuset et al., 2017) video segmentation. Setup. We perform video object segmentation using pretrained models on DAVIS 2017. We follow the evaluation protocol in DINO (Caron et al., 2021) and SelfPatch (Yun et al., 2022), which does not require extra training costs. It evaluates the quality of frozen representations of image patches by segmenting scenes with the nearest neighbor between consecutive frames. Followed by SelfPatch, we report three evaluation metrics: (a) mean region similarity $\mathcal{J}_m$, (b) mean contour-based accuracy $\mathcal{F}_m$, and (c) their average score $J\&\mathcal{F}_m$. **Results.** In Table 5, PQCL can stably improve the two baselines (iBOT and DINO), which is explained in Eq. 9. Specifically, for DINO, PQCL improves the $J\&\mathcal{F}_m$ score from 60.7 to 63.7, and for iBOT, PQCL improves the $J\&\mathcal{F}_m$ from 61.3 to 63.8. We see that the gain of PQCL on DINO is larger than PQCL on iBOT, which is probably because DINO only has the global objective on $[CLS]$ token, while iBOT has a masked patch-wise objective to learn spatial-sensitive information.

4.4. Ablation Study

Ablation on patch size. Since PQCL mainly benefits from the query positional embedding, we explore the influence of the number of query patches. Specifically, we fix the

[1] https://github.com/facebookresearch/dino

Table 2. Accuracy on MS-COCO. Mask R-CNN and Cascade R-CNN are adopted and trained with the 1x schedule. All the results are obtained by using our same finetune protocol for fair comparison, for the two tasks respectively. Epoch refers to the number of pretraining.

Method	Backbone	Framwork	#Epochs	#Param.	#Views.	Object Detection			Instance Segmentation		
						AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{mk}	AP^{mk}_{50}	AP^{mk}_{75}
Moco-V2 (Chen et al., 2020b)	ResNet-50		200	23M	2×224^2	38.9	59.2	42.4	35.5	56.2	37.8
SwAV (Caron et al., 2020)	ResNet-50		200	23M	2×224^2	38.5	60.4	41.4	35.4	57.0	37.7
DenseCL (Wang et al., 2021)	ResNet-50		200	23M	2×224^2	40.3	59.9	44.3	36.4	57.0	39.2
ReSim (Xiao et al., 2021)	ResNet-50		200	23M	2×224^2	40.3	60.6	44.2	36.4	57.5	38.9
DetCo (Xie et al., 2021a)	ResNet-50		200	23M	2×224^2	40.1	61.0	43.9	36.4	58.0	38.9
Moco V3 (Chen et al., 2021)	ViT-S/16	Mask RCNN	300	23M	2×224^2	39.8	62.6	43.1	37.1	59.6	39.2
MoBY (Xie et al., 2021b)	ViT-S/16		300	22M	2×224^2	41.1	63.7	44.8	37.3	60.3	39.8
DINO (Caron et al., 2021)	ViT-S/16		300	22M	2×224^2	40.8	63.4	44.2	37.3	59.9	39.5
SelfPatch (Yun et al., 2022)	ViT-S/16		200	22M	2×224^2	42.1	64.9	46.1	38.5	61.3	40.8
iBOT (Zhou et al., 2022)	ViT-S/16		200	22M	2×224^2	42.6	65.7	47.0	39.0	61.7	41.3
PQCL (Ours)	ViT-S/16		200	22M	2×224^2	**43.1**	**66.0**	**47.4**	**39.3**	**62.2**	**41.6**
DINO (Caron et al., 2021)	ViT-S/16		300	22M	2×224^2	45.2	64.9	47.8	38.9	61.2	41.7
DINO (Caron et al., 2021)	ViT-S/16		800	22M	$2 \times 224^2 + 10 \times 96^2$	46.8	66.7	50.3	40.6	63.7	43.2
iBOT (Zhou et al., 2022)	ViT-S/16	Cascade RCNN	300	22M	2×224^2	45.4	65.1	49.0	39.6	62.1	41.7
PQCL (Ours)	ViT-S/16		200	22M	2×224^2	46.2	65.5	49.8	39.9	62.3	42.6
PQCL (Ours)	ViT-S/16		300	22M	2×224^2	**47.7**	**67.0**	**51.3**	**41.1**	**64.0**	**44.2**

Table 3. **ADE20K semantic segmentation** performances of the recent self-supervised approaches pre-trained on ImageNet. The metrics mIoU, aAcc, and mAcc denote the mean intersection of union, all pixel accuracy, and mean class accuracy, respectively.

Method	Arch	Backbone	#Iter	#Epochs	#Params	#Views	mIoU	aAcc	mAcc
MoCo-v2 (Chen et al., 2020b)	FPN	ResNet50	40k	200	23M	2×224^2	35.8	77.6	45.1
SwAV (Caron et al., 2020)	FPN	ResNet50	40k	200	23M	2×224^2	35.4	77.5	44.9
DenseCL (Wang et al., 2021)	FPN	ResNet50	40k	200	23M	2×224^2	37.2	78.5	47.1
MocoV3 (Chen et al., 2021)	FPN	ViT-S/16	40k	300	23M	2×224^2	35.3	78.9	45.9
MoBY (Xie et al., 2021b)	FPN	ViT-S/16	40k	300	23M	2×224^2	39.5	79.9	50.5
DINO (Caron et al., 2021)	FPN	ViT-S/16	40k	300	23M	2×224^2	38.3	79.0	49.4
DINO (Caron et al., 2021)	UperNet	ViT-S/16	160k	300	23M	2×224^2	42.3	80.4	52.7
SelfPatch (Yun et al., 2022)	FPN	ViT-S/16	40k	200	23M	2×224^2	41.2	80.7	52.1
SelfPatch (Yun et al., 2022)	UperNet	ViT-S/16	160k	200	23M	2×224^2	43.2	81.5	53.9
DINO (Caron et al., 2021)	UperNet	ViT-S/16	160k	800	23M	$2 \times 224^2 + 10 \times 96^2$	44.4	55.5	81.7
iBOT (Zhou et al., 2022)	UperNet	ViT-S/16	160k	200	23M	2×224^2	44.1	55.3	81.4
PQCL (Ours)	UperNet	ViT-S/16	160k	200	23M	2×224^2	**45.2**	**56.0**	**81.9**

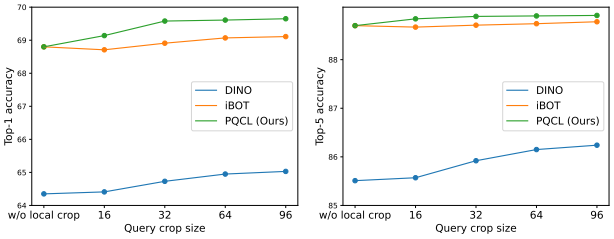

Figure 3. Comparison on ImageNet-1K across different query patch sizes with 100 epochs pretraining.

crop ratio as 0.25, and set the batch size 512, distributed training on ImageNet with 100 epochs in 8 V100 GPUs. We fix the query crop ratio as 0.25 and change the number of patches (16×16) from 1, 4, 9, 16, and 36, where the query image sizes are 16×16, 32×32, 48×48, 64×64, and 96×96, respectively. For fair comparisons, for iBOT (Zhou et al., 2022) and DINO (Caron et al., 2021), we use the query images as local crops proposed in SwAV (Caron et al., 2020). Then we add the local and global objectives in iBOT:

$$\mathcal{L}_{local-global} = \mathcal{H}(sg(g_{\gamma'}(f^{[CLS]}_{\theta'}(\mathbf{x}_A)), g_{\gamma}(f^{[CLS]}_{\theta}(\mathbf{x}_Q)))) \tag{10}$$

where $\mathbf{x}_Q$ is the query crop. We illustrate the top-1 and top-5 classification accuracy in Fig. 3. When w/o query crop, PQCL degenerates to iBOT (fixed positional embedding). We find with the increasing query crop size, both three methods (DINO, iBOT and PQCL) stably increase the downstream classification accuracy. We also find when we set query crop size as 32 (top-1 accuracy is 69.58%), PQCL can get comparable results with crop size as 96 (top-1 accuracy is 69.65%). An intriguing thing is when we set query crop size as 16 (only one patch as the local crop), both iBOT and DINO will drop a little accuracy, while PQCL can improve a lot. The reason may be that the proposed cross-attention scheme between positional embedding and raw patches helps learn the semantic information of the single patch (in each attention block, the query patch and raw patches will perform cross attention to learn the query patch

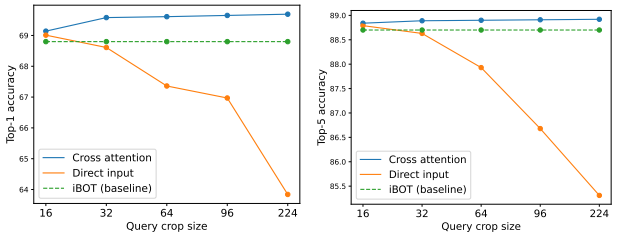

Figure 4. Top-1 and top-5 classification on ImageNet-1K of cross attention and directly inputs with different query crop sizes.

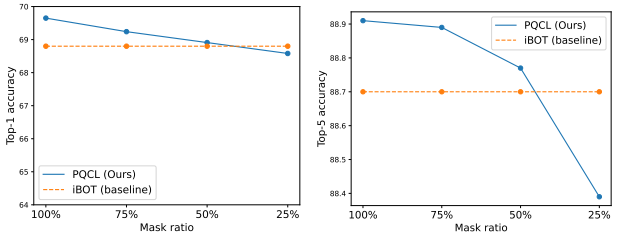

Figure 5. Comparison of mask ratios on ImageNet-1K with 100 epochs pretraining.

information), while for iBOT and DINO, all elements of self attention matrix are equal 1, since there's only one patch.

Effect of cross attention. In our initial experiments, we directly concatenate the global patches (patches divided from global views) and query patches (patches divided from query views), and feed them into transformer and compute the loss. However, as shown in Fig. 4 we find it's heavily sensitive to the patch size, and we conjecture the reason is that the input of pretraining and downstream finetune have different distribution. For example, 96*96 query view will be divided into 36 patches. Then, the inputs in pretraining stage will contain 232 (196 global views and 36 query masked tokens) patches, while for downstream finetune, the inputs will only have 196 patches. As shown in Fig. 4, for direct input, when the number of query patches is larger than 4, PQCL gets lower accuracy than baseline iBOT. And we when we set the size of query view equal to the global view (224*224), the accuracy of self attention (direct input) will drop with a large range. However, when using the cross attention mechanism, the gain becomes larger with the increasing query crop size, since for attention block, the appended query patches will not contribute to raw patches, and do not change the distribution shift of pretraining and downstream inputs. We also find the performance of query patch size 224 has no significant gain against query patch size 96, but only brings more complexity. Therefore, in our main experiments, we set query patch size as 96.

Ablation on query crop ratios. Followed by iBOT (Zhou et al., 2022) and DINO (Caron et al., 2021), for main results, we set the query crop ratio as 0.25 as default. We also conduct a set of experiments to find the best crop ratio.

Table 4. Top-1 and top-5 linear probing classification accuracy of the proposed PQCL under 100 epoch pretraining on ImageNet-1K.

query crop ratio	0.05	0.10	0.15	0.20	0.25	0.40
top-1 accuracy	69.15	69.27	69.36	69.45	**69.65**	69.51
top-5 accuracy	88.77	88.81	88.85	88.92	**89.02**	88.94

Table 5. Video segmentation on DAVIS 2017 of the SOTA self-supervised approaches pre-trained on ImageNet. PQCL-DINO and PQCL-iBOT mean use DINO (Caron et al., 2021) and iBOT (Zhou et al., 2022) as baselines, respectively. All the results are obtained by following the training recipe of SelfPatch (Yun et al., 2022). The metrics $\mathcal{J}_m$, $\mathcal{F}_m$, and $J\&\mathcal{F}_m$ denote mean region similarity, contour-based accuracy, and their average, respectively.

Method	Backbone	$J\&\mathcal{F}_m$	$\mathcal{J}_m$	$\mathcal{F}_m$
MoCo-v2 (Chen et al., 2020b)	ResNet50	55.5	56.0	55.0
SwAV (Caron et al., 2020)	ResNet50	57.4	57.6	57.3
DenseCL (Wang et al., 2021)	ResNet50	50.7	52.6	48.9
ReSim (Xiao et al., 2021)	ResNet50	49.3	51.2	47.3
DetCo (Xie et al., 2021a)	ResNet50	56.7	57.0	56.4
MoCo-v3 (Chen et al., 2021)	ViT-S/16	53.5	51.2	55.9
MoBY (Xie et al., 2021b)	ViT-S/16	54.7	52.0	57.3
DINO (Caron et al., 2021)	ViT-S/16	60.7	59.1	62.4
SelfPatch (Yun et al., 2022)	ViT-S/16	62.7	60.7	64.7
PQCL-DINO (Ours)	ViT-S/16	63.4	61.4	65.3
iBOT (Zhou et al., 2022)	ViT-S/16	61.3	60.1	64.0
PQCL-iBOT (Ours)	ViT-S/16	**63.8**	**61.7**	**65.5**

Specifically, we fix the query crop size as 96 (36 query patches), and train the model in 100 epochs with batch size 512 on 8 V100 GPUs. Table 4 shows the results under different query crop ratios. We change the query crop ratio from [0.05, 0.10, 0.15, 0.20, 0.25, 0.40], and we find PQCL is not sensitive to query crop ratio.

Range of mask ratios. The main idea of PQCL is using the relative positional encoding of query views and the global views to predict the patch-wise information of the query views. At high level, we completely mask the query views (100%), and increase the difficulty of masked image modeling task. To demonstrate PQCL can actually enhance the difficulty of pretraining, we conduct a group of experiments to learn with different mask ratios. Fig. 5 illustrates the top-1 and top-5 linear probing classification accuracy under different mask ratios. With the decrease of mask ratios, both top-1 and top-5 classification accuracy drop. We think that is because with less masked areas, the pretraining is simpler, and hurt the downstream performance. When mask ratio is set 0.25, the accuracy is dropping below baseline iBOT, which may due to that the default mask ratio of iBOT is 0.3, and the 0.25 mask ratio would reduce the difficulty of iBOT pretraining, leading the accuracy decreasing.

Output dimension. We follow the structure of the projection head in DINO and iBOT with l_2-normalized bottleneck and without batch normalization. We study the impact of

Table 6. Linear probing and 10-NN accuracy of PQCL on ImageNet-1K with 100 epochs pretrain by output dimensions.

Method	4096		16384		8192	
	top-1	top-5	top-1	top-5	top-1	top-5
Linear	69.31	88.93	69.01	88.79	**69.65**	**89.07**
KNN (10-NN)	63.92		63.81		64.09	

Table 7. Linear probing classification accuracy of PQCL on ImageNet-1K with 100 epochs pretrain by different architecture.

Method	vanilla		semi-shared		shared	
	top-1	top-5	top-1	top-5	top-1	top-5
iBOT	68.48	88.49	68.16	88.24	68.80	88.70
PQCL	69.06	88.75	69.21	88.81	**69.65**	**89.07**

output dimension K of the last layer. Similar to iBOT and DINO, we do not observe substantial performance gain brought by larger output dimensions. Therefore, we choose K = 8192 in our main experiments by default.

Architecture of projection head. Similar to DINO and iBOT, PQCL use projection head to avoid collapse. We find using a shared projection head between query patches and global images will slightly improve the performance. Note that the head for patch tokens in the student network only see the masked tokens throughout the training, the distribution of which mismatches tokens with natural textures. Therefore, following iBOT, we also conduct an experiment using a non-shared head for the student network but a shared head for the teacher network (since teacher network separately input the query views and global views) denoted as semi-shared head. Specifically, we pretrain PQCL in 100 epochs with 512 batch size. Table 7 shows the results under different shared strategies. Vanilla of PQCL means masked token in global images, $[CLS]$ token of global images and query token in query views of both teacher and student network use separate projection head, while semi-shared of PQCL means only student network use separate projection head, and teacher network use shared one. Different from iBOT (semi-shared architecture would hurt the performance), semi-shared architecture in PQCL can also improve a little accuracy, as in our teacher branch, we separately input global view and query views.

4.5. Connection to Peer Methods

Relation to iBOT (Zhou et al., 2022). Similarity. iBOT is the baseline of PQCL, and both iBOT and PQCL learn to predict masked patches in latent space. **Difference.** iBOT learn to predict masked patches of the same view (*e.g.*, $\mathbf{X}_A \rightarrow \mathbf{X}_{Am}$, where $\mathbf{X}_{Am}$ means the masked version of view A with masked ratio 30%). In contrast, PQCL use the global view to learn a completely masked query view (*e.g.*, $\mathbf{X}_Q \rightarrow (\mathbf{X}_A, \mathbf{M}_{Qm})$, where $\mathbf{M}_{Qm}$ means the completely masked query views and the relative positional embedding.), which would increase the difficulty of pretraining.

Relation to SIM (Tao et al., 2022). Similarity. Both SIM and PQCL learn to predict masked patches by relative positional embeddings. **Differences.** SIM directly uses the plain ViT and introduces a decoder network to predict the masked patches, which would bring more parameters and computational costs. In contrast, directly using the ViT in PQCL would make PQCL heavily sensitive to patch size due to the distribution shift of inputs in pretraining and finetuning stage. Hence, PQCL designs the cross attention mechanism in each attention block of the transformer encoder between the completely masked query view and the global view. Besides, the cross attention mechanism also makes PQCL learn without extra learnable parameters.

Relation to ADCLR (Zhang et al., 2023). Similarity. Both ADCLR and PQCL use cross attention mechanisms to learn the query patches information. **Differences.** The main idea of ADCLR is to use the query patches to replace the $[CLS]$ token to learn accurate and spatial-sensitive information. Therefore, ADCLR uses query patches with pixel information as query in both student and teacher networks. However, this manner would reduce the difficulty of pretraining. In contrast, in our PQCL, we input the completely masked query view to the student network and the query view with pixel information to the teacher network to increase the variance and the difficulty of pretraining.

5. Conclusion

We have proposed PQCL, to perform patch-level contrasts without patch correspondence via masked positional query view. Besides, the well-designed cross attention between positional embedding and raw patches makes it applicable to SOTA transformer-based contrastive methods (*e.g.*, DINO, iBOT) with further improvement, especially on dense prediction tasks. Experiments on image classification, object detection and segmentation on various public benchmarks have shown its effectiveness. Specifically, For ViT/B, PQCL outperforms iBOT 0.9% top-1 accuracy by linear probing on ImageNet. For ViT/S, PQCL outperforms baseline iBOT by 2.4mAPbb and 1.5% mAPmk on detection and segmentation on MS-COCO dataset, respectively. For semantic segmenta- tion, PQCL outperforms iBOT 0.8% mIoU on ADE20K. Finally, we conduct comprehensive ablation studies to demonstrate the robustness of PQCL.

Acknowledgments

The work was partly supported by NSFC (62222607) and Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102).

References

Cai, Z. and Vasconcelos, N. Cascade r-cnn: Delving into high quality object detection. In *CVPR*, 2018.

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

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

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

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

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

Chen, X., Ding, M., Wang, X., Xin, Y., Mo, S., Wang, Y., Han, S., Luo, P., Zeng, G., and Wang, J. Context autoencoder for self-supervised representation learning. *arXiv preprint arXiv:2202.03026*, 2022.

Dalal, N. and Triggs, B. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

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

Dong, X., Bao, J., Zhang, T., Chen, D., Zhang, W., Yuan, L., Chen, D., Wen, F., and Yu, N. Peco: Perceptual codebook for bert pre-training of vision transformers. *arXiv preprint arXiv:2111.12710*, 2021.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2020.

Fang, Y., Dong, L., Bao, H., Wang, X., and Wei, F. Corrupted image modeling for self-supervised visual pre-training. *arXiv preprint arXiv:2202.03382*, 2022.

Ge, C., Liang, Y., Song, Y., Jiao, J., Wang, J., and Luo, P. Revitalizing cnn attention via transformers in self-supervised visual representation learning. *NeurIPS*, 2021.

Gidaris, S., Singh, P., and Komodakis, N. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. *NeurIPS*, 27, 2014.

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

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *CVPR*, 2016.

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

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

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377*, 2021.

Jiang, B., Luo, R., Mao, J., Xiao, T., and Jiang, Y. Acquisition of localization confidence for accurate object detection. In *ECCV*, 2018.

Jin, P., Huang, J., Liu, F., Wu, X., Ge, S., Song, G., Clifton, D., and Chen, J. Expectation-maximization contrastive learning for compact video-and-language representations. *NeurIPS*, 2022.

Jin, P., Huang, J., Xiong, P., Tian, S., Liu, C., Ji, X., Yuan, L., and Chen, J. Video-text as game players: Hierarchical banzhaf interaction for cross-modal representation learning. *CVPR*, 2023a.

Jin, P., Li, H., Cheng, Z., Li, K., Ji, X., Liu, C., Yuan, L., and Chen, J. Diffusionret: Generative text-video retrieval with diffusion model. *arXiv preprint arXiv:2303.09867*, 2023b.

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

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. Feature pyramid networks for object detection. In *CVPR*, 2017.

Loshchilov, I. and Hutter, F. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

Loshchilov, I. and Hutter, F. Fixing weight decay regularization in adam. 2018.

Pont-Tuset, J., Perazzi, F., Caelles, S., Arbeláez, P., Sorkine-Hornung, A., and Van Gool, L. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017.

Tao, C., Zhu, X., Su, W., Huang, G., Li, B., Zhou, J., Qiao, Y., Wang, X., and Dai, J. Siamese image modeling for self-supervised vision representation learning, 2022.

Van Horn, G., Mac Aodha, O., Song, Y., Cui, Y., Sun, C., Shepard, A., Adam, H., Perona, P., and Belongie, S. The inaturalist species classification and detection dataset. In *CVPR*, 2018.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *NeurIPS*, 2017.

Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P.-A. Extracting and composing robust features with denoising autoencoders. In *ICML*, 2008.

Wang, X., Zhang, R., Shen, C., Kong, T., and Li, L. Dense contrastive learning for self-supervised visual pretraining. In *CVPR*, 2021.

Wang, X., Fan, H., Tian, Y., Kihara, D., and Chen, X. On the importance of asymmetry for siamese representation learning. In *CVPR*, 2022a.

Wang, Z., Li, Q., Zhang, G., Wan, P., Zheng, W., Wang, N., Gong, M., and Liu, T. Exploring set similarity for dense self-supervised representation learning. In *CVPR*, 2022b.

Wei, C., Fan, H., Xie, S., Wu, C.-Y., Yuille, A., and Feichtenhofer, C. Masked feature prediction for self-supervised visual pre-training. *arXiv preprint arXiv:2112.09133*, 2021.

Xiao, T., Liu, Y., Zhou, B., Jiang, Y., and Sun, J. Unified perceptual parsing for scene understanding. In *ECCV*, 2018.

Xiao, T., Reed, C. J., Wang, X., Keutzer, K., and Darrell, T. Region similarity representation learning. In *ICCV*, 2021.

Xie, E., Ding, J., Wang, W., Zhan, X., Xu, H., Sun, P., Li, Z., and Luo, P. Detco: Unsupervised contrastive learning for object detection. In *ICCV*, 2021a.

Xie, Z., Lin, Y., Yao, Z., Zhang, Z., Dai, Q., Cao, Y., and Hu, H. Self-supervised learning with swin transformers. *arXiv preprint arXiv:2105.04553*, 2021b.

Xie, Z., Lin, Y., Zhang, Z., Cao, Y., Lin, S., and Hu, H. Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning. In *CVPR*, 2021c.

Xie, Z., Zhang, Z., Cao, Y., Lin, Y., Bao, J., Yao, Z., Dai, Q., and Hu, H. Simmim: A simple framework for masked image modeling. *arXiv preprint arXiv:2111.09886*, 2021d.

Yang, C., Wu, Z., Zhou, B., and Lin, S. Instance localization for self-supervised detection pretraining. In *CVPR*, 2021.

Yun, S., Lee, H., Kim, J., and Shin, J. Patch-level representation learning for self-supervised vision transformers. In *CVPR*, 2022.

Zbontar, J., Jing, L., Misra, I., LeCun, Y., and Deny, S. Barlow twins: Self-supervised learning via redundancy reduction. In *ICML*, 2021.

Zhang, R., Isola, P., and Efros, A. A. Colorful image colorization. In *ECCV*, 2016.

Zhang, S., Zhu, F., Yan, J., Zhao, R., and Yang, X. Zero-cl: Instance and feature decorrelation for negative-free symmetric contrastive learning. In *ICLR*, 2021.

Zhang, S., Qiu, L., Zhu, F., Yan, J., Zhang, H., Zhao, R., Li, H., and Yang, X. Align representations with base: A new approach to self-supervised learning. In *CVPR*, 2022.

Zhang, S., Zhu, F., Zhao, R., and Yan, J. Patch-level contrasting without patch correspondence for accurate and dense contrastive representation learning. In *ICLR*, 2023. URL https://openreview.net/forum?id=10R_bcjFwJ.

Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., and Torralba, A. Scene parsing through ade20k dataset. In *CVPR*, 2017.

Zhou, J., Wei, C., Wang, H., Shen, W., Xie, C., Yuille, A., and Kong, T. Image BERT pre-training with online tokenizer. In *ICLR*, 2022.

Ziegler, A. and Asano, Y. M. Self-supervised learning of object parts for semantic segmentation. In *CVPR*, 2022.