Learning a Neural Association Network for Self-supervised Multi-Object Tracking Supplementary Material

Shuai Li[1]
lishuai@iai.uni-bonn.de
Michael Burke[2,3]
michael.g.burke@monash.edu
Subramanian Ramamoorthy[3]
s.ramamoorthy@ed.ac.uk
Juergen Gall[1,4]
gall@iai.uni-bonn.de

[1] University of Bonn
Bonn, Germany
[2] Monash University
Melbourne, Australia
[3] University of Edinburgh
Edinburgh, UK
[4] Lamarr Institute for Machine Learning and Artificial Intelligence
Germany

1 Implementation Details

Preprocessing Detections. During training, we use Faster RCNN [2] detections, and our training formulation attempts to track K objects in a single clip, where K can vary among clips. We threshold the detections based on the detection confidence values to remove potential false positives. The number of remaining detections in the first frame of each clip defines K. To compensate for missing detections, we use the KCF [3] tracker, initialized independently for each detection at the first frame of the clip. In case of missing detections at certain frames, we use the bounding box output of KCF. If this results in more than K detections in the next frame, we discard detections with a low intersection-over-union with tracked bounding boxes. This preprocessing operation enables us to collect $K \times T$ detections from a single video clip, where T is the clip length. While K can differ for each clip, we keep T the same for all clips. For the sake of computational efficiency, we generate the video clips with $T = 10$ from each training sequence. Overall, we generate 260 videos from the MOT17 training set to train the association network that is used for tracking on MOT17 and MOT20, the same preprocessing procedure is used to train matching network on BDD100K.
Training. Our implementation is based on the PyTorch framework. We use the detections' first frame coordinates to initialize mean μ_1 of the first state $\mathbf{x}_1$, and the covariance matrix Σ_1 is initialized as a diagonal matrix with variance set to 300. For the motion model $\mathbf{F}$ in the Kalman filter, a constant velocity model is utilized. The process noise σ_q in the diagonal covariance matrix $\mathbf{Q}$ is set to 150 and the observation noise σ_r in the diagonal covariance matrix $\mathbf{R}$ is set to 5. This enables the Kalman filter to rely more on the observation model during training. We have also experimented with updating the parameters of $\mathbf{Q}$ during training but the differences were marginal as long as $\mathbf{Q}$ is initialized to a large value. For each bounding box $\mathbf{z}$, we resize it to 224×224 and feed it into ϕ_θ, which is parametrized by ImageNet pretrained ResNet-50 [7], to obtain the appearance embedding $\phi_\theta(\mathbf{z})$, followed by

	$s_{\min}$	HOTA ↑	MOTA ↑	IDF1 ↑	IDSW ↓
$\kappa = 5$	0.7	61.2	64.0	68.8	712
	0.75	61.9	64.0	70.0	690
	0.8	62.2	64.1	70.3	668
	0.85	62.4	**64.1**	70.5	**652**
	0.9	**62.5**	64.1	**70.6**	656

	$s_{\min}$	HOTA ↑	MOTA ↑	IDF1 ↑	IDSW ↓
$\kappa = 10$	0.7	59.9	64.0	66.0	697
	0.75	60.8	64.0	67.6	669
	0.8	61.8	64.0	69.4	**647**
	0.85	**62.3**	**64.1**	**70.4**	649
	0.9	62.3	64.1	70.2	653

Table 1: Tracking performance under different combinations of hyperparameters on the MOT-17 training set.

	HOTA↑/IDF1↑/IDSW↓ $\sigma_{\text{vel}} = \frac{1}{320}$	HOTA↑/IDF1↑/IDSW↓ $\sigma_{\text{vel}} = \frac{1}{160}$	HOTA↑/IDF1↑/IDSW↓ $\sigma_{\text{vel}} = \frac{1}{80}$
$\sigma_{\text{pos}} = \frac{1}{40}$	62.0/69.9/731	62.0/69.8/725	61.4/69.0/783
$\sigma_{\text{pos}} = \frac{1}{20}$	62.0/69.9/727	62.0/69.9/731	62.0/69.8/725
$\sigma_{\text{pos}} = \frac{1}{10}$	61.8/69.7/773	62.0/69.9/727	62.0/69.8/731
$\sigma_{\text{pos}} = \frac{1}{5}$	61.4/69.2/834	61.8/69.7/773	62.0/69.9/727

Table 2: Impact of different parameters for the Kalman filter on the MOT17 training set with public detections. The results are reported without appearance model.

L_2 normalization. Adam [4] optimizer is used during training. We train $g_\theta(\cdot)$ with a learning rate of 5×10^{-3} for 10 epochs followed by fine-tuning ϕ_θ with a learning rate of 10^{-4} for another 3 epochs. Note that we only use the *unlabeled* detections from the MOT17 training set to fine-tune ϕ_θ.

2 Additional Ablation Studies

We study the influence of the parameters κ and $s_{\min}$ (11) for the tracking performance on the MOT17 training set. The results are shown in Table 1. The results show that our approach is not very sensitive to the parameters. As default values, we use $\kappa = 5$ and $s_{\min} = 0.85$.

3 Effect of Kalman Filter's hyper-parameters

During inference, we initialize the mean of each track as: $\mu_{\text{init}} = (x, y, w, h, 0, 0, 0, 0)$ where x, y, w, h denotes the bound-box's center and width/height. The initial covariance is dependent on the specific detection, *i.e.*, $\Sigma_{\text{init}} = \text{diag}((2\sigma_{\text{pos}}w)^2, (2\sigma_{\text{pos}}h)^2, (2\sigma_{\text{pos}}w)^2, (2\sigma_{\text{pos}}h)^2, (10\sigma_{\text{vel}}w)^2, (10\sigma_{\text{vel}}h)^2, (10\sigma_{\text{vel}}w)^2, (10\sigma_{\text{vel}}h)^2)$ where σ_{pos} and σ_{vel} are the variance for bounding box's position and velocity, respectively. For process covariance: $\mathbf{Q} = \text{diag}((\sigma_{\text{pos}}w)^2, (\sigma_{\text{pos}}h)^2, (\sigma_{\text{pos}}w)^2, (\sigma_{\text{pos}}h)^2, (\sigma_{\text{vel}}w)^2, (\sigma_{\text{vel}}h)^2, (\sigma_{\text{vel}}w)^2, (\sigma_{\text{vel}}h)^2)$ and for measurement noise: $\mathbf{R} = \text{diag}((\sigma_{\text{pos}}w)^2, (\sigma_{\text{pos}}h)^2, (\sigma_{\text{pos}}w)^2, (\sigma_{\text{pos}}h)^2)$.

We study the influence of these hyper-parameters on the performance on the MOT17 training set using public detections. We only use the motion affinity network during tracking. The results in Table 2 suggest that the tracking performance is not very sensitive to the choice of process and measurement noise. Therefore, we set $\sigma_{\text{pos}} = \frac{1}{20}$ and $\sigma_{\text{vel}} = \frac{1}{160}$ in the final model during inference. In particular, we use the same noise parameters of the Kalman filter for MOT17, MOT20 and BDD100K [10].

We also provide inference time of our approach in Table 3. Thanks to our online inference procedure, our approach is fast and can be used in real-time applications.

Association	MOT17	MOT20
mot	96	12
mot+app	33	6

Table 3: Inference speed (FPS) on different datasets.

Method	Sup.	HOTA ↑	MOTA ↑	IDF1 ↑	IDSW ↓
MOTR [11]	✓	57.8	73.4	68.6	2439
MeMOTR [8]	✓	58.8	72.8	71.5	1902
MOTRv2 [13]	✓	62.0	78.6	75.0	2619
UCSL [9]	✗	58.4	73.0	70.4	-
OUTrack [6]	✗	58.7	73.5	70.2	4122
ByteTrack [12]	✗	63.1	<u>80.3</u>	77.3	2196
U2MOT [5]	✗	<u>64.2</u>	79.9	<u>78.2</u>	<u>1506</u>
Lu *et al.* [4]	✗	**65.0**	**80.9**	**79.6**	1749
Ours	✗	62.1	76.7	75.7	**1092**

Table 4: Benchmark results on MOT17 test set using **private** detections, ✓ indicates fully-supervised and ✗ means self-supervised methods. The best and second best self-supervised performances are shown in bold and underlined numbers, respectively

4 Results on MOT17 using Private Detections

We also compare our approach with other works under the private detection protocol in Table 4. We use the same YOLOX detector as [12]. Results show that our tracker achieves competitive results with the state of the art [5, 7] and even outperforms several transformer-based methods [1, 11, 13] that require expensive identity-level supervision and have a much larger number of parameters.

References

[1] Ruopeng Gao and Limin Wang. Memotr: Long-term memory-augmented transformer for multi-object tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9901–9910, 2023.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[3] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. High-speed tracking with kernelized correlation filters. *IEEE transactions on pattern analysis and machine intelligence*, 37(3):583–596, 2014.

[4] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[5] Kai Liu, Sheng Jin, Zhihang Fu, Ze Chen, Rongxin Jiang, and Jieping Ye. Uncertainty-aware unsupervised multi-object tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9996–10005, 2023.

[6] Qiankun Liu, Dongdong Chen, Qi Chu, Lu Yuan, Bin Liu, Lei Zhang, and Nenghai Yu. Online multi-object tracking with unsupervised re-identification learning and occlusion estimation. *Neurocomputing*, 483:333–347, 2022.

[7] Zijia Lu, Bing Shuai, Yanbei Chen, Zhenlin Xu, and Davide Modolo. Self-supervised multi-object tracking with path consistency. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19016–19026, 2024.

[8] Sha Meng, Dian Shao, Jiacheng Guo, and Shan Gao. Tracking without label: Unsupervised multiple object tracking via contrastive similarity learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16264–16273, 2023.

[9] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*, 28:91–99, 2015.

[10] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2636–2645, 2020.

[11] Fangao Zeng, Bin Dong, Yuang Zhang, Tiancai Wang, Xiangyu Zhang, and Yichen Wei. Motr: End-to-end multiple-object tracking with transformer. In *European conference on computer vision*, pages 659–675. Springer, 2022.

[12] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. Bytetrack: Multi-object tracking by associating every detection box. In *European conference on computer vision*, pages 1–21. Springer, 2022.

[13] Yuang Zhang, Tiancai Wang, and Xiangyu Zhang. Motrv2: Bootstrapping end-to-end multi-object tracking by pretrained object detectors. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22056–22065, 2023.