Supplementary Materials

Kalun Ho[1,2,3], Amirhossein Kardoost[3], Franz-Josef Pfreundt[1,2], Janis Keuper[4], and Margret Keuper[3]

[1] Fraunhofer Center Machine Learning, Germany
[2] CC-HPC, Fraunhofer ITWM, Kaiserslautern, Germany
[3] Data and Web Science Group, University of Mannheim, Germany
[4] Institute for Machine Learning and Analytics, Offenburg University, Germany

1 Comparison with Supervised Trackers

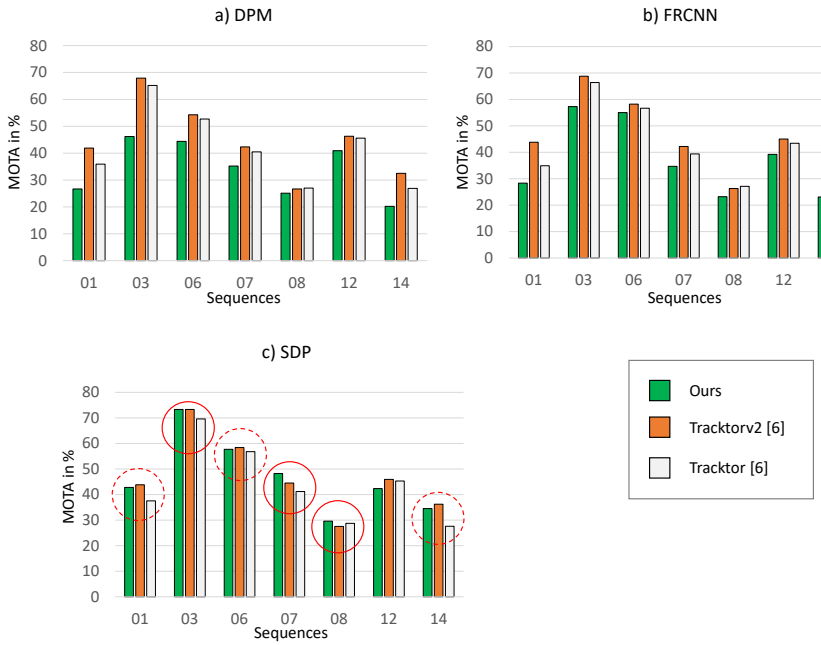

Fig. 1: Comparison with the tracker Tracktor[6] for each sequence grouped by detector. Red circle indicates that our proposed method outperforms both versions, while the dashed circle indicates that we are better than the originally published version from ICCV19.

We compare our proposed method with the tracker [6] on the MOT17 dataset, which has two versions in the MOT benchmark. Figure 1 shows the comparison of the per sequence evaluation grouped by detectors. The red circle indicates

that our proposed method outperforms both versions while the dashed circle shows that our method is better than its first version. Tracking-by-Detection is susceptible to the poor detector quality and Figure 1 shows that our proposed self-supervised approach is competitive with the tracker on the SDP detector. It struggles with the poor detections given by DPM and FRCNN. Note that, in contrast to supervised approaches, we directly use the provided bounding boxes and can not retrain nor refine the localization. Our best result is reached on the high quality SDP detector, where we are on par with the SotA tracker (46.9% vs. 47.1%, on average). In sequence 07 and 08 in the SDP detections, our proposed method outperforms the latest MOT17 submission of Tracktorv2 [6] while it reaches exactly the same score for the MOT17-03-SDP sequence.

2 Graph creation

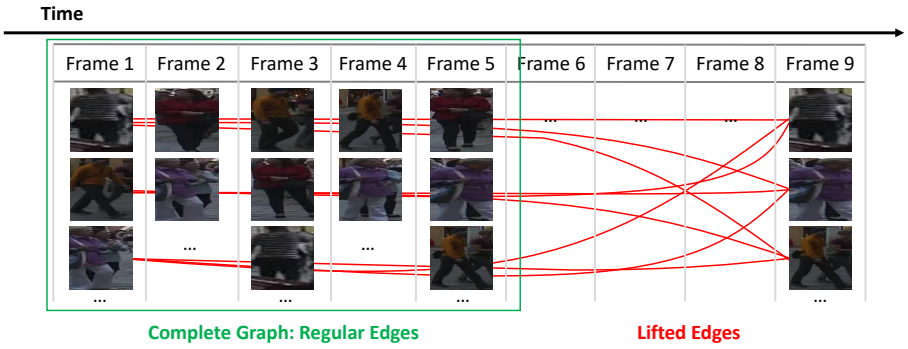

Fig. 2: Graph creation for minimum cost multicut problem. We insert two kind of edges: regular in green and lifted in red.

In this section, we briefly explain, how the graph is created in order to solve the minimum cost multicut problem. There are two kinds of edges: regular and lifted edges. Figure 2 shows an example of the graph for the temporal distance 1-5 at frame 1: within the selected distance window (e.g. distance 1-5), all detections at frame 1, 2, 3, 4 and 5 are nodes and a complete graph is built where regular edges are inserted. Lifted edges on the other hand are inserted at frame 8 in the example above. Table 2 of our main paper shows experiments with distance 1-3 and distance 1-5. Furthermore, we achieve the best result using lifted edges at range 10, 20 and 30 frames.

2.1 Experiment on Lifted Multicut

Here, we show an empirical study on the selection of the frame distance range of the inserted lifted edges in our proposed method. Adding lifted edges to the minimum cost multicut problem improves the performance of our model in almost all

metrics. The following study was conducted on a subset of the MOT17 Dataset, which contains in total 9 out of the training 21 sequences. As mentioned in our main paper, the lifted edges are inserted into the setup with temporal distances of 1-5 of regular edges.

Table 1 shows the different experiments and the resulting MOTA scores with the applied edge lifting setting. The first column (*No*) represents the experiment number while the frame distance for the lifted edges is indicated in the second column. A *frame distance*= 0 means no lifted edges are inserted (our baseline). Since the evaluation of our study is based on a subset of the dataset, the listed performances are different from the reported total MOTA score in the main paper, which is 49.8% (subset is 45.5%). Note that the normal, non-lifted edges are always present. In experiment number 25-27, of Table 1, we added lifted edges for multiple distances as follows: the *frame distance* of "$10-100, 10$" means that a lifted edges for each detection are added for the distance range of 10 to 100 is a stepsize of 10, e.g. for distance $10, 20, 30, ...100$. The best scores for each metric over all experiments are marked bold.

Oberservation. Although varying the long range edge radius has only very little effect on the MOTA, MOTP and MOTAL scores, the other metrics seem to benefit. One observation is the decrease in identity switches (IDs) compared to the baseline in all cases when introducing the lifted edges.

Combining Multiple Lifting Distances. Experiment 3 and 5 uses lifted edges for a single distance with ranges of 10 and 20 frames. The IDs are 458 and 460, respecitvely. The combination of both experiments is shown in Experiment 15 (*e.g. frame distance* for two distances 10 and 20), which further reduces the total IDs to 450. Other metrics such as FP or FN are also improved in the combined setup. Similar oberservations are also shown in Experiment 16-23.

Evaluation and Final Setup. The highest tracking accuracy (MOTA) is achieved in Experiment 23, 24 and 25 with a MOTA score of 45.8%. Table 2 compares these three best setups. The highest score for each metric is marked bold again. Although Experiment 23 seems to have the overall best score among the three setups, **we decided to choose setup of Experiment 24** due to the fact that the lowest IDs is achieved. Furthermore, we think that adding lifed edges for three distances is a good compromise and does not "over-engineer" the graph.

3 Visualization

In this section, we present additional visualizations similar to the ones reported in our main paper. It is divided into two parts. In 3.1, the nearest neighbors of a selected detection based on fixed frame distances are shown. The distance between a pair detection hereby is measured based on the Euclidean distance in the latent space. Two models are compared: 1) an ordinary AutoEncoder trained

Table 1: Empirical Study on Lifted Multicuts. Different frame distances are applied to evaluate the tracking performance for the sequence 05, 09 and 13 for the detectors SDP, DPM and FRCNN. The highest score in each metric category is marked bold.

No	Frame Dist	MOTA	MOTP	MOTAL	F1	Rcll	Prc	IDs	MT	ML	FP	FN
1	0	45.5	81.5	46.2	14.8	50.1	92.8	476	182	302	2,768	35,778
2	5	45.5	81.5	46.3	15.2	50.1	92.9	470	183	**299**	2,741	35,746
3	10	45.6	81.5	46.3	15.5	50.1	92.9	458	**186**	301	2,733	35,772
4	15	45.6	81.5	46.3	15.4	50.1	92.9	463	**186**	301	2,742	35,738
5	20	45.7	81.5	46.3	15.5	50.1	93.0	460	185	301	2,701	35,770
6	30	45.7	**81.6**	46.3	15.5	50.1	92.9	464	183	301	2,725	35,730
7	40	45.6	81.5	46.3	15.5	50.1	92.9	462	183	301	2,722	35,767
8	50	45.7	81.5	46.3	15.5	50.1	93.0	467	183	301	2,718	35,752
9	60	45.7	81.5	46.3	**15.7**	50.1	92.9	468	183	**299**	2,732	35,716
10	70	45.6	81.5	46.3	15.6	50.1	92.9	472	183	**299**	2,736	35,731
11	80	45.6	81.5	46.3	15.6	50.1	92.9	474	183	**299**	2,738	35,749
12	90	45.7	81.5	46.3	**15.7**	**50.2**	92.9	472	183	**299**	2,742	35,713
13	100	45.6	81.5	46.3	**15.7**	50.1	92.9	471	183	**299**	2,740	35,733
14	150	45.6	81.5	46.3	15.2	50.1	92.9	470	183	**299**	2,738	35,743
15	10+20	45.7	81.5	46.3	15.4	50.1	93.0	450	**186**	301	2,709	35,738
16	20+30	45.7	81.5	46.3	15.4	50.1	93.0	456	185	301	2,696	35,757
17	30+40	45.7	81.5	46.3	15.5	50.1	93.0	462	185	301	2,695	35,741
18	40+50	45.7	81.5	46.3	15.5	50.1	93.0	460	185	301	2,699	35,760
19	50+60	45.7	81.5	46.4	15.6	50.1	93.0	466	185	**299**	2,717	35,715
20	60+70	45.7	81.5	46.3	15.5	**50.2**	92.9	466	183	**299**	2,727	35,708
21	70+80	45.6	81.5	46.3	**15.7**	50.1	92.9	465	183	**299**	2,726	35,754
22	80+90	45.7	81.5	46.4	**15.7**	50.1	93.0	476	184	**299**	2,694	35,735
23	90+100	**45.8**	81.5	**46.5**	15.6	**50.2**	**93.2**	471	185	**299**	**2,637**	35,691
24	**10+20+30**	**45.8**	81.5	46.4	15.1	50.1	93.0	444	**186**	301	2,691	35,730
25	10 - 100, 10	**45.8**	**81.6**	46.4	15.2	**50.2**	93.0	452	**186**	301	2,715	35,682
26	5 - 30, 5	45.7	81.5	46.3	15.1	**50.2**	92.8	**442**	**186**	301	2,775	**35,663**
27	10 - 100, 10	45.6	**81.6**	46.3	15.1	50.1	92.9	456	**186**	301	2,758	35,724

with reconstruction loss only and 2) our proposed method with clustering loss. After including the clustering loss, the first (top 1) nearest neighbor detection is more likely to be on the correct person than before. Subsection 3.2 illustrates the latent space of our proposed trained AutoEncoder for selected sequences. Furthermore, the visualization also includes the predicted cluster labels and the achieved tracking performance on that sequence.

3.1 Nearest Neighbour

We compare two trained AutoEncoder models and show how the Multiple Object Tracking problem can benefit from our proposed method. We pick one bounding box and retrieve its nearest neighbor based on the latent space distance. The assumption is that similar object should be very close together in the latent

Table 2: List of the best setup in terms of tracking accuracy. Experiment 24 shows has the lowest number of IDs compared to other setups.

No	Frame Dist	MOTA	MOTP	MOTAL	F1	Rcll	Prc	IDs	MT	ML	FP	FN
23	90+100	**45.8**	81.5	**46.5**	**15.6**	**50.2**	**93.2**	471	185	**299**	**2,637**	35,691
24	**10+20+30**	**45.8**	81.5	46.4	15.1	50.1	93.0	**444**	**186**	301	2,691	35,730
25	10 - 100, 10	**45.8**	**81.6**	46.4	15.2	**50.2**	93.0	452	**186**	301	2,715	35,682

space, even over a long frame distance. Figure 3 illustrates such an example: the nearest neighbor of the very top left image of a) and b) for a distance of 40 frame is computed. We plotted the result for every 5 frames. a) represents the result with an AutoEncoder based on reconstruction loss only while b) is our proposed model. We can observe in a) that the nearest neighbor at frame 25 is a false positive with distance 5.53. The True Positive is located in the third row with a distance of 5.71. In b) we can see that this is "moved" to the first row. At the same time, the Euclidean distance is reduced to 5.15. A similar observation is shown in Figure 4 and Figure 5. Our proposed method often yields the same cluster ID when looking at the its nearest neighbor.

3.2 Latent Space Visualization

We are interested in how the cluster distribution of the results looks like in the latent space of our proposed model. To visualize this, we feed image data of the bounding boxes into the model and visualize the features using TSNE. The cluster IDs are illustrated in different colors which are obtained from solving the Lifted Multicut Problem. The visualization shows that one person may undergo a change in appearance over time since one color (label) may be "spread" over a large area in the latent space. Our idea is to reduce this spreading behavior by introducing an additional clustering loss term during training. Figure 6 shows the result of a the scene MOT17-04-SDP with a MOTA score of 75.9%. The clusters are very well separated. A moving person can be identified in the latent space when its appearance is changing slightly over time, given the assumption that the camera is static, which is the case in this MOT17-04. For instance the person with black jacket on top center of the example: the data are distributed in a "line-shape". Figure 7 illustrates an example of MOT17-10-SDP, which was recorded with a moving camera during night. The scene shown in Figure 8 uses the noisy detector DPM. The large colored cluster at the center or at the bottom left are a good examples for such noises detections. The same scene using a FRCNN detector is shown in Figure 9. This data is significantly less noisy. It allows the auto-encoder to learn the variation of individual persons better. These variations are shown in the strong "line"-shapes.

Fig. 3: MOT17-09-SDP: a) No Clustering loss: False Positive in Frame 25 and Frame 40. b) Our proposed method with clustering loss: Frame 25 is correctly assigned with our proposed setup: the detection moves from the third row to the first. The euclidean distance is reduced from 5.71 to 5.15, respectively. However, frame 40 still remains false.

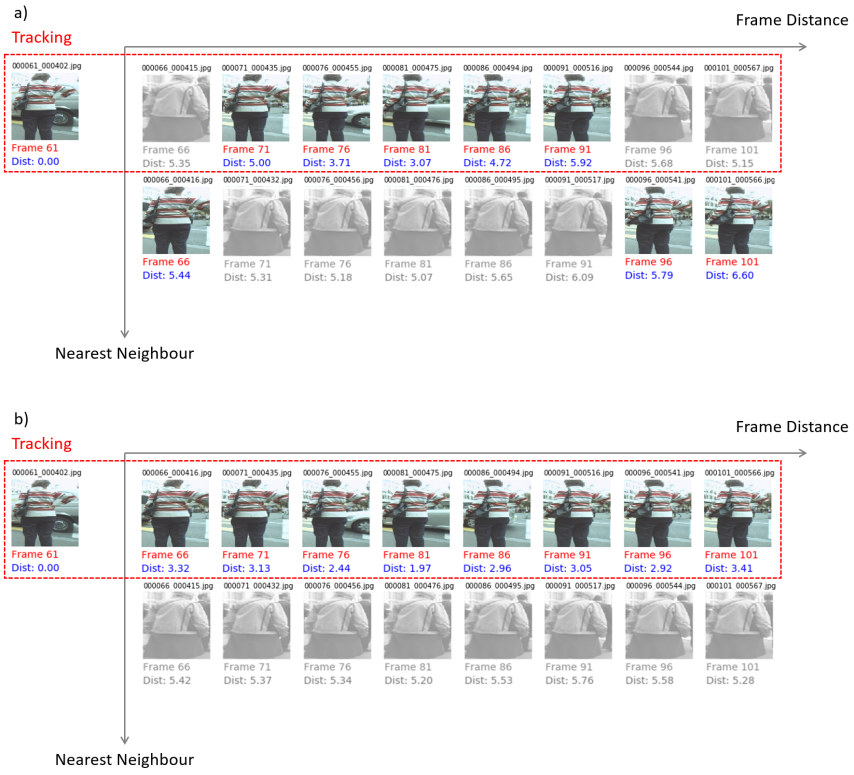

Fig. 4: MOT17-05-SDP: a) No Clustering loss: False Positive in Frame 66, 96 and 101. These appear in the second row, which represents the second nearest neighbour. The distances between the first and second row are very close (between 5.0 and 6.0). b) Our proposed method with clustering loss: All nearest neighbour is correctly assigned and the euclidean distance is reduced accordingly: the distances in the first row are significant lower than the second row.

Fig. 5: MOT17-13-SDP: a) No Clustering loss: False Positive in Frame 31. b) Our proposed method with clustering loss: this is corrected and the distance of the nearest neighbour is reduced from 3.62 (False Positive) to 3.19 (True Positive).

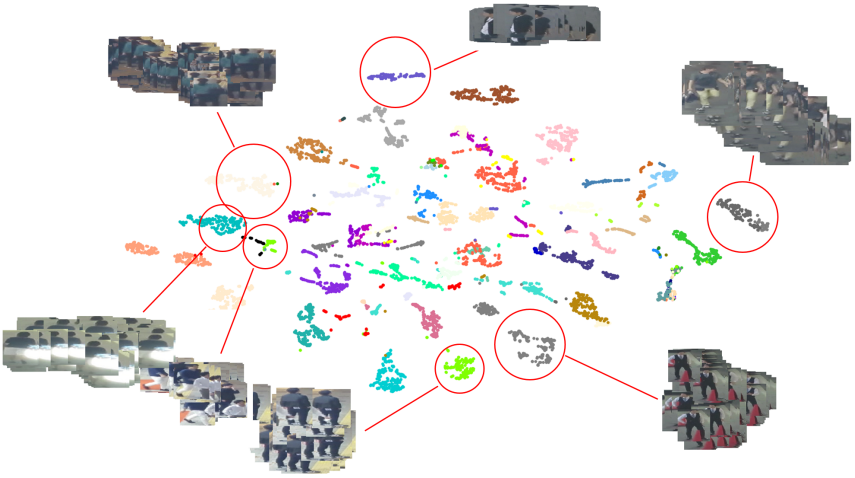

Fig. 6: MOT17-04-SDP: TSNE Visualization with MOTA score of 75.9%. Most clusters are well separated.

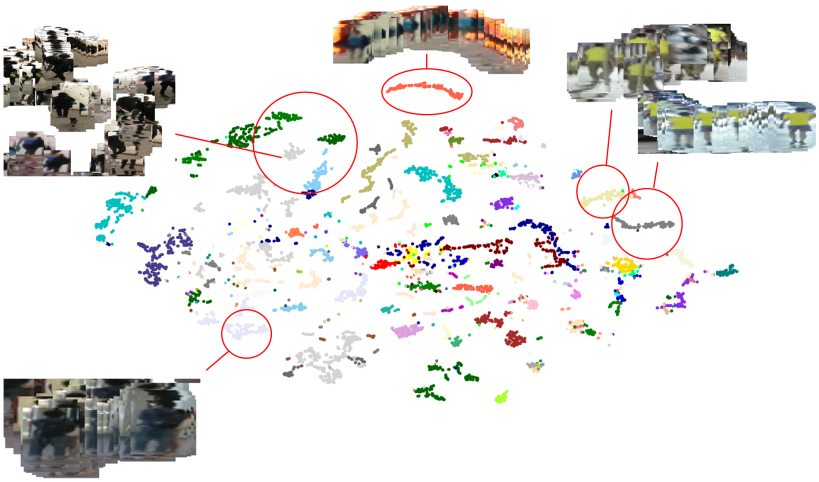

Fig. 7: MOT17-10-SDP: TSNE Visualization with MOTA score of 68.3%.

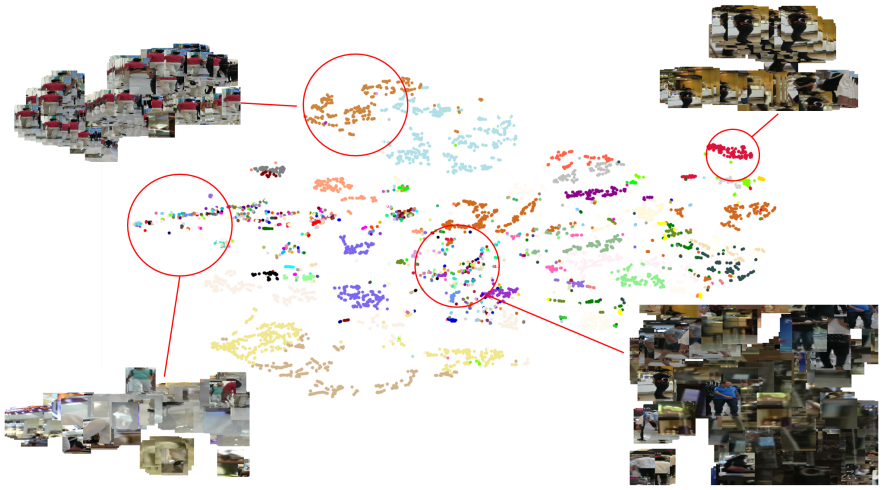

Fig. 8: MOT17-11-DPM: TSNE Visualization with MOTA score of 53.8%. There is a dense colored cloud at the center. This is also observed in the bottom left example. This is due to the noisy bounding boxes from the DPM-detector.

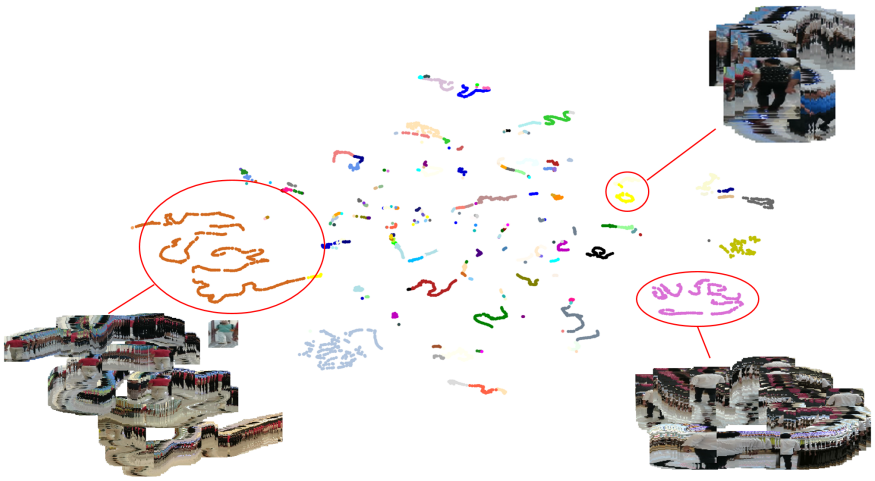

Fig. 9: MOT17-11-FRCNN: TSNE Visualization with MOTA score of 57.7%.