Revisiting Pixel-Level Contrastive Pre-Training on Scene Images: Supplementary Material

Zongshang Pang[1] Yuta Nakashima[1] Mayu Otani[2] Hajime Nagahara[1]

[1] Osaka University [2] CyberAgent, Inc.

A. More quantitative results

A.1. Transfer results with standard deviations.

In Table 1 of the main text, we reported transfer results across four downstream tasks. As reported in previous work [1, 4], the evaluation results on such transfer benchmarks have certain variances across different runs of fine-tuning. Therefore, for all the reproducible models in Table 1 of the main text, we report their average results together with associated standard deviations calculated from 5, 3, 3, and 5 independent runs on PASCAL VOC detection, COCO object detection & instance segmentation, Cityscapes segmentation, and PASCAL VOC segmentation in Tables 1-3.

Table 1. PASCAL VOC object detection results.

Method	AP	AP_{50}	AP_{75}
BYOL	55.73 ± 0.28	81.77 ± 0.16	61.60 ± 0.25
MoCo-V2+	54.63 ± 0.26	81.42 ± 0.16	60.48 ± 0.27
ORL	55.82 ± 0.30	82.09 ± 0.20	62.28 ± 0.35
Odin	56.91 ± 0.24	82.43 ± 0.22	63.29 ± 0.41
SlotCon	54.47 ± 0.31	81.88 ± 0.17	60.29 ± 0.51
PixCon-Sim	57.34 ± 0.24	82.36 ± 0.15	63.94 ± 0.34
PixCon-Coord	57.21 ± 0.26	82.61 ± 0.17	63.40 ± 0.29
PixCon-SR	57.55 ± 0.25	82.83 ± 0.18	64.04 ± 0.27

Table 2. Cityscapes segmentation & PASCAL VOC segmentation.

Method	City. Seg.	VOC Seg.
	mIoU	mIoU
BYOL	75.28 ± 0.15	70.21 ± 0.28
MoCo-V2+	75.62 ± 0.08	71.07 ± 0.23
ORL	75.43 ± 0.18	70.71 ± 0.33
Odin	75.72 ± 0.09	70.77 ± 0.31
SlotCon	76.11 ± 0.04	71.65 ± 0.26
PixCon-Sim	76.11 ± 0.15	72.64 ± 0.28
PixCon-Coord	75.83 ± 0.17	72.31 ± 0.32
PixCon-SR	76.62 ± 0.10	72.95 ± 0.29

A.2. A step-by-step investigation from DenseCL to PixCon-Sim.

After applying the MoCo-v2+/BYOL training pipeline, MoCo-v2-based DenseCL becomes PixCon-Sim, which delivers consistently better transfer performance. It is thus interesting to investigate which newly introduced component in the new pipeline is contributing to the better transfer performance.

As shown in Table 4, SyncBN can be used to replace the ShuffleBN in MoCo-v2 without affecting much the transfer performance. Asymmetric predictors did not have an apparent contribution. Momentum ascending, symmetric loss, and BYOL augmentation all contribute to better transfer performance, which is consistent with the observation made in the MoCo-v2+ paper [2]. However, we found that symmetric loss and BYOL augmentation deliver a more consistent performance boost when applied together.

Though asymmetric predictors and SyncBN did not improve the transfer performance, they have been shown in [2] to contribute to linear probing accuracy on the pre-training dataset. If linear probing accuracy is not considered, it might be interesting to investigate the effect of removing these two techniques. However, to align with previous region-level methods, which invariantly incorporate all the BYOL components, we do so as well by default and leave the investigation for future work.

A.3. SlotCon&PixPro with image-level loss.

DenseCL [4] and the proposed PixCon framework both require the image-level loss to work well. However, for the SoTA region-level methods, SlotCon [5] and PixPro [7], the former does not contain an image-level loss while the latter does not use it by default. Therefore, we would like to investigate whether an additional image-level loss will help these two methods. The experiments are based on the officially released codes of SlotCon [1] and PixPro [2]

For SlotCon, we add an additional image-level branch

[1] https://github.com/CVMI-Lab/SlotCon.
[2] https://github.com/zdaxie/PixPro.

Table 3. COCO object detection & instance segmentation.

Method	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{mk}	AP^{mk}_{50}	AP^{mk}_{75}
BYOL	39.53 ± 0.12	59.41 ± 0.07	43.33 ± 0.24	35.62 ± 0.10	56.56 ± 0.14	38.17 ± 0.12
MoCo-V2+	39.78 ± 0.08	59.74 ± 0.05	43.58 ± 0.18	35.92 ± 0.10	56.95 ± 0.09	38.48 ± 0.13
ORL	40.24 ± 0.15	60.02 ± 0.12	44.31 ± 0.23	36.39 ± 0.15	57.38 ± 0.14	38.82 ± 0.15
Odin	40.42 ± 0.10	60.43 ± 0.11	44.56 ± 0.18	36.55 ± 0.13	57.48 ± 0.10	39.34 ± 0.12
SlotCon	40.81 ± 0.09	61.03 ± 0.06	44.79 ± 0.16	36.78 ± 0.11	57.99 ± 0.06	39.52 ± 0.14
PixCon-Sim	40.53 ± 0.06	60.52 ± 0.10	44.19 ± 0.22	36.64 ± 0.06	57.54 ± 0.11	39.24 ± 0.15
PixCon-Coord	40.27 ± 0.10	60.25 ± 0.09	43.91 ± 0.23	36.47 ± 0.13	57.41 ± 0.09	39.18 ± 0.14
PixCon-SR	40.81 ± 0.09	60.97 ± 0.04	44.80 ± 0.23	36.84 ± 0.11	57.93 ± 0.12	39.62 ± 0.17

Table 4. Investigating the effect of components in MoCo-v2+/BYOL on DenseCL's transfer performance.

Method	COCO		VOC Seg.
	AP^{bb}	AP^{mk}	mIoU
DenseCL	39.6	35.7	71.6
+ SyncBN	39.6	35.6	71.7
+ Asymmetric Predictor	39.6	35.7	71.7
+ Momentum Ascending	40.1	36.2	72.1
+ Symmetric Loss	40.3	36.4	71.5
+ BYOL Aug. (**PixCon-Sim**)	40.5	36.6	72.6
- Symmetric Loss	39.8	36.0	72.2

consisting of a projector and a predictor to the original architecture. We then applied to SlotCon the MoCo-v2(+) loss as the image-level loss, which utilizes a momentum queue for storing negative keys. The weight for the image-level loss is set as the same as those for the other two loss terms in SlotCon. As shown in Table 5, SlotCon does not benefit from the additional image-level learning. Before conducting this experiment, we made sure that we could use the official code to reproduce the results we obtained by evaluating the officially released weights.

For PixPro without the image/instance-level loss, which is the default setting in the released pre-training script, both the COCO detection and VOC segmentation results have major gaps compared to those reported in Table 1 in the main text, which may be the version with instance-level loss reproduced by SlotCon authors. After we set the weight for the instance branch from 0 to 1, as per the PixPro paper, and conducted pre-training with the instance branch, the reproduced results matched what had been reported in terms of COCO but failed in terms of VOC segmentation. The results are listed in Table 5. There could be some potential reasons for this. For one, PixPro was originally trained with a large batch size (1024), but we used a batch size of 512 for a fair comparison with PixCon and SlotCon. For another, it is described in the PixPro paper that the SimCLR loss was used for instance-level learning, while in reality BYOL's cosine-similarity-based loss is applied.

Table 5. SlotCon&PixCon with image-level losses.

Method	COCO		VOC Seg.
	AP^{bb}	AP^{mk}	mIoU
SlotCon	40.8	36.8	71.7
SlotCon + image	40.5	36.6	70.2
PixPro	40.1	36.1	71.0
PixPro + image	40.5	36.6	69.8

A.4. Attempts to relax the use of prior knowledge in region-level learning.

Among the region-level learning methods, there are two that also consider pixel-level features, *i.e.*, PixPro and SlotCon. As opposed to pure pixel-level learning applied in DenseCL and the proposed PixCon, PixPro applies pixel-to-region matching based on self-attention to explicitly learn regional semantics. On the other hand, SlotCon enforces pixel-level features to be grouped under learnable prototypes, the number of which is tuned for them to capture region-level semantics. Besides, SlotCon also applies an attention-based region-level loss. The common first step between pixel or pixel-to-region losses is to find pixel-level positive matches. DenseCL and PixCon find such matches mainly by bootstrapping feature similarities, while PixPro and SlotCon utilize a safer source of information based on prior knowledge, *i.e.*, spatial coordinates.

As we have discussed in Section 5.3 in the main text, similarity-based matching encourages learning regional semantics more than coordinate-based matching. Thus, if we desire to learn regional semantics *without explicitly applying region-level learning*, similarity-based matching is the key. PixPro and SlotCon are equipped with coordinate-based matching, but they need to explicitly leverage region-level losses. One question that naturally comes to mind is: Will similarity-based matching facilitate *explicit* region-level learning? In other words, we may want to know whether it helps to augment/replace the coordinate-based matching in PixPro or SlotCon with bootstrapping-driven similarity-based matching. We have made several attempts

in this direction but did not witness any improvements. The results are shown in Table 6. We provide our analyses of the results below.

SlotCon+Pix. means that we augment SlotCon with an additional pixel-level learning branch, for which we apply the PixCon pixel-level loss (without semantic reweighting). We can observe that simply augmenting SlotCon with similarity-based pixel-level learning does not help. **SlotCon-Coord.+Sim.** means that we replace the coordinate-based matching with similarity-based matching, and this scenario leads to a significant performance drop. This is expected as similarity-based matching needs the image-level loss as a basis for semantically meaningful features, whereas SlotCon's region-level loss, similar to the similarity-based matching, also relies on bootstrapping feature similarities. Therefore, the scenario, **SlotCon-Coord.+Sim.+Img.**, where the image-level loss is added, shows a more reasonable performance, which still does not match the original performance. Moreover, as shown in Table 5, SlotCon does not benefit from the image-level loss to begin with. When we tried to augment the original coordinate-based loss with the similarity-based loss on the same branch (**SlotCon+Sim.**), we observed a similar performance drop. Semantic reweighting (SR) helps regain part of the original performance. We observe similar trends for PixPro but only report SlotCon results here as we have only managed to verify the reproducibility of SlotCon's code.

What could account for the failure? Compared to the straightforward pixel-level loss in PixCon, SlotCon as well as PixPro takes a step forward to further bootstrap feature similarities/attentions for conducting region-level learning. Compared to similarity-based matching, which is already driven by bootstrapping, coordinate-based matching is apparently a safer tool for providing better semantically meaningful features, at least at the initial stage, to support such region-level bootstrapping. Semantic reweighting helps avoid part of the negative effect of bootstrapping by incorporating spatial information, but it still relies on similarity-based matching.

Similar to PixPro and SlotCon, the proposed PixCon framework is another step towards making dense representation learning less restricted by human prior knowledge via relying more on bootstrapping. Attempting to combine PixCon and region-level bootstrapping is yet another effort in the same direction, but remains challenging for now and interesting for future work.

A.5. COCO+ results.

To investigate whether PixCon-SR can further benefit from more scene-centric training images, we conduct pre-training with the COCO+ dataset and provide the corresponding transfer results in Table 7.

We can observe that all the reported methods have gained from leveraging more scene-centric images for pre-training. It is interesting to see that SlotCon has substantially better performance on VOC detection, COCO detection, instance segmentation, and VOC segmentation (though we did not manage to reproduce the numbers exactly the same as those in the original paper, the differences are reasonably small). UniVIP also witnessed an impressive performance boost on VOC detection after utilizing COCO+ for pre-training.

PixCon-SR experienced consistent transfer performance improvements across the benchmarks and remains competitive compared to region-level methods. Interestingly, PixCon-SR falls behind SlotCon on ADE20k when pre-trained on COCO, but catches up after COCO+ pre-training. SlotCon has a smaller relative improvement on ADE20k after pre-training on COCO+ compared to that of PixCon-SR.

Overall, region-level methods seem to have relatively more performance improvements on some but not all benchmarks after utilizing more scene-centric training images. One assumption for this is that they may be good at capturing the distribution of semantics in the pre-training datasets, which can benefit the transfer to downstream tasks where the datasets have similar distributions. PixCon-SR, instead, does not enjoy drastic performance boosts on specific datasets, but its transferability can indeed consistently generalize to various downstream tasks and is overall competitive to that of region-level methods.

Table 6. Attempts to combine similarity-based matching with SlotCon. See text for analyses.

Method	COCO		VOC Seg.
	AP^{bb}	AP^{mk}	mIoU
SlotCon	40.8	36.8	71.7
SlotCon + Pix.	40.7	36.6	70.6
SlotCon - Coord. + Sim.	39.7	35.7	68.3
SlotCon - Coord. + Sim. + Img.	40.5	36.5	69.7
SlotCon + Sim.	40.5	36.6	69.5
SlotCon + Sim. + SR	40.7	36.7	70.5

B. More qualitative analyses

More visualizations of self-attention maps. In Figure 1, we show self-attention maps produced using different models' backbone features. The self-attention is calculated as the cosine similarities between a specific pixel's feature and all the features in the same image. Besides PixCon variants, we also compare with three previous methods, including MoCo-v2+ [2], SlotCon [5], and ORL [6].

Visualizations of matches with in-box queries but low matching similarities. When formulating the semantic reweighting strategy, we assume that matches with in-box queries, which lie in the intersected area of query and key views, are highly likely to own semantically consistent keys

Table 7. Transfer results from COCO+ pertaining. The results of SlotCon and PixCon-SR are reported as the averages of 5, 3, 3, 5, and 3 independent runs for VOC detection, COCO detection&instance segmentation, Cityscapes segmentation, VOC segmentation, and ADE20k segmentation, respectively. Except for PixCon-SR, all the methods are region-level methods. (†: re-prod. w/ official weights.)

Method	Dataset	VOC detection			COCO		City. Seg.	VOC Seg.	ADE20k
		AP	AP_{50}	AP_{75}	AP^{bb}	AP^{mk}	mIoU	mIoU	mIoU
ORL† [6]	COCO	55.8	82.1	62.3	40.2	36.4	75.4	70.7	-
UniVIP [3]		56.5	82.3	62.6	40.8	36.8	-	-	-
SlotCon† [5]		54.5	81.9	60.3	40.8	36.8	76.1	71.7	38.7
PixCon-SR (ours)		57.6	82.8	64.0	40.8	36.8	76.6	73.0	38.0
ORL [6]	COCO+	-	-	-	40.6	36.7	-	-	-
UniVIP [3]		58.2	83.3	65.2	41.1	37.1	-	-	-
SlotCon† [5]		57.0	83.0	63.4	**41.7**	**37.6**	76.6	**74.1**	**38.9**
PixCon-SR (ours)		**58.5**	**83.4**	65.2	41.2	37.1	**77.0**	73.9	38.8

regardless of the query-key similarities, as they are guaranteed to have semantic correspondences in the key view. In Figure 2, we visualize the correspondences between in-box query pixels and their matched key pixels. We can observe that even at an early stage of training, most of the in-box queries with low matching similarities still have semantically consistent key pixels. This validates our assumption that in-box queries tend to have semantically consistent keys regardless of their matching similarities. As the training goes further, the matches are also getting more accurate despite the magnitudes of similarities.

References

[1] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020. 1

[2] Junqiang Huang, Xiangwen Kong, and Xiangyu Zhang. Revisiting the critical factors of augmentation-invariant representation learning. In *ECCV*, 2022. 1, 3

[3] Zhaowen Li, Yousong Zhu, Fan Yang, Wei Li, Chaoyang Zhao, Yingying Chen, Zhiyang Chen, Jiahao Xie, Liwei Wu, Rui Zhao, et al. Univip: A unified framework for self-supervised visual pre-training. In *CVPR*, 2022. 4

[4] Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li. Dense contrastive learning for self-supervised visual pre-training. In *CVPR*, 2021. 1

[5] Xin Wen, Bingchen Zhao, Anlin Zheng, Xiangyu Zhang, and Xiaojuan Qi. Self-supervised visual representation learning with semantic grouping. *NeurIPS*, 2022. 1, 3, 4

[6] Jiahao Xie, Xiaohang Zhan, Ziwei Liu, Yew Soon Ong, and Chen Change Loy. Unsupervised object-level representation learning from scene images. *NeurIPS*, 2021. 3, 4

[7] Zhenda Xie, Yutong Lin, Zheng Zhang, Yue Cao, Stephen Lin, and Han Hu. Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning. In *CVPR*, 2021. 1

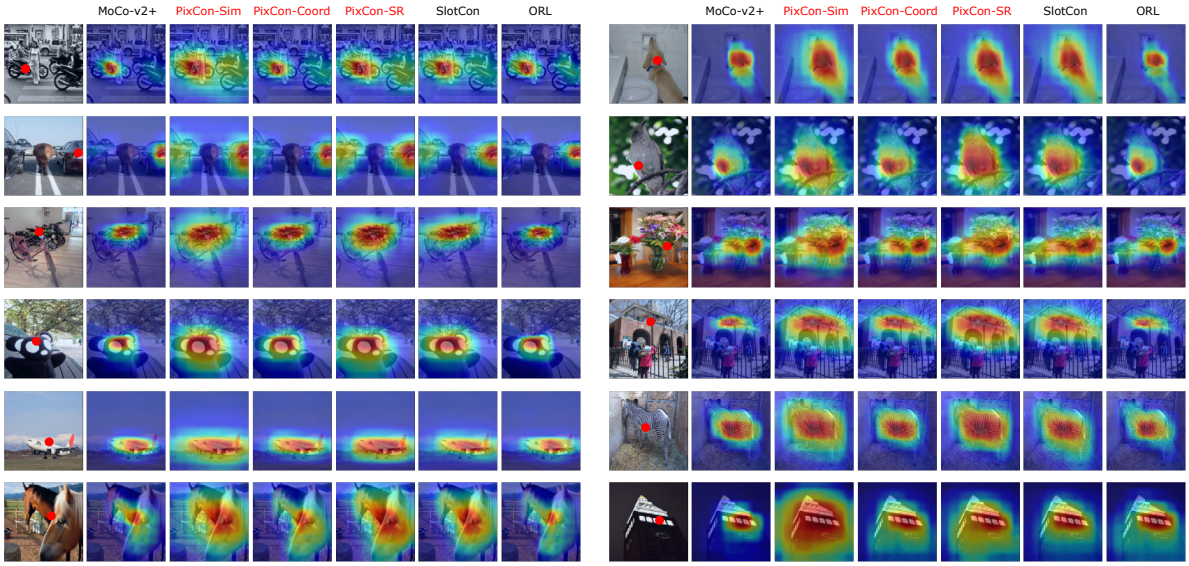

Figure 1. More visualization of self-attention maps.

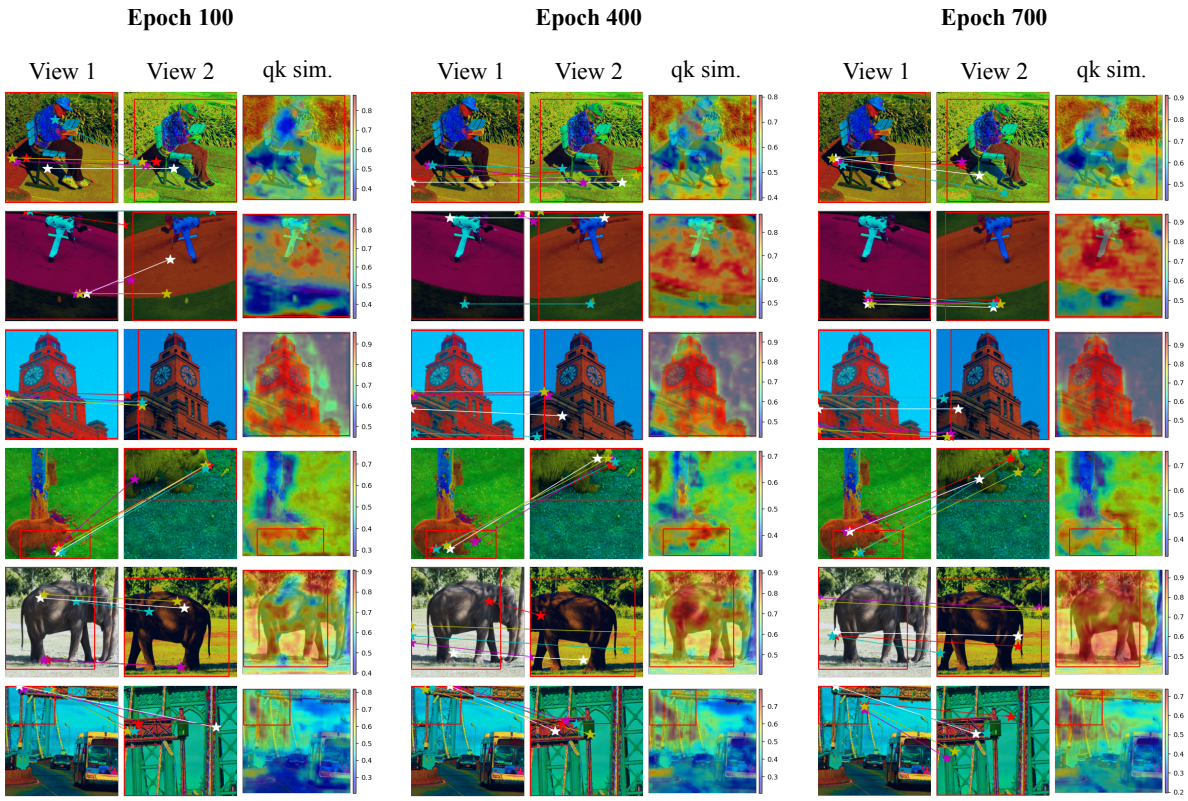

Figure 2. For each query view (view 1), we calculate the cosine similarities between its backbone features and those of the key view (view 2) at different training epochs. We keep five in-box query pixels that have the lowest similarities with their matched keys using similarity-based matching. The input images are randomly cropped and resized into 1024×1024, and then go through the other default data augmentations. The large input size is to more precisely visualize the correspondences. "qk sim." stands for the backbone feature similarities between the query and its matched key pixels and is only visualized for the query view.