

**CVPR 2019 Submission #6191**  
**Eliminating Exposure Bias and Metric Mismatch**  
**in Multiple Object Tracking**  
Supplementary materials

**Abstract**

*This supplementary material contains this pdf, and a video explaining loss and training procedure. Tracking results are available at <https://cvlab.epfl.ch/research/research-surv/training-procedure/>. In this pdf, we present a detailed discussion of the ablation study that did not fit into the main text and give full benchmark results of our approach. This includes separate results for each sequence and a breakdown of our comparison to **SORT** on the **DukeMTMC** dataset. Finally, we provide additional details about our training protocol, and description of baselines.*

**1. Videos**

The accompanying video has 3 parts. The first shows tracklet examples. The second depicts valid tracklets, their true **IDF** score, and the one our network predicts. The third shows examples of input and output tracklets our network regresses by the means of *iou*, *lab*, and *sft*.

**2. Ablation study**

**Ablation Study.** It was performed on the validation data of **DukeMTMC** using the last 15000 training frames in each camera view.

Tab. 1 depicts the results organized in four columns.

- **Changes in dataset.** We quantify the impact of degrading the training dataset generation procedure of Section 3, by:

1. using random tracklets between all pairs of detections;
2. using tracklets obtained by combining at most two ground truth trajectories;
3. adding to the training data not all tracklets observed during growing phase, but only those present in the final solution;

4. doing pruning using predicted score of the tracklet as cutoff;
5. doing pruning by retaining fixed number of tracklets with best scores;

1), 2), and 3) yield a smaller and less diverse training data, which had a detrimental effect on the results of tracking. 4) did not allow us to train any reasonable model, because of the computational explosion of the trajectories with very similar scores, that were all taken into training data. 5) proved ineffective for the same reason - training data contained many very similar trajectories.

- **Changes in the loss function.** We modify the loss function, described in Section 4.1 by:

1. using loss of  $\|IDF(D, T) - S(\Phi(D))\|_2$ ;
2. regressing the value of *IDF* directly, without splitting the task into accounting for false positives or false negatives;
3. not modifying the input detections based on the regression of bounding box shifts;
4. removing  $L_{sft}$  component from the loss function;
5. posing task as a classification task, where tracklet belongs to the positive class *iff* all detections overlap with some ground truth trajectory with *IoU* of at least 0.5.

1) resulted in small decrease, probably due to the fact that multiple loss components acted as regularizers. 2) gave even worse results, because understanding the behaviour of *IDF* function is much harder than understanding behaviour of false positives and false negatives, which we regress through *lab* and *iou*. Difference between 3) and 4) shows that simply having  $L_{sft}$  as part of the loss function improves the results, acting as a regularizer. 5) doesn't result in a very good trained model due to many overlapping sequences, some of

#	$\Delta$ Dataset	IDF	$\Delta$ Loss	IDF	$\Delta$ Training	IDF	$\Delta$ Tracking	IDF
1	Dataset: all pairs	71.5	Loss on IDF	69.5	-hardmining	72.1	Batch 6	72.6
2	Dataset: mix of two	70.7	Regressing IDF	63.4	-balanced dataset	69.9	IP solution	73.8
3	Selected only	63.6	-bbox regression	72.4	pretraining	71.9		
4	Prunning by score	—	-bbox loss	66.3	2 layer LSTM	74.1		
5	Prunning by count	54.2	classification	41.8				

Table 1. Ablation study. Left, middle and right columns show possible changes in dataset creation procedure, loss function, training and tracking procedure, as well as respective values of **IDF** metric with respect to reference solution (**IDF** 74.6). Details about each change are given in Sec. 2.

which have *IoU* greater than 0.5 in every frame, and some don’t, and it is hard for the model to distinguish between the two.

- **Changes in the training procedure.** We modify the training procedure of Section 4.2 by:

1. not using hard-mining;
2. not balancing the dataset;
3. pre-training embedding for appearance and geometric features separately;
4. using 2 layer LSTM, instead of a single layer, as depicted in Fig.2, (a);

- **Changes in the tracking procedure.** We modify the tracking procedure of Section 3 by:

1. using shorter batch in tracking (3s, same as during training, instead of 6s);
2. selecting final solution by an IP trying to maximize objective of Eq. 1, rather than adding trajectories one by one greedily;

Additionally, while it may seem logical to use  $L_{iou}$  to predict the IoU between the modified bounding box  $\mathbf{d}_n + sft_t$  and the ground truth bounding box  $\mathbf{g}_n$ , in practice that makes it harder to train the network as it finds an easy solution of regressing empty bounding boxes, which never intersect with the ground truth, thus always making a perfect prediction of  $L_{iou}$ . Instead, we use the network during inference in the autocontext mode: we predict the bounding boxes, update the input tracklet with them, and then regress the intersection over union of the new tracklet to compute the value of  $S$ .

### 3. Detailed Benchmark Results

Here we now give a description of tracking metrics in Tab. 2 and full results for all benchmarks in Tab. 3, 4, 5. Legend information and results for **MOT15** dataset were collected from the benchmark website <https://motchallenge.net/> on the 6th of May, 2018, while

results for **MOT17** and **DukeMTMC** datasets were collected on the 30th of October, 2018. Our tracker results are available there under the names **SAS** and **SAS\_full** for **DukeMTMC** benchmark, **SAS\_MOT15** for **MOT15** benchmark, and **SAS\_MOT17** for **MOT17** benchmark.

We also report results of our comparison to **SORT** on the validation data we used for DukeMTMC dataset in Tab. 6. We tuned the parameters of the method (*max\_age*, *min\_hits*, detection quality cutoff) on the same data we used for training for ablation study, using grid search.

## 4. Training Protocol

We have trained the model with Adam with the fixed learning rate of 0.001. Our embedding layer consists of a fully connected layer, followed by a batch normalization layer. Size of the hidden state of LSTM were 300. In all cases we kept  $C_{iou} = 0.6$ ,  $C_{score} = 0.6$  and trained with batches of length 3s. Thanks to abundance of training data, we used fps of 3 for **DukeMTMC** dataset. For **MOT15** and **MOT17** datasets, we trained the model with the maximum frequency every sequence allowed, to increase the amount of training data. During inference, we used batches of length 6s. We used the bounding box shift regression only in combination with the DPM [4] detector, as for other types of detectors it did not prove useful. Nevertheless, we kept  $L_{sft}$  as a part of a loss function. We plan to make our implementation (in Python and using Tensorflow) publicly available upon acceptance of the paper.

For each **MOT15** sequence group (KITTI, ADL, etc.), we trained on all sequences excluding the group, using them for validation purposes, and ran inference on the test sequences from the same group. For **MOT17**, we used **PathTrack** for pre-training of the model, and training sequences for validation. We trained re-identification network on **CUHK03** dataset.

The coefficient, which we used to multiply the probabilities before softmax to allows the probabilistic merging, described in the paper, was annealed from 10 to 0.1 in 30 epochs.

Measure	Better	Perfect	Description
MOTA	higher	100	Multiple Object Tracking Accuracy [1]. This measure combines three error sources: false positives, missed targets and identity switches.
MOTP	higher	100	Multiple Object Tracking Precision [1]. The misalignment between the annotated and the predicted bounding boxes.
IDF1	higher	100	IDF [14]. The ratio of correctly identified detections over the average number of ground-truth and computed detections.
FAF	lower	0	The average number of false alarms per frame.
MT	higher	100	Mostly tracked targets. The ratio of ground-truth trajectories that are covered by a track hypothesis for at least 80% of their respective life span.
ML	lower	0	Mostly lost targets. The ratio of ground-truth trajectories that are covered by a track hypothesis for at most 20% of their respective life span.
FP	lower	0	The total number of false positives.
FN	lower	0	The total number of false negatives (missed targets).
ID Sw.	lower	0	The total number of identity switches.
Frag.	lower	0	The total number of times a trajectory is fragmented (i.e. interrupted during tracking).
Hz	higher	Inf.	Processing speed (in frames per second excluding the detector) on the benchmark.

Table 2. Metrics description.

Method	IDF1	IDP	IDR	MOTA	MOTP	FAF	MT	ML	FP	FN	ID Sw.	Frag.
<b>OURS</b>	84.0	89.4	79.2	76.0	76.0	0.09	950	72	66,783	186,974	169	1256
<b>MHT</b>	80.3	87.3	74.4	78.3	78.4	0.05	914	72	35,580	193,253	406	1,116
<b>REID</b>	79.2	89.9	70.7	68.8	77.9	0.07	726	143	52,408	277,762	449	1,060
<b>CDSC</b>	77.0	87.6	68.6	70.9	75.8	0.05	740	110	38,655	268,398	693	4,717
<b>OURS-geom</b>	76.5	83.9	70.3	69.3	74.8	0.10	813	89	76,059	248,224	426	2,081
<b>PTRACK</b>	71.2	84.8	61.4	59.3	78.7	0.09	666	234	68,634	361,589	290	783
<b>BIPCC</b>	70.1	83.6	60.4	59.4	78.7	0.09	665	234	68,147	361,672	300	801

Table 3. Full benchmark results on Easy set of sequences of **DukeMTMC** dataset.

Method	IDF1	IDP	IDR	MOTA	MOTP	FAF	MT	ML	FP	FN	ID Sw.	Frag.
<b>OURS</b>	76.8	89.3	67.4	65.4	75.3	0.12	450	87	35,596	210,639	267	977
<b>REID</b>	71.6	85.3	61.7	60.9	76.8	0.14	375	104	40,732	237,974	572	993
<b>OURS-geom</b>	65.5	79.3	55.8	59.1	74.0	0.14	379	102	39,576	251,256	972	1,855
<b>CDSC</b>	65.5	81.4	54.7	59.6	75.4	0.09	348	99	26,643	260,073	1,637	5,024
<b>PTRACK</b>	65.0	81.8	54.0	54.4	77.1	0.14	335	104	40,978	283,704	661	1,054
<b>BIPCC</b>	64.5	81.2	53.5	54.6	77.1	0.14	338	103	39,599	283,376	652	1,073
<b>MHT</b>	63.5	73.9	55.6	59.6	76.7	0.19	400	80	55,038	231,527	1,468	1,801

Table 4. Full benchmark results on Hard set of sequences of **DukeMTMC** dataset.

Method	MOTA	IDF1	MT	ML	FP	FN	ID Sw.	Frag.	Hz	Hardware
<b>OURS-geom</b>	22.2	27.2	3.1	61.6	5,591	41,531	700	1,240	8.9	2.5 GHz CPU
<b>SORT</b>	21.7	26.8	3.7	49.1	8,422	38,454	1,231	2,005	1,112.1	1.8 GHz CPU
<b>LP2D</b>	19.8	—	6.7	41.2	11,580	36,045	1,649	1,712	112.1	2.6Hz 16 CPU
<b>RNN</b>	19.0	17.1	5.5	45.6	11,578	36,706	1,490	2,081	165.2	3GHz, CPU

Table 5. Full benchmark results on **MOT15** dataset.

Method	Cam1	Cam2	Cam3	Cam4	Cam5	Cam6	Cam7	Cam8	Overall
<b>OURS</b>	82.1/76.8	70.9/67.7	88.5/84.0	70.9/63.0	58.9/49.9	88.5/81.6	71.4/71.6	66.0/66.4	74.6/70.1
<b>SORT</b>	23.7/37.8	27.7/51.2	26.5/53.0	25.7/40.5	28.6/68.0	23.0/54.9	27.6/56.8	16.4/37.0	24.9/49.9

Table 6. Comparison to **SORT** method on the validation data for DukeMTMC dataset, **IDF/MOTA**.

Method	MOTA	IDF1	MT%	ML%	FP	FN	IDs	Frag	Hz
<b>Ours</b>	44.2	57.2	16.1	44.3	29,473	283,611	1,529	2,644	4.8
<b>DMAN</b>	48.2	55.7	19.3	38.3	26,218	263,608	2,194	5,378	0.3
<b>JCC</b>	51.2	54.5	20.9	37.0	25,937	247,822	1,802	2,984	1.8
<b>MOTDT17</b>	50.9	52.7	17.5	35.7	24,069	250,768	2,474	5,317	18.3
<b>MHTBLSTM</b>	47.5	51.9	18.2	41.7	25,981	268,042	2,069	3,124	1.9
<b>EDMT17</b>	50.0	51.3	21.6	36.3	32,279	247,297	2,264	3,260	0.6
<b>FWT</b>	51.3	47.6	21.4	35.2	24,101	247,921	2,648	4,279	0.2

Table 7. Full benchmark results on **MOT17** dataset.

## 5. Baselines

Here we describe in more details the baselines we compare to in the paper.

### Algorithms that ignore Appearance Cues.

- **LP2D** [9] is the highest-scoring appearance-less original baseline presented with MOT15. It formulates tracking in terms of solving a linear program.
- **RNN** [12] relies on a recurrent neural network and is similar to ours in spirit because it uses RNN for tracking in a straightforward way. However it is trained using a different loss and approach to create the training data.
- **PTRACK** [11] aims to improve results of other methods by refining the trajectories they produce, to maximize an approximation of the **IDF** metric. The approximation is hand-designed, and not learned as in our approach.
- **SORT** [2, 13] combines Kalman filtering with a Hungarian algorithm and currently is the fastest one on the **MOT15** dataset.

### Algorithms that exploit Appearance Cues.

- **MHT** [17] performs multiple hypothesis tracking, aided, among others, by pose features extracted from convolution pose machines [16].
- **CDSC** [15] uses domination set clustering to perform within- and across-camera tracking. It employs image features from ResNet-50 [5] pre-trained on ImageNet.
- **REID** [19] performs hierarchical clustering of tracklets, and leverages the re-identification model of [18] pre-trained on 7 different datasets.
- **BIPCC** [14] clusters detections with similar appearance by solving a binary integer problem. This is a baseline method for the **DukeMTMC** dataset.
- **DMAN** [20] uses dual attention networks to perform data association by focusing on relevant image parts and temporal fragments.
- **JCC** [7] handles multiple object tracking and motion segmentation as a joint co-clustering problem. It solves it by local search to group pixels and bounding boxes. This returns both tracks and segmentation.

- **MOTDT17** [10] performs hierarchical data association by grouping detections using a learned re-identification metric, exploiting geometric features, and Kalman filter.
- **MHTBLSTM** [8] resembles our approach in spirit. It uses a multiple hypothesis tracker and a sequence model to score the tracks. However, it is trained using only ground-truth sequence with at most one false positive and sometimes missed detections.
- **EDMT17** [3] relies on a multiple hypothesis tracker. Its growing and pruning phase depend on learned detection-detection and detection-scene association models that are used to better score detections and hypotheses.
- **FWT** [6] solves a binary quadratic problem to optimally group head and body detections, obtained separately.

## References

- [1] K. Bernardin and R. Stiefelhausen. Evaluating Multiple Object Tracking Performance: the Clear Mot Metrics. *EURASIP Journal on Image and Video Processing*, 2008, 2008.
- [2] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft. Simple Online and Realtime Tracking. In *International Conference on Image Processing*, 2016.
- [3] J. Chen, H. Sheng, Y. Zhang, and Z. Xiong. Enhancing Detection Model for Multiple Hypothesis Tracking. In *Conference on Computer Vision and Pattern Recognition*, 2017.
- [4] P. Felzenszwalb, D. Mcallester, and D. Ramanan. A Discriminatively Trained, Multiscale, Deformable Part Model. In *Conference on Computer Vision and Pattern Recognition*, June 2008.
- [5] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.
- [6] R. Henschel, L. Leal-Taixé, D. Cremers, and B. Rosenhahn. Fusion of Head and Full-Body Detectors for Multi-Object Tracking. In *Conference on Computer Vision and Pattern Recognition*, 2018.
- [7] M. Keuper, S. Tang, B. Andres, T. Brox, and B. Schiele. Motion Segmentation and Multiple Object Tracking by Correlation Co-Clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

- [8] C. Kim, F. Li, and J. M. Rehg. Multi-Object Tracking with Neural Gating Using Bilinear LSTM. In *European Conference on Computer Vision*, 2018.
- [9] L. Leal-taixe, A. Milan, I. Reid, S. Roth, and K. Schindler. Motchallenge 2015: Towards a Benchmark for Multi-Target Tracking. In *ARXIV*, 2015.
- [10] C. Long, A. Haizhou, Z. Zijie, and S. Chong. Real-Time Multiple People Tracking with Deeply Learned Candidate Selection and Person Re-Identification. In *International Conference on Multimedia and Expo*, 2018.
- [11] A. Maksai, X. Wang, F. Fleuret, and P. Fua. Globally Consistent Multi-People Tracking Using Motion Patterns. In *International Conference on Computer Vision*, 2017.
- [12] A. Milan, S.H. Rezatofighi, A. Dick, I. Reid, and K. Schindler. Online Multi-Target Tracking Using Recurrent Neural Networks. In *AAAI*, 2017.
- [13] S. Murray. Real-Time Multiple Object Tracking-A Study on the Importance of Speed. *arXiv preprint arXiv:1709.03572*, 2017.
- [14] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi. Performance Measures and a Data Set for Multi-Target, Multi-Camera Tracking. In *European Conference on Computer Vision*, 2016.
- [15] Y. T. Tesfaye, E. Zemene, A. Prati, M. Pelillo, and M. Shah. Multi-Target Tracking in Multiple Non-Overlapping Cameras Using Constrained Dominant Sets. *arXiv preprint arXiv:1706.06196*, 2017.
- [16] S. E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional Pose Machines. In *Conference on Computer Vision and Pattern Recognition*, 2016.
- [17] K. Yoon, Y. Song, and M. Jeon. Multiple Hypothesis Tracking Algorithm for Multi-Target Multi-Camera Tracking with Disjoint Views. *IET Image Processing*, 2018.
- [18] X. Zhang, H. Luo, X. Fan, W. Xiang, Y. Sun, Q. Xiao, W. Jiang, C. Zhang, and J. Sun. Alignedreid: Surpassing Human-Level Performance in Person Re-Identification. *arXiv preprint arXiv:1711.08184*, 2017.
- [19] Z. Zhang, J. Wu, lx. Zhang, and C. Zhang. Multi-Target, Multi-Camera Tracking by Hierarchical Clustering: Recent Progress on DukeMTMC Project. *arXiv preprint arXiv:1712.09531*, 2017.
- [20] J. Zhu, H. Yang, N. Liu, M. Kim, W. Zhang, and M. Yang. Online Multi-Object Tracking with Dual Matching Attention Networks. In *European Conference on Computer Vision*, 2018.