Supplementary Materials of "MultiSiam: Self-supervised Multi-instance Siamese Representation Learning for Autonomous Driving"

Kai Chen[1] Lanqing Hong[2] Hang Xu[2] Zhenguo Li[2] Dit-Yan Yeung[1]
[1]Hong Kong University of Science and Technology [2]Huawei Noah's Ark Lab
kai.chen@connect.ust.hk {honglanqing, xu.hang, li.zhenguo}@huawei.com dyyeung@cse.ust.hk

A. Proof for Equation (12)

We prove Equation (12) similarly with Chen *et al.* [1]. All the terminologies are kept consistent with Section 3. We first consider the case when momentum $\tau = 0$ (*i.e.*, the parameters of online and target networks are always the same). By definition, for $\forall i \in [1, H], j \in [1, W]$, the predictor $q_\theta(\cdot)$ is expected to minimize:

$$\mathbb{E}_r \left[||q_\theta(R_{i,j}, \Delta C_{i,j}) - Kmeans(R'_{i,j})||_2^2 \right], \tag{1}$$

where r is a random variable representing the feature of a random view (*e.g.*, data augmentation) of image x. For the simplicity of analysis, we use the mean square root $|| \cdot ||_2^2$ here, which is equivalent to cosine distance after the vectors are l_2-normalized. Then the optimal solution to $q_\theta(\cdot)$ should satisfy:

$$q_\theta^{optimal}(R_{i,j}, \Delta C_{i,j}) = \mathbb{E}_r \left[Kmeans(R'_{i,j}) | \Delta C_{i,j}, R_{i,j} \right], \tag{2}$$

for any pixel of any image, which is equivalent to Equation (12). Note that, here r denotes a conditional distribution conditioned on the online feature map R and the offset map ΔC, instead of a uniform distribution in Chen *et al.* [1]. When $\tau \neq 0$, the target network is a exponential moving average of the online network, which also helps estimate the expectation in Equation (2), as suggested in Chen *et al.* [1].

B. More Implementation Details

B.1. MultiSiam without K-means

We further implement a *MultiSiam without K-means* in Section 4.2 by calculating a per-pixel cosine distance of 2D features with all other modules unchanged to verify the guidance effect of K-means. Specifically, the consistency loss between the online network's prediction Q and the aligned target feature R' is now defined as:

$$\mathcal{L}_{2D\_wo\_cluster} \triangleq \frac{1}{HW} \sum_{i=1}^{H} \sum_{j=1}^{W} -cos(Q_{i,j}, R'_{i,j}), \tag{3}$$

without adopting K-means clustering on the target network. All other modules including the IoU threshold, the feature alignment and the self-attention remain unchanged.

B.2. MoCo-based MultiSiam

Here we briefly introduce our simple implementation of MoCo-based *MultiSiam* with *RoI alignment*. All the terminologies are kept consistent with Section 3. Since MoCo does not adopt a separate predictor, feature alignment is now deployed before projectors instead of after as the BYOL-based *MultiSiam* does. Specifically, the aligned online and target feature maps R and R' are represented as:

$$R = RoIAlign(F, B), \tag{4}$$

$$R' = RoIAlign(F', B'), \tag{5}$$

which will be fed into the projectors to get the projected online and target feature maps G and G' as:

$$G_{i,j} = \sum_{i'=1}^{H} \sum_{j'=1}^{W} sim(R_{i,j}, R_{i',j'}) \cdot g_\theta(R_{i',j'}), \tag{6}$$

$$G' = g_\xi(R'). \tag{7}$$

Note that self-attention module is now incorporated into the online projector. K-means is then performed on G' to get the centroid of each pixel, denoted as $Kmeans(G'_{i,j})$, which will be considered as the positive sample for the pixel at the same relative position of the online feature map (*i.e.*, $G_{i,j}$). The final per-pixel contrastive loss $\mathcal{L}_{i,j}$ and the MoCo-based 2D clustering consistency loss $\mathcal{L}_{2D\_cluster}$ is defined as (for $\forall i \in [1, H], j \in [1, W]$):

$$\mathcal{L}_{2D\_cluster} \triangleq \frac{1}{HW} \sum_{i=1}^{H} \sum_{j=1}^{W} \mathcal{L}_{i,j}, \tag{8}$$

$$\mathcal{L}_{i,j} \triangleq \frac{exp(G_{i,j} \cdot k_{i,j}^{+}/\tau)}{exp(G_{i,j} \cdot k_{i,j}^{+}/\tau) + \Sigma_{k^-} exp(G_{i,j} \cdot k^{-}/\tau)}, \tag{9}$$

where $k_{i,j}^{+} = Kmeans(G'_{i,j})$ is the positive sample of $G_{i,j}$, while $\{k^-\}$ are negative samples maintained by a separate

Method	PASCAL VOC			COCO Mask R-CNN 90k (1x)						Cityscapes			BDD100K
	AP	AP$_{50}$	AP$_{75}$	AP^{bb}	AP$^{bb}_{50}$	AP$^{bb}_{75}$	AP^{mk}	AP$^{mk}_{50}$	AP$^{mk}_{75}$	AP	AP$_{50}$	mIOU	mIOU
Rand Init	33.8	60.2	33.1	31.0	49.5	33.2	28.5	46.8	30.4	25.4	51.1	65.3	50.7
Supervised	53.5	81.3	58.8	38.9	59.6	42.7	35.4	56.5	38.1	32.9	59.6	74.6	58.8
InstDist	55.2	80.9	61.2	37.4	57.6	40.6	34.1	54.6	36.4	33.0	60.1	73.3	57.2
SwAV	56.1	82.6	62.7	38.5	60.4	41.4	35.4	57.0	37.7	33.9	62.4	73.0	57.1
MoCo	55.9	81.5	62.6	38.5	58.9	42.0	35.1	55.9	37.7	32.3	59.3	75.3	59.7
MoCo-v2	57.0	82.4	63.6	38.9	59.4	42.4	35.5	56.5	38.1	33.9	60.8	75.7	60.0
DetCo	57.8	82.6	64.2	39.5	60.3	43.1	35.9	56.9	38.6	34.7	63.2	76.5	60.9
DenseCL	58.7	82.8	65.2	40.3	59.9	44.3	36.4	57.0	39.2	34.3	62.5	75.7	59.3
MultiSiam	57.8	**83.0**	65.0	**40.7**	**61.7**	**44.5**	**37.0**	**58.6**	**39.7**	**34.9**	**63.8**	**77.2**	**61.7**

Table A1. **Comparisons between methods pre-trained on ImageNet.** Although designed for multi-instance circumstances, our proposed *MultiSiam* still achieves state-of-the-art performance and demonstrates strong generalization ability to single-centric-object datasets.

Method	Mask R-CNN 180k (2x)						RetinaNet 90k (1x)			RetinaNet 180k (2x)			RetinaNet 12k		
	AP^{bb}	AP$^{bb}_{50}$	AP$^{bb}_{75}$	AP^{mk}	AP$^{mk}_{50}$	AP$^{mk}_{75}$	AP	AP$_{50}$	AP$_{75}$	AP	AP$_{50}$	AP$_{75}$	AP	AP$_{50}$	AP$_{75}$
Rand Init	36.7	56.7	40.0	33.7	53.8	35.9	24.5	39.0	25.7	32.2	49.4	34.2	4.0	7.9	3.5
Supervised	40.6	61.3	44.4	36.8	58.1	39.5	37.4	56.5	39.7	38.9	58.5	41.5	24.3	40.7	25.1
MoCo	40.8	61.6	44.7	36.9	58.4	39.7	36.3	55.0	39.0	38.7	57.9	41.5	20.2	33.9	20.8
MoCo-v2	40.9	61.5	44.7	37.0	58.7	39.8	37.2	56.2	39.6	39.3	58.9	42.1	22.2	36.9	23.0
DetCo	41.5	62.1	45.6	37.6	59.2	40.5	38.0	57.4	40.7	39.8	59.5	42.4	23.6	38.7	24.6
MultiSiam	**42.1**	**63.2**	**46.1**	**38.2**	**60.2**	**41.1**	**38.4**	**57.9**	**41.2**	**40.0**	**59.6**	**42.8**	**23.8**	**39.8**	24.5

Table A2. **Comparisons on COCO objection detection and instance segmentation.** All methods are pre-trained on ImageNet for 200 epochs. As we can see, *MultiSiam* outperforms all self-supervised counterparts in all downstream settings.

Method	Mask R-CNN 12k					
	AP^{bb}	AP$^{bb}_{50}$	AP$^{bb}_{75}$	AP^{mk}	AP$^{mk}_{50}$	AP$^{mk}_{75}$
Rand Init	10.7	20.7	9.9	10.3	19.3	9.6
Supervised	28.4	48.3	29.5	26.4	45.2	25.7
InstDist	24.2	41.5	25.1	22.8	38.9	23.7
SwAV	25.5	46.2	25.4	24.8	43.5	25.3
MoCo	25.6	43.4	26.6	23.9	40.8	24.8
MoCo-v2	26.6	44.9	27.7	24.8	42.1	25.7
DetCo	27.9	46.9	29.3	26.0	44.2	26.9
MultiSiam	**30.3**	**50.6**	**31.8**	**28.5**	**47.8**	**29.8**

Table A3. **Comparisons on COCO objection detection and instance segmentation by training 12k iterations.** Our *MultiSiam* exceeds all baseline methods with a larger margin compared to fine-tuning 90k and 180k iterations.

momentum queue containing features from different images following DenseCL [6]. τ is the temperature hyperparameter, which is set to be 0.2 by default.

C. Pre-training on ImageNet

Setup. We further pre-train our *MultiSiam* on the single-centric-object ImageNet dataset to verify its generalization ability. All hyper-parameters are kept the same with Waymo pre-trainings without specific tuning. We pre-train for 200 epochs and report the transfer performance on different downstream tasks. Note that here we cite the results of DetCo [8] without using Rand-Augmentation during pre-training for a fair comparison with other methods.

Downstream tasks. We choose six representative downstream tasks to evaluate the features following DetCo [8], including object detection on Pascal VOC [3], objection detection and instance segmentation on COCO [4], instance and semantic segmentation on Cityscapes [2] and semantic segmentation on BDD100K [9].

For VOC object detection, we train a Faster R-CNN (C4-backbone) on VOC `trainval07+12` set for 24k iterations and evaluate on VOC `test` set. We evaluate COCO object detection and instance segmentation by fine-tuning on COCO `train2017` set and test on COCO `val2017` set. Here we adopt both two-stage Mask R-CNN (FPN-backbone) and one stage RetinaNet with three training schedules, including standard 90k (1x), 180k (2x) in [7] and an extreme 12k-iteration schedule following DetCo for fast convergence, since it is possible to get competitive results on COCO even training from scratch but with enough iterations. For Cityscapes instance segmentation, we train a Mask R-CNN (FPN-backbone) for 24k iterations, while we fine-tune a FCN-16s for 90k iterations on both `train` sets and evaluate on the corresponding `val` sets for Cityscapes and BDD100K semantic segmentation.

Discussion. As shown in Table A1, A2 and A3, although originally designed for multi-instance circumstances, our *MultiSiam* still achieve state-of-the-art performance on all downstream benchmarks under different settings after pre-trained on ImageNet, revealing the generalization ability of

MultiSiam. The improvement is more significant when finetuning for 12k iterations compared with 90k and 180k settings, indicating that *MultiSiam* can effectively fasten model convergence. Also the robustness to hyper-parameters will decrease the deployment difficulty to other domain-specific circumstances like medical images.

D. More Ablation Studies

D.1. Ablations on Optimization

Minimum crop scale. During data augmentation, we will first select the scale of the output crop by randomly choosing a percentage of the original image scale between a minimum value and 100%, followed by random cropping and flipping. The results show that it's more beneficial to use a smaller minimum value to capture scale-invariance for downstream visual tasks.

Base learning rate. We find the optimal base learning rate for downstream dense visual tasks is much larger than that for image classification in BYOL, which demonstrates the differences between image-level visual tasks and pixel-level visual perception problems.

Base momentum. As we can see in Table A4(c), using a larger base momentum and a more stable target network will help increase the transfer performance of the learned visual representations.

D.2. Ablations on Self-attention Module

We verify whether to use the residual connections in the self-attention mechanism of *MultiSiam* as the original Non-local Network [5] in Table A5. *RoI alignment* works slightly better with residual connections, while *offset alignment* suffers from significant performance drop. We argue that using residual connections with *offset alignment* might hurt the prediction effectiveness under long range offset circumstances.

Minimum Crop Scale	Base Learning Rate	Base Momentum	mIOU
(a) Minimum Crop Scale			
0.08	0.3	0.99	**71.9**
0.2	0.3	0.99	71.7
(b) Base Learning Rate			
0.08	0.3	0.99	71.9
0.08	1.0	0.99	**72.9**
(c) Base Momentum			
0.2	0.1	0.99	71.6
0.2	0.1	0.996	**72.3**

Table A4. **Ablations on optimization hyper-parameters.** (a) Minimum crop scale; (b) base learning rate; (c) base momentum. All results are evaluated on Cityscapes *val* set over three independent trials, same as the main paper.

Feature Alignment	Residual Connection	mIOU
RoI		71.2
RoI	✓	71.4
Offset		**71.9**
Offset	✓	70.0

Table A5. **Ablations on residual connections in self-attention.** As we can see, residual connections work well with *RoI alignment* but bring performance drop for *offset alignment*.

References

[1] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. *arXiv:2011.10566*, 2020. 1

[2] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 2

[3] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. In *ICCV*, 2010. 2

[4] 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. 2

[5] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018. 3

[6] Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li. Dense contrastive learning for self-supervised visual pre-training. *arXiv:2011.09157*, 2020. 2

[7] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. 2019. 2

[8] Enze Xie, Jian Ding, Wenhai Wang, Xiaohang Zhan, Hang Xu, Zhenguo Li, and Ping Luo. Detco: Unsupervised contrastive learning for object detection. *arXiv:2102.04803*, 2021. 2

[9] Fisher Yu, Wenqi Xian, Yingying Chen, Fangchen Liu, Mike Liao, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving video database with scalable annotation tooling. *arXiv:1805.04687*, 2018. 2