Appendix

The supplemental material contains additional analysis, and ablation studies. All these are not included in the main paper due to the space limit.

A. More Results

A.1. Results with Longer Finetuning

Table 10 shows results with Mask-RCNN with FPN for 2× schedule on COCO dataset for ImageNet pretrained model. OURS improve over ReSim pretrained backbone by 1.6% AP on COCO detection and 1.4% AP on COCO segmentation.

Method	Pretrain epoch	Detection			Instance-seg.		
		AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{mk}	AP^{mk}_{50}	AP^{mk}_{75}
Scratch	-	36.7	56.7	40.0	33.7	53.8	35.9
Supervised	90	40.6	61.3	44.4	36.8	58.1	39.5
MoCo v2	200	40.9	61.5	44.6	37.0	58.4	39.6
ReSim-FPN^T	400	41.9	62.4	45.9	37.9	59.4	40.6
OURS	400	**43.5**	**63.7**	**47.5**	**39.3**	**61.3**	**42.1**

Table 10: **Object detection and instance segmentation on COCO for finetuning with 2× schedule.** We use Mask-RCNN with FPN for finetuning.

A.2. COCO Object and Instance Segmentation

In Table 11, we show more results of SSL methods on COCO object detection and instance segmentation using Mask-RCNN with FPN network for 90k iterations. Note that SoCo (4-views) [38] uses an additional views for pretraining to boost the performance.

A.3. VOC Object Detection

In Table 12, we show additional results for PASCAL VOC object detection using faster-RCNN with FPN network. We use the pretrained models released by the authors and finetune on the downstream task using Detectron2 framework.

A.4. COCO Keypoint Esitmation.

In Table 13, we show more results for COCO keypoint estimation. We use the pretrained models released by the authors and finetune on the downstream task using Detectron2 framework.

B. Analyzing Vision Transformer Backbone

We perform experiments using a vision transformer backbone to evaluate whether our method benefits transformer pretraining. We use DINO [2] self-supervised learning framework with Swin Transformer [28] backbone to evaluate our approach on vision transformer. We use Swin-T variant which has similar parameters as ResNet50, please refer to the original paper [28] for more details on the Swin Transformer backbone. We use AdamW optimizer with base learning rate of 0.0005 for batch-size 256 and weight-decay of 0.04. We pretrain the model for 300 epochs on 16 GPUs with 64 batch-size per GPU. For Swin with our approach, we set weight parameter $\alpha = 0.5$. We use similar image transformations as the other experiments in the main paper for pretraining. The backbones are trained on ImageNet training set. We tried to also pretrain the backbone on COCO dataset, however, the network did not converge. We use mmdetection framework [3] to train and evaluate on the downstream dataset with Swin transformer backbone. For faster-RCNN and mask-RCNN, we use ResNet50-FPN variant, train the models with AdamW optimizer for learning rate 0.0001 and weight-decay 0.05 on 8 GPUs with batch-size 2 per GPU. For segmentation, we use mmseg framework [9]. We use UperNet [40] for segmentation on VOC, CityScapes, and ScanNet. We use AdamW optimizer with learning rate 0.00006 and weight-decay 0.01. We train on PASCAL VOC for 20k iterations, and 40k iterations on the other datasets.

In Table 14, we report results for the PASCAL VOC, MS COCO, CityScapes, and ScanNet datasets. For PASCAL VOC detection, we use a Faster-RCNN framework with ResNet-50 FPN backbone, and for segmentation, we use the UperNet [40] framework. LC-loss improves detection performance by 1.6% AP and segmentation results by 2.1% mIoU. For COCO object

Method	Pretrain epoch	Detection			Instance-seg.		
		AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{mk}	AP^{mk}_{50}	AP^{mk}_{75}
Scratch	-	31.0	49.5	33.2	28.5	46.8	30.4
Supervised	90	38.9	59.6	42.7	35.4	56.5	38.1
MoCo	200	38.5	58.9	42.0	35.1	55.9	37.7
MoCo v2	200	40.4	60.2	44.2	36.4	57.2	38.9
InfoMin	200	40.6	60.6	44.6	36.7	57.7	39.4
BYOL	300	40.4	61.6	44.1	37.2	58.8	39.8
VADeR	200	39.2	59.7	42.7	35.6	56.7	38.2
ReSim-FPN^T	200	39.8	60.2	43.5	36.0	57.1	38.6
SoCo	400	43.0	63.3	47.1	38.2	60.2	41.0
SoCo (4-views)	400	**43.2**	**63.5**	**47.4**	**38.4**	**60.2**	**41.4**
$DetCon_S$	1000	41.8	-	-	37.4	-	-
$DetCon_B$	1000	42.7	-	-	38.2	-	-
DenseCL	200	40.3	59.9	44.3	36.4	57.0	39.2
DetCo	800	40.1	61.0	43.9	36.4	58.0	38.9
InsLoc	400	42.0	62.3	45.8	37.6	59.0	40.5
PixPro	400	41.4	61.6	45.4	-	-	-
OURS	200	42.0	62.5	46.3	37.9	59.5	40.8
OURS	400	42.5	62.9	46.7	38.3	60.0	41.1
OURS	600	**42.8**	**63.4**	**46.6**	**38.6**	**60.4**	**41.5**

Table 11: **Object detection and instance segmentation fine-tuned on COCO.** We use Mask R-CNN R50-FPN (1× schedule), and report bounding-box AP (AP^{bb}) and mask AP (AP^{mk}).

Method	Pretrain epoch	Detection		
		AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}
Supervised [21]	90	53.2	81.7	58.2
Moco v2 [6]	200	55.6	81.3	61.8
BYOL [17]	300	55.0	83.1	61.1
DetCo [42]	800	56.7	82.3	63.0
ReSim-FPN [41]	200	57.8	82.7	65.4
SCRL [34]	800	57.2	83.8	63.9
SoCo [38]	400	57.4	82.6	64.7
DenseCL [37]	200	56.6	81.8	62.9
PixPro [43]	400	58.7	82.9	65.9
OURS	400	**60.1**	**84.2**	**67.8**

Table 12: **Object detection on PASCAL VOC.** We use faster RCNN with FPN for VOC object detection. We finetune all the layers including the pretrained backbone.

detection and instance segmentation, we use mask-RCNN with a ResNet-50 FPN backbone. Our method outperforms the baseline by 0.8% in detection and 0.4% in segmentation. We perform segmentation on the CityScapes and ScanNet datasets using the UperNet framework. Our approach increases the mIoU by 0.1% for CityScapes and 1.0% for ScanNet. More technical details about the finetuning setup are in the supplementary material. We note that the improvement for transformer is not as impressive as for the ResNet-50 backbone, which suggests that the self-supervised vision transformer might already have more spatial information in its feature representation than ResNet-50. This also aligns with the conclusion from [2] that the vision transformer contains information related to scene layout in the features.

Method	Pretrain Epoch	AP	AP_{50}	AP_{75}
Supervised	90	65.7	87.2	71.5
Moco v2	200	65.9	86.9	71.6
BYOL	300	66.3	87.4	72.4
VADeR*	200	66.1	87.3	72.1
SCRL*	1000	66.5	87.8	72.3
ReSim-FPN	200	66.6	87.4	72.8
DenseCL	200	66.2	87.3	71.9
PixPro	400	66.6	**87.8**	72.8
OURS	400	**67.2**	87.4	**73.7**

Table 13: **COCO keypoint estimation.** We use the publicly available ImageNet-pretrained checkpoints released by the authors and finetune on the COCO dataset with Keypoint R50-FPN network for 90k iteration. (*) denotes scores from the original papers.

Method	PASCAL VOC				COCO						Cityscapes	ScanNet
	AP^b	AP^b_{50}	AP^b_{75}	mIoU	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}	mIoU	mIoU
DINO(Swin)	51.4	80.4	56.0	73.6	40.2	62.3	43.9	37.6	59.3	40.4	78.0	62.1
DINO(Swin) + LC-loss	53.0	81.0	57.8	75.7	41.0	62.9	44.8	38.0	59.9	40.9	78.1	63.1

Table 14: **Results for the Swin transformer backbone pretrained with the DINO framework.** The models are pretrained on the ImageNet1K dataset for 300 epochs. LC-loss loss improves the baseline consistently across datasets and tasks.

C. More Ablations

In Table 15, we show more ablation studies on learning rate, and momentum for the target network update during pre-training. The evaluation is performed with similar settings for VOC object detection with faster-RCNN-FPN and on COCO object detection with mask-RCNN-FPN in terms of average AP. We use image-size 160 for Table 15b, and image-size 224 for Table 15a.

lr	VOC	COCO
0.3	58.9	42.0
0.5	59.6	42.0
1.0	59.2	41.7

(a) **learning rate**.

Mom.	VOC	COCO
0.99	59.1	41.3
0.996	58.4	41.6
0.999	56.6	40.4

(b) **Momentum**.

Table 15: **Ablation studies**

D. Computational Complexity.

Our LC-loss has minimum overhead over the BYOL for the training time. To train our network for 200 epochs, it takes around 31 hour 30 min, whereas without the LC-loss it takes 31 hour 17 min. Hence, it is just 1% slower than BYOL. In terms of GFLOPS, our LC-loss has a minimum overhead of 8.54 GFLOPS, whereas BYOL has a minimum overhead of 8.29 GFLOPS.

E. More Results on Few-shot Image Classification.

Table 16 shows few-shot learning results for SUpervised, BYOL, PixPro and ours. We use our pre-trained models as fixed feature extractors, and perform 5-way 5-shot few-shot learning on 7 datasets from diverse domains using a logistic regression classifier. It reports the 5-shot top-1 accuracy for the 7 diverse datasets. Results show the the best ours achieves better image classification scores than PixPro. However, the best method for few-shot transfer learning learning is BYOL. Similar finding are also reported in [25].

Method	EuroSAT[22]	CropDisease[30]	ChestX[36]	ISIC[8]	Sketch[35]	DTD[7]	Omniglot[26]	Avg
Supervised	85.8	92.5	25.2	43.4	86.3	81.9	93.0	72.6
BYOL	88.3	93.7	26.5	42.3	86.8	83.5	94.7	73.7
PixPro	80.5	86.4	26.5	41.2	81.5	73.9	92.2	68.9
OURS	84.5	90.1	25.2	41.9	85.6	80.2	91.5	71.3

Table 16: **Few-shot learning results on downstream datasets.** The pre-trained models are used as fixed feature extractors We report top-1 accuracy for 5-way 5-shot averaged over 600 episodes. We use the publicly available pre-trained backbone as feature extractor for the few-shot evaluation.