Supplementary Material

Patch-level Representation Learning for Self-supervised Vision Transformers

A. Pre-training details

For unsupervised pre-training, we use the ImageNet [7] dataset for large-scale pre-training (see Sec. 4) and the MS COCO [16] dataset with the `train2017` split for medium-scale pre-training (see Sec. 5). Code is available at https://github.com/alinlab/SelfPatch.

ImageNet pre-training details. In Sec. 4, we perform unsupervised pre-training using ViT-S/16 [20] on the ImageNet [7] dataset for 200 epochs with a batch size of 1024. In the case of the joint usage of DINO [2] and our method (*i.e.*, DINO + SelfPatch), we generally follow the training details of Caron *et al*. [2], including the optimizer and the learning rate schedule. Specifically, we use the AdamW [18] optimizer with a linear warmup of the learning rate during the first 10 epochs, and the learning rate is decayed with a cosine schedule. We also follow the linear scaling rule [9]: $lr = 0.0005 \cdot \text{batchsize}/256$. We use 2 global crops and 8 local crops (*i.e.*, $2 \times 224^2 + 8 \times 96^2$) for multi-crop augmentation [1, 2]. For our aggregation module, we use two class-attention blocks [21] without Layerscale normalization [21]. For the final output dimension of the projection head, we use $K = 65536$ for the SSL projection head following Caron *et al*. [2] and $K = 4096$ for our projection head. In Sec. 4, we use a publicly available DINO pre-trained model[1] with 300 training epochs on the ImageNet as the baseline, which also use the same hyperparameters as the above.

MS COCO pre-training details. In Sec. 5, we perform unsupervised pre-training using ViT-Ti/16 [20] on the MS COCO [16] dataset with `train2017` split for 200 training epochs with a batch size of 256. In the case of the joint usage of DINO [2] and our method (*i.e.*, DINO + SelfPatch), we use 2 global crops and 2 local crops (*i.e.*, $2 \times 224^2 + 2 \times 96^2$) for the multi-crop augmentation and $K = 4096$ for both SSL and SelfPatch projection head. In the case of the joint usage of MoBY [26] and our method (*i.e.*, MoBY + SelfPatch), we also perform the pre-training for 200 training epochs with a batch size of 256, and follow the training details of Xie *et al*. [26] for both ViT-Ti/16 [20] and Swin-T [17].

B. Evaluation details

For evaluation, we perform object detection and instance segmentation on MS COCO [16], semantic segmentation on ADE20K [27], and video object segmentation on DAVIS-2017 [19].

COCO object detection and instance segmentation. MS COCO [16] is large-scale object detection, segmentation, and captioning dataset: in particular, `train2017` and `val2017` splits contain 118K and 5K images, respectively. We follow the basic configuration of `mmdetection`[2] [3] for fine-tuning Mask R-CNN [10] with FPN [15] under the standard `1x` schedule. In addition, we follow several details of El-Nouby *et al*. [8] for integrating Mask R-CNN with ViT-S/16.

ADE20K semantic segmentation. ADE20K [27] is a semantic segmentation benchmark containing 150 fine-grained semantic categories and 25k images. We follow all the configurations of `mmsegmentation`[3] [6] for fine-tuning Semantic FPN [12] with 40k iterations and an input resolution of 512×512. We also perform large-scale fine-tuning experiments using UPerNet [25] with 160k iterations and an input resolution of 512×512 in Appendix C.

DAVIS 2017 video object segmentation. DAVIS 2017 [19] is a video object segmentation dataset containing 60 training, 30 validation, and 60 testing videos. We follow the evaluation protocol of Jabri [11] and Caron *et al*. [2], which evaluates the quality of frozen representations of image patches by segmenting scenes with the nearest neighbor between consecutive frames.

C. UPerNet on ADE20K semantic segmentation

Here, we additionally evaluate semantic segmentation performances of DINO and DINO + SelfPatch for a large-scale fine-tuning setup, *i.e.*, a larger network and longer iterations. Specifically, we use UPerNet [25] with 160k iterations following Liu *et al*. [17], while Wang [22] and El-Nouby *et al*. [8] do use Semantic FPN [12] with 40k iterations as we follow originally. Tab. 1 summarizes the results. We emphasize that DINO + SelfPatch still achieves consistent improvements over DINO in

[1] https://github.com/facebookresearch/vissl.
[2] https://github.com/open-mmlab/mmdetection.
[3] https://github.com/open-mmlab/mmsegmentation.

all the metrics; *e.g*., DINO + SelfPatch achieves 0.9, 1.1, and 1.2 points higher than DINO in terms of the mIoU, aAcc, and mAcc metrics, respectively. This comparison under the large-scale fine-tuning setup also verifies the effectiveness of SelfPatch.

Method	Network	Param.(M)	Iteration	mIoU	aAcc	mAcc
DINO [2]	ViT-S/16 + Semantic FPN	26	40k	38.3	79.0	49.4
+ SelfPatch (ours)	ViT-S/16 + Semantic FPN	26	40k	**41.2**	**80.7**	**52.1**
DINO [2]	ViT-S/16 + UPerNet	58	160k	42.3	80.4	52.7
+ SelfPatch (ours)	ViT-S/16 + UPerNet	58	160k	**43.2**	**81.5**	**53.9**

Table 1. **Transferring performances to ADE20K semantic segmentation** using Semantic FPN [12] and UPerNet [25] with 40k and 160k iterations, respectively. All models are pre-trained on the ImageNet [7] dataset using ViT-S/16. The metrics, mIoU, aAcc, and mAcc, denote the mean intersection of union, all pixel accuracy, and mean class accuracy, respectively.

D. Linear classification on ImageNet

We here evaluate the quality of pre-trained representations for the image classification task under the conventional linear evaluation protocol [2, 4, 24]. Specifically, we train a supervised linear classifier on top of frozen features without the projection head following the details of Caron *et al*. [2]; we use the SGD optimizer with a batch size of 1024 during 100 training epochs and report central-crop top-1 accuracy. Tab. 2 summarizes the results. Here, we would like to emphasize that our primary applications of interest are dense prediction tasks (i.e., not classification tasks), where patch-level representation learning can be more effective. Nevertheless, DINO + SelfPatch can outperform DINO even for ImageNet classification under the same 300 training epochs; ours and DINO achieve 75.6% and 75.1%, respectively. Also, DINO + SelfPatch consistently outperforms other self-supervised ViT baselines: MoCo-v3 [5] and MoBY [26]. It shows that our method is not only able to enhance the performances on dense prediction tasks, but also maintain competitive performance on image classification.

Method	Backbone	Epoch	Top-1 acc.
MoCo-v3 [5]	ViT-S/16	300	73.2
MoBY [26]	ViT-S/16	300	72.8
DINO [2]	ViT-S/16	300	75.1
+ SelfPatch (ours)	ViT-S/16	300	**75.6**

Table 2. **ImageNet linear classification** performances of the recent self-supervised ViTs pre-trained on the ImageNet [7] benchmark. We train a supervised linear classifier on top of frozen features and report central-crop top-1 accuracy.

E. Comparison with concurrent work

Concurrent to our work, EsViT [13] introduces the region-matching (*i.e*., matching image patches) task for Vision Transformers that considers the region correspondence (*i.e*., matching the two most similar regions) between two differently augmented images. In particular, the region-matching task also has been investigated for ResNet; for example, DenseCL [23] also matches the two most similar spatial representations between the two augmented images. One key difference from our method is that the region-matching task finds positive pairs from two strongly augmented images, which necessarily requires overlapping regions and may find noisy positives (*i.e*., not positives) in early training, while our method utilizes adjacent patches in the same augmented image as the positives, which is a reasonable way to find reliable positives without constraints of overlapping regions.

To further compare our method with the region matching task, we pre-train EsViT using ViT-Ti/16 on the MS COCO dataset [16] (*i.e*., the same training details in Appendix A), and perform three evaluation downstream tasks: (a) COCO object detection and instance segmentation, (b) ADE20K semantic segmentation, and (c) DAVIS 2017 video object segmentation. As shown in Tab. 3, our method consistently outperforms EsViT with a large margin in all the metrics, *e.g*., (a) +2.8 AP^{bb} on COCO detection, +1.7 AP^{mk} on COCO detection, (b) +3.5 mIoU on ADE20K segmentation, and (c) +3.5 $(\mathcal{J}\&\mathcal{F})_m$ on DAVIS segmentation. We believe that restricting positive candidates to neighboring patches plays an essential role in constructing effective patch-level self-supervision, and this work would guide a new research direction for patch-level self-supervised learning.

Method	Backbone	COCO Detection AP^{bb}	$\text{AP}^{\text{bb}}_{50}$	$\text{AP}^{\text{bb}}_{75}$	COCO Segmentation AP^{mk}	$\text{AP}^{\text{mk}}_{50}$	$\text{AP}^{\text{mk}}_{75}$	ADE20K Segmentation mIoU	aAcc	mAcc	DAVIS Segmentation $(\mathcal{J}\&\mathcal{F})_m$	$\mathcal{J}_m$	$\mathcal{F}_m$
DINO [2]	ViT-Ti/16	28.0	48.8	28.4	26.9	45.8	27.7	24.9	73.4	33.3	55.1	52.8	57.4
+ SelfPatch (ours)	ViT-Ti/16	**30.7**	**51.4**	**32.2**	**28.6**	**48.2**	**29.6**	**29.5**	**75.5**	**39.2**	**57.0**	**56.1**	**57.8**
EsViT [13]	ViT-Ti/16	27.9	49.0	28.0	26.9	45.9	27.7	26.0	73.5	34.5	53.5	50.8	56.2

Table 3. **Transferring performances to various downstream tasks**: COCO object detection and instance segmentation, ADE20K semantic segmentation, and DAVIS 2017 video object segmentation. All models are pre-trained on the MS COCO [16] dataset with `train2017` split using ViT-Ti/16. We use the same evaluation details in Appendix B.

F. Importance of positional encoding

In this section, we investigate the importance of positional encoding (PE) in a dense prediction task, similar to Chen *et al*. [5]. Specifically, we pre-train ViT-S/16 models on COCO with or without PE, and evaluate their segmentation performances on DAVIS. Table 4 shows that learning PE is still effective even under SelfPatch, while SelfPatch consistently improves the performance regardless of PE. It implies that the role of positional inductive bias in a dense prediction task would be quite important, and our method, SelfPatch, orthogonally contributes to improving patch-level representations.

Method	PE	$(\mathcal{J}\&\mathcal{F})_m$	$\mathcal{J}_m$	$\mathcal{F}_m$
DINO	✓	55.1	52.8	57.4
DINO + SelfPatch	✓	**57.0**	**56.1**	**57.8**
DINO		51.7	49.5	54.0
DINO + SelfPatch		**52.9**	**50.5**	**55.2**

Table 4. **Importance of positional encoding (PE).** All models are pretrained on the MS COCO [16] dataset with `train2017` split using ViT-S/16. We use the same evaluation details for DAVIS 2017 video object segmentation in Appendix B.

G. Effects of the number of positive patches under varying patch sizes

We primarily focus on the popular setup of 224×224 images and 16×16 patches, where $k = 4$ works as we validated throughout the paper. However, this choice may not be optimal for other setups; we additionally perform an ablation study on a different dataset, ImageNet-10 [14], with 8×8, 16×16 and 32×32 patches from 224×224 images. Table 5 shows their segmentation performances on DAVIS and 20-NN (*i.e.,* 20 nearest neighbor classifier) classification performances following Caron *et al*. [2]. Overall, it suggests that the effective number of positives may depend on the relative size of patches in an image. For example, $k = 4, 6$ achieves the best performance for 8×8 and 16×16 patch sizes on both the dense prediction and the classification tasks, while $k = 2$ does for the patch size 32×32. This is because smaller patches would contain more positive patches in their neighbors. Hence, we recommend to use $k = 4$ in general cases (*e.g.*, 8×8 and 16×16), but $k = 2$ when considering a larger patch size (*e.g.*, 32×32) for 224×224 images.

Method (Patch size)	k	$(\mathcal{J}\&\mathcal{F})_m$ 32×32	16×16	8×8	Acc. 32×32	16×16	8×8
DINO	-	24.9	37.6	48.7	76.0	83.8	85.0
DINO + SelfPatch	1	34.9	46.5	50.6	80.0	85.0	87.0
DINO + SelfPatch	2	**36.5**	52.9	57.3	**80.2**	**86.0**	88.0
DINO + SelfPatch	4	36.3	**53.1**	61.7	75.0	85.8	**88.4**
DINO + SelfPatch	6	36.3	52.6	**62.0**	75.6	85.0	87.4
DINO + SelfPatch	8	33.2	50.4	60.4	75.2	82.6	87.2

Table 5. Effects of the positive number k under varying the different patch sizes. All models are pre-trained on ImageNet-10 [14], and evaluated on DVAIS video segmentation and ImageNet-10 classification.

In addition, we count the number of positive patches in the COCO [16] validation images by using their ground-truth segmentation labels and found 4.3 ± 1.2 ($k \approx 4$) adjacent positives (on average) for 16×16 patches from 224×224 images.

Here, we measure the cosine similarities among adjacent patches and use the threshold of 0.95 for counting the positives. Interestingly, we observe that further utilizing the ground-truth segmentation labels for the adjacent positive selection can improve ours from 57.0 to 59.5 $(\mathcal{J}\&\mathcal{F})_m$ score on DAVIS [19]. We believe that developing an unsupervised adaptive selection scheme on k would be an interesting direction to explore.

References

[1] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*, 2020. 1

[2] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *arXiv preprint arXiv:2104.14294*, 2021. 1, 2, 3

[3] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 1

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 2

[5] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. *arXiv preprint arXiv:2104.02057*, 2021. 2, 3

[6] MMSegmentation Contributors. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. `https://github.com/open-mmlab/mmsegmentation`, 2020. 1

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 1, 2

[8] Alaaeldin El-Nouby, Hugo Touvron, Mathilde Caron, Piotr Bojanowski, Matthijs Douze, Armand Joulin, Ivan Laptev, Natalia Neverova, Gabriel Synnaeve, Jakob Verbeek, et al. Xcit: Cross-covariance image transformers. *arXiv preprint arXiv:2106.09681*, 2021. 1

[9] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017. 1

[10] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 1

[11] Allan Jabri, Andrew Owens, and Alexei A Efros. Space-time correspondence as a contrastive random walk. *arXiv preprint arXiv:2006.14613*, 2020. 1

[12] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6399–6408, 2019. 1, 2

[13] Chunyuan Li, Jianwei Yang, Pengchuan Zhang, Mei Gao, Bin Xiao, Xiyang Dai, Lu Yuan, and Jianfeng Gao. Efficient self-supervised vision transformers for representation learning. *arXiv preprint arXiv:2106.09785*, 2021. 2, 3

[14] Yunfan Li, Peng Hu, Zitao Liu, Dezhong Peng, Joey Tianyi Zhou, and Xi Peng. Contrastive clustering. *arXiv preprint arXiv:2009.09687*, 2020. 3

[15] 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. 1

[16] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 1, 2, 3

[17] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021. 1

[18] Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam. 2018. 1

[19] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017. 1, 4

[20] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. *arXiv preprint arXiv:2012.12877*, 2020. 1

[21] Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. *arXiv preprint arXiv:2103.17239*, 2021. 1

[22] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. *arXiv preprint arXiv:2102.12122*, 2021. 1

[23] 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. 2

[24] Zhirong Wu, Yuanjun Xiong, Stella Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance-level discrimination. In *CVPR*, 2018. 2

[25] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 418–434, 2018. 1, 2

[26] Zhenda Xie, Yutong Lin, Zhuliang Yao, Zheng Zhang, Qi Dai, Yue Cao, and Han Hu. Self-supervised learning with swin transformers. *arXiv preprint arXiv:2105.04553*, 2021. 1, 2

[27] 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. 1