Supplementary Material: Quasi-Dense Similarity Learning for Multiple Object Tracking

Jiangmiao Pang[1] Linlu Qiu[2] Xia Li[3] Haofeng Chen[4] Qi Li[1] Trevor Darrell[5] Fisher Yu[3]

[1]Zhejiang University [2]Georgia Institute of Technology [3]ETH Zürich
[4]Stanford University [5]UC Berkeley

In this supplementary material, we present detailed configuration of the tracker, additional experiments and ablation studies. We also investigate oracle performance, analyze failure cases, and show some patch visualizations.

A. Hyper-parameters

We show the configuration of our tracker in Algorithm 1. Some of the parameters, such as number of frames to keep backdrops and the matching metric, are fixed. For more details, please refer to our released source code.

Algorithm 1 Configuration of the tracker in QDTrack.

```
tracker=dict(
    type='QuasiDenseEmbedTracker',
    # score threshold to start a new track
    init_score_thr=0.8,
    # score threshold to continue a track
    obj_score_thr=0.5,
    # score threshold for data association
    match_score_thr=0.5,
    # number of frames to keep tracks
    memo_tracklet_frames=10,
    # number of frames to keep backdrops
    memo_backdrop_frames=1,
    # momentum to update the embeddings
    memo_momentum=0.8,
    # duplicate removal to tackle multi-targets cases
    nms_backdrop_iou_thr=0.3,
    nms_class_iou_thr=0.7,
    # the matching metric
    match_metric='bisoftmax')
```

Dataset specific parameters. Our object association only relies on appearance, so it is robust to different motion patterns in different datasets. The experiments share the same tracking parameters except TAO, because TAO uses 3D mAP, instead of CLEAR MOT metrics, for evaluation.

On TAO, the terms "init_score_thr" and "obj_score_thr" are set to 0.0001 to obtain a high recall. Considering the numerous tracks with these thresholds, we do not maintain backdrops in these experiments.

B. Supplementary experiments

MOT17 with public detectors Following the strategy in Tracktor [1] and CenterTrack [7], we evaluate our method with public detectors on MOT17. That is, a new trajectory is only initialized from a public detection bounding box. As shown in Table 1, our method outperforms existing results by a large margin. Our method outperforms CenterTrack by 3.1 points on MOTA and 5.5 points on IDF1.

TAO Table 2 presents detailed results on the TAO [4] dataset. Although QDTrack does not perform zero-shot and few-shot learning for the long-tail categories, our method is still a stronger baseline method on this dataset and paves the way for future studies.

BDD100K Segmentation Tracking The results on the BDD100K segmentation tracking validation set are presented in Table 3.

Table 1: Results on MOT17 test set with public detector. Note that we do not use extra data for training. ↑ means higher is better, ↓ means lower is better. * means external data besides COCO and ImageNet is used.

Dataset	Method	MOTA ↑	IDF1 ↑	MOTP ↑	MT ↑	ML ↓	FP ↓	FN ↓	IDs ↓
Public MOT17	Tracktor++v2 [1]	56.3	55.1	78.8	498 (21.1)	831 (35.3)	**8866**	235449	1987 (34.1)
	GSM_Tracktor [6]	56.4	57.8	77.9	523 (22.2)	813 (34.5)	14379	230174	1485 (25.1)
	MPNTrack* [3]	58.8	61.7	78.6	679 (28.8)	788 (33.5)	17413	213594	**1185 (19.1)**
	Lif_T* [5]	60.5	65.6	78.3	637 (27.0)	791 (33.6)	14966	206619	1189 (18.8)
	CenterTrackPub* [7]	61.5	59.6	78.9	621 (26.4)	752 (31.9)	14076	200672	2583 (40.1)
	Ours	**64.6**	**65.1**	**79.6**	**761 (32.3)**	**666 (28.3)**	14103	**182998**	2652 (39.3)

Table 2: Results on TAO challenge benchmark.

Method	Split	AP50	AP75	AP	AP50(S)	AP50(M)	AP50(L)
SORT_TAO [4]	*val*	13.2	-	-	-	-	-
Ours	*val*	16.1	5.0	7.0	2.4	4.6	9.6
SORT_TAO [4]	*test*	10.2	4.4	4.9	7.7	8.2	15.2
Ours	*test*	12.4	4.5	5.2	3.7	8.3	18.8

Table 3: Results on the BDD100K segmentation tracking validation set. I: ImageNet. C: COCO. S: Cityscapes. B: BDD100K. "frozen" means adopting the pretrained model from the BDD100K tracking set and only finetune the mask head.

Method	Pretrained	mMOTSA ↑	mMOTSP ↑	mIDF1 ↑	ID sw. ↓
SORT [2]	I, C, S	11.4	59.7	22.1	15408
Ours	I, C, S	20.2	59.3	36.0	1681
Ours (frozen)	I, B	26.6	64.9	45.3	954

Table 4: Ablation studies of momentum of the embeddings on BDD100K tracking validation set. Note the model for this table is re-trained that the results are slightly different from the results in the main paper.

Momentum	mMOTA ↑	mIDF1 ↑	MOTA ↑	IDF1 ↑
0.6	37.0	50.9	63.3	71.4
0.7	37.0	50.9	63.3	71.3
0.8	37.0	50.7	63.3	71.1
0.9	37.0	50.6	63.3	70.8
1.0	37.0	50.5	63.3	70.5

C. Additional ablation studies

Momentum of the embeddings. Assume there is an existing track and its embedding is E_0. This track is associated to an object on the current frame and its embedding is E_1. The new embedding of this track will be $m * E_1 + (1 - m) * E_0$, where m is the momentum. The momentum does not improve the results too much but it considers the history of embeddings. We show the ablation studies of different values of momentum in Table 4.

Sensitivity of the γ_1 and γ_2 in Eq. 7. We found γ_2 does not change the final results while γ_1 does. If γ_1 is higher than 0.5, the performance will drop, but does not matter if it is lower than 0.5.

D. Oracle analysis

We investigate the performances of two types of oracles: detection oracle and tracking oracle on BDD100K tracking validation set. For detection oracle, we directly extract feature embeddings of the ground truth objects in each frame and associate them using our method. For tracking oracle, we use ground truth tracking labels to associate the detected objects.

Detection oracle The results are shown in Table 5. We can observe that all MOTAs are higher than 94%, and some of them are even close to 100%. This is because we use the ground truth boxes directly so that the number of false negatives and false positives are close to 0.

The metric IDF1 and ID Switches can measure the performance of identity consistency. The average IDF1 over the 8 classes is 88.8%, which is 38 points higher than our result. The gaps on classes "car" and "pedestrain" are only 11.1 points and 19.3 points between oracle results and our results respectively, while gaps on other classes are exceeding 30 points. These results show that if highly accurate detection results are provided, our method can obtain robust feature embeddings and associate objects effectively. However, the huge performance gaps also indicate the demand of promoting detection algorithms in the video domain. We also notice that the total number of ID switches in the oracle experiment is higher than ours. This is due to the high object recalls in the oracle experiments, as more detected instances may introduce more ID switches accordingly.

Tracking oracle The results are shown in Table 6. We can observe that when associating object directly with tracking labels, the mIDF1 is only boosted by 4.3 points. This promising oracle analysis shows the effectiveness of our method and indicates that our method is bounded more by detection performance than tracking performance.

E. Failure case analysis

Our method can distinguish different instances even they are similar in appearance. However, there are still some failure cases. We show them below with figures, in which we use yellow color to represent false negatives, red color to represent false positives, and cyan color to represent ID switches. The float number at the corner of each box indicates the detection score, while the integer indicates the object identity number. We use green dashed box to highlight the objects we want to emphasize.

Object classification Inaccurate classification confidence is the main distraction for the association procedure because false negatives and false positives destroy the one-to-one matching constraint. As shown in Figure 1, the false negatives are mainly small objects or occluded objects under crowd scenes. The false positives are objects that have similar appearances to annotated objects, such as persons in the mirror or advertising board, etc.

Inaccurate object category is a less frequent distraction caused by classification. The class of the instance may switch between different categories, which mostly belong to the same super-category. Figure 2 shows an example. The category of the highlighted object changes from "rider" to "pedestrian" when the bicycle is occluded. Our method fails in this case because we require the associated objects have the same category.

Table 5: Detection oracle analysis. The numbers in the round brackets mean the gaps between oracle results and our results.

Category	Set	MOTA ↑	IDF1 ↑	MOTP ↑	FN ↓	FP ↓	ID Sw. ↓	MT ↑	ML ↓
Pedestrian	val	94.3	79.5 (+19.3)	99.8	1	1	3226	3506	0
Rider	val	95.8	88.5 (+40.4)	99.9	0	0	107	134	0
Car	val	97.7	86.1 (+11.1)	99.9	0	0	7716	13189	0
Bus	val	99.2	93.0 (+31.2)	100.0	0	0	72	196	0
Truck	val	98.8	90.3 (+33.8)	100.0	0	0	340	726	0
Bicycle	val	88.2	79.5 (+31.8)	98.7	8	8	470	243	0
Motorcycle	val	97.0	94.5 (+37.8)	99.8	0	0	27	44	0
Train	val	99.4	98.7 (+98.7)	100.0	0	0	2	6	0
All	val	96.3	88.8 (+38.0)	99.8	9	9	11960	18044	0

Table 6: Tracking oracle analysis. The numbers in the round brackets mean the gaps between oracle results and our results.

Category	Set	MOTA ↑	IDF1 ↑	MOTP ↑	FN ↓	FP ↓	ID Sw. ↓	MT ↑	ML ↓
Pedestrian	val	54.7	71.2 (+11.0)	77.6	14990	10095	755	1835	367
Rider	val	31.4	52.6 (+4.5)	76.6	1390	242	115	16	56
Car	val	74.3	82.9 (+7.9)	84.1	54585	31014	2309	8759	1141
Bus	val	38.2	65.8 (+4.0)	86.1	3532	2031	57	61	41
Truck	val	37.0	60.9 (+4.4)	84.7	12719	4259	247	149	239
Bicycle	val	30.6	55.6 (+7,9)	75.4	2031	714	125	60	58
Motorcycle	val	14.6	51.7 (-5.0)	76.4	443	292	35	10	18
Train	val	-0.6	0.0 (+0.0)	0.0	308	2	0	0	6
All	val	35.0	55.1 (+4.3)	70.1	89998	48649	3643	10890	1926

These failure cases caused by object classification suggest the improvements on video object detection algorithms. We can exploit temporal or tracking information to improve the detectors, thus obtaining better tracking performance.

Object truncation/occlusion Object truncation/occlusion causes inaccurate object localization. As shown in Figure 3, the highlighted objects are truncated by other objects. The detector detects two objects. One of them is a false positive box that only covers a part of the object. The other one is a box with a lower detection score but covers the entire object. This case may influence the association process if the two boxes have similar feature embeddings.

An instance may have totally different appearances before and after occlusion that result in low similarity scores. As shown in Figure 4, only the front of the car appears before occlusion, while only the rear of the car appears after occlusion. Our method can associate two boxes if they cover the same discriminative regions of an object, not necessarily the exact same region. However, if two boxes cover totally different regions of the object, they will have a low matching score.

Another corner case is the extreme high-level truncation. As shown in Figure 5, the highly truncated objects only appear a little when they just enter or leave the camera view. We cannot distinguish different instances effectively according to the limited appearance information.

F. Visualizations

We show the visualizations of different instance patches during the testing procedure in Figure 6. The detected objects in each frame are matched to prior objects via bi-directional softmax. The prior objects include tracks in the consecutive frame, vanished tracks, and backdrops. We annotate them with different colors. Each detected object is enclosed by the same color of its matched object. We can observe that most false positives in the current frame are matched to backdrops, which demonstrates keeping backdrops during the matching procedure helps reduce the number of false positives.

G. Qualitative results

We show some qualitative results of our method on BDD100K dataset and MOT17 dataset in Figure 7 and Figure 8 respectively. The results are sampled from a certain interval for illustrative purposes.

References

[1] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixé. Tracking without bells and whistles. *arXiv preprint arXiv:1903.05625*, 2019.

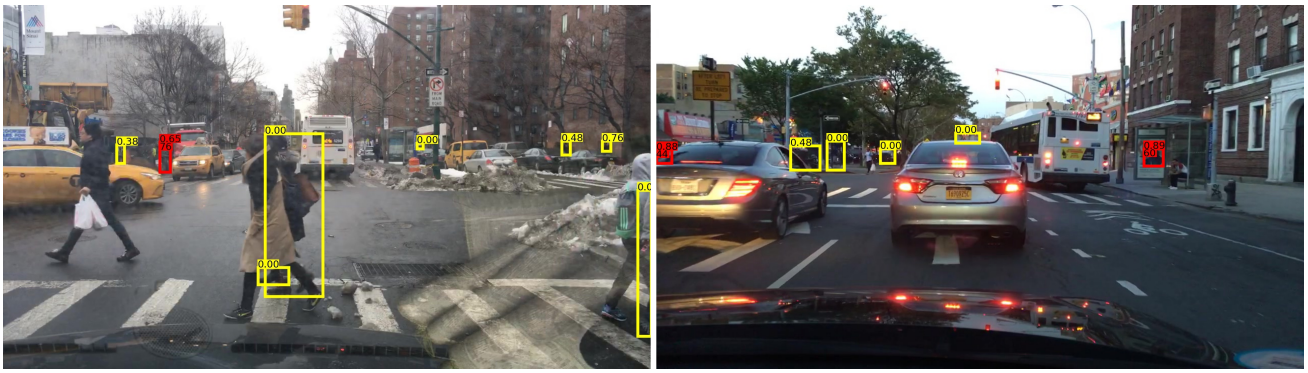

Figure 1: Failure cases caused by inaccurate classification confidences. The objects enclosed by yellow rectangles are false negatives, and the objects enclosed by red rectangles are false positives.

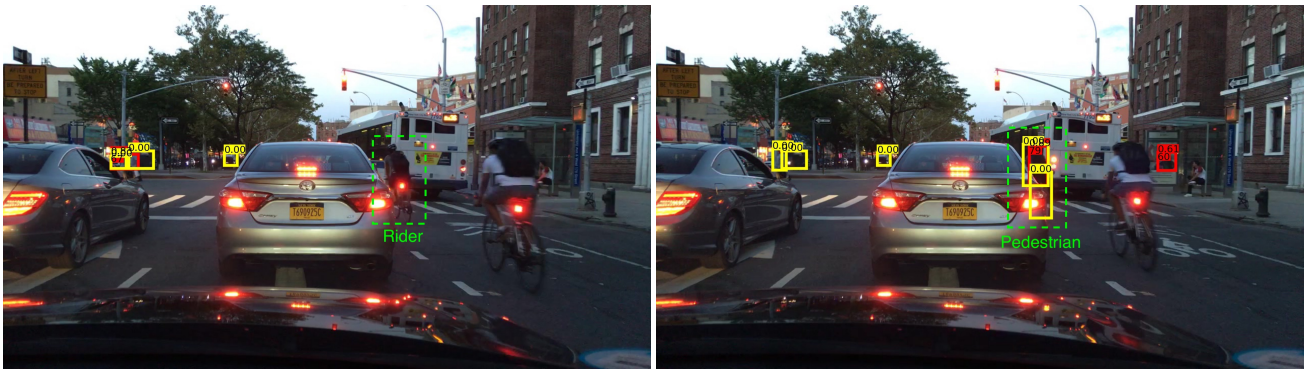

Figure 2: Failure case caused by inaccurate object category. The category of the highlighted object changes from "rider" to "pedestrian" due to the occlusion of the bicycle. They cannot be associated because they do not satisfy the category consistency.

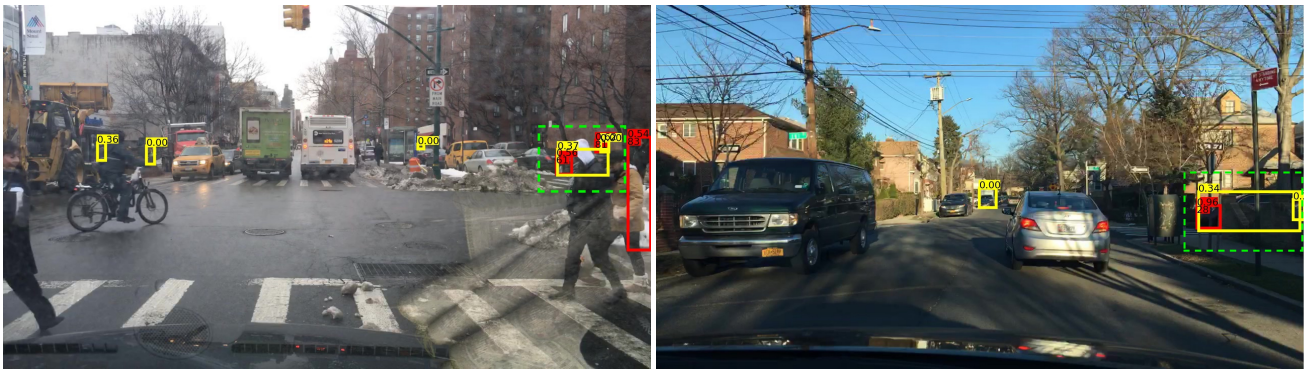

Figure 3: Inaccurate object localization caused by truncation. The red false positive box only covers part of the object, while the yellow box covers the entire object. They may have similar feature embeddings thus influencing the association procedure.

[2] Alex Bewley, ZongYuan Ge, Lionel Ott, Fabio Tozeto Ramos, and Ben Upcroft. Simple online and realtime tracking. In *International Conference on Image Processing*, 2016.

[3] Guillem Brasó and Laura Leal-Taixé. Learning a neural solver for multiple object tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[4] Achal Dave, Tarasha Khurana, Pavel Tokmakov, Cordelia Schmid, and Deva Ramanan. Tao: A large-scale benchmark for tracking any object. In *European Conference on Computer Vision*, 2020.

[5] Andrea Hornakova, Roberto Henschel, Bodo Rosenhahn, and Paul Swoboda. Lifted disjoint paths with application in multiple object tracking. In *International Conference on Machine Learning*, 2020.

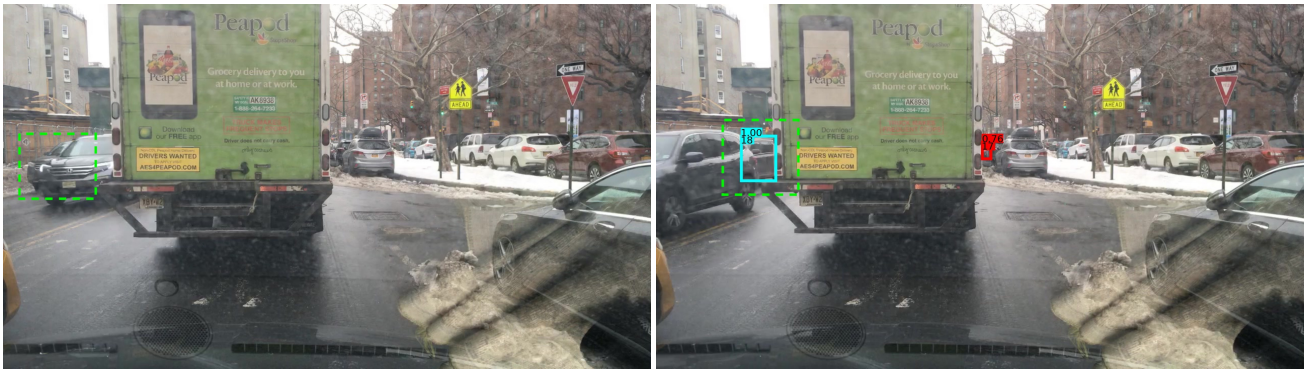

Figure 4: Two detected objects in different frames cover totally different regions of the object thus having low appearance similarity.

Figure 5: Our method cannot distinguish different instances effectively according to the limited appearance information in highly truncated objects.

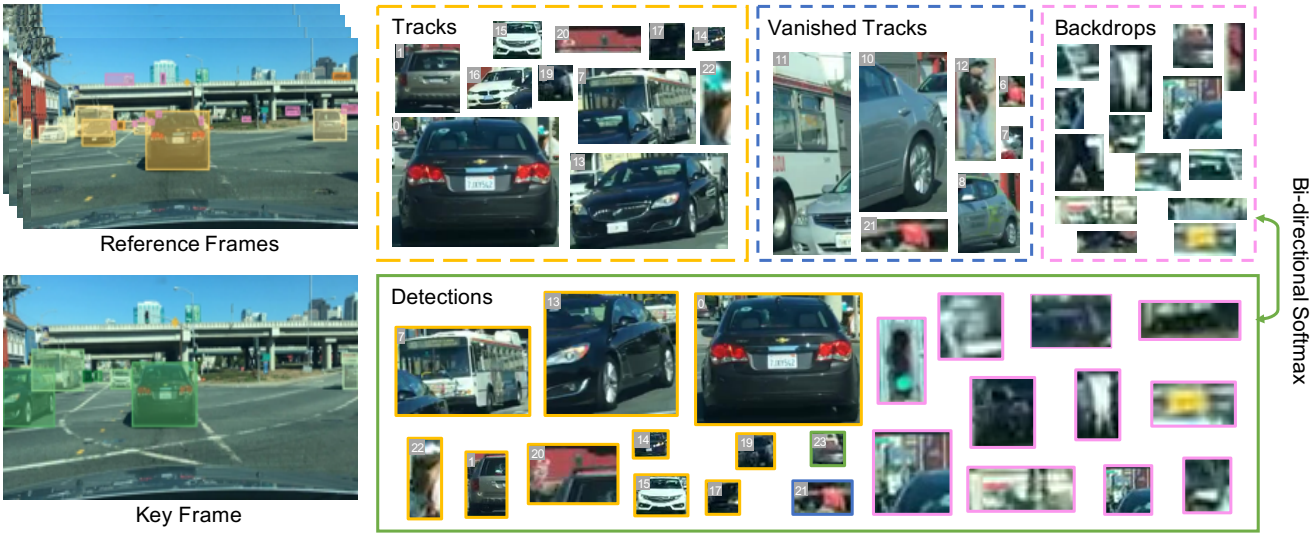

Figure 6: The visualizations of different instance patches during the testing procedure. The detected objects in the current frame are matched to tracklets in the consecutive frame, vanished tracklets, and backdrops via bi-directional softmax

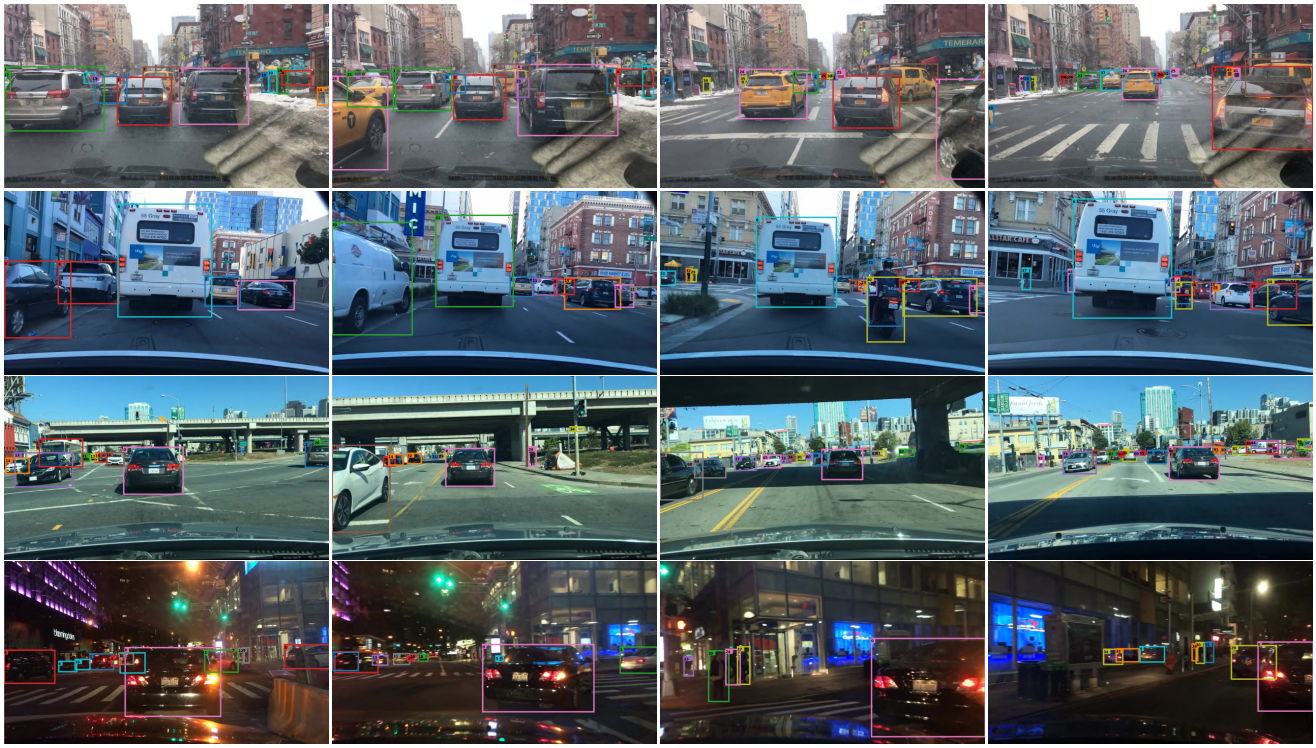

Figure 7: Qualitative results of our method on BDD100K dataset.

Figure 8: Qualitative results of our method on MOT17 dataset.

[6] Qiankun Liu, Qi Chu, Bin Liu, and Nenghai Yu. Gsm: Graph similarity model for multi-object tracking. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, 2020.

[7] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. In *European Conference on Computer Vision*, 2020.