MOTSynth: How Can Synthetic Data Help Pedestrian Detection and Tracking?

Matteo Fabbri[1,3] Guillem Brasó[2] Gianluca Maugeri[1] Orcun Cetintas[2] Riccardo Gasparini[1,3]
Aljoša Ošep[2] Simone Calderara[1] Laura Leal-Taixé[2] Rita Cucchiara[1]
[1]University of Modena and Reggio Emilia, Italy [2]Technical University of Munich, Germany
[1]{firstname.lastname}@unimore.it [2]{firstname.lastname}@tum.de
[3]GoatAI S.r.l.
[3]{firstname.lastname}@goatai.it

Dataset	#Clips	#Frames	#Instances	3D	Occl.	Pose Est.	Inst. Segm.	Depth Est.	Type
KITTI [14]	50	22,000	160,000	✓	✓			✓	AD
nuSCENES [7]	1,000	40,000	280,000	✓					AD
BDD100k-MOTS [30]	70	14,000	129,000		✓		✓		AD
BDD100k-MOT [30]	1,600	100,000	3,300,000		✓				AD
Waymo Open [24]	1,150	230,000	2,700,000	✓					AD
TAO [10]	2,907	148,235	175,723						DV
PoseTrack [3]	1,356	46,000	276,000			✓			DV
MOTS [25]	4	2,862	26,894		✓		✓		US
MOT-17 [21]	14	11,235	292,733		✓				US
MOT-20 [11]	8	13,410	1,652,040		✓				US
VIPER [18]	187	254,064	2,750,000	✓	✓		✓		AD
GTA [19]	-	250,000	3,875,000				✓	✓	DV
JTA [12]	512	460,800	15,341,242	✓	✓	✓			US
MOTSynth	768	1,382,400	40,780,800	✓	✓	✓	✓	✓	US

Table 1: Overview of the publicly available datasets for pedestrian detection and tracking. For each dataset, we report the numbers of clips, annotated frames and instances. We also report the presence of 3D data and occlusion information, as well as the availability of labels for pose estimation, instance segmentation, and depth estimation. The last column shows the data type: autonomous driving (AD), diverse (DV) or urban surveillance (US).

1. Overview

In this supplementary, we provide (i) extended version of the Tab. 1 (dataset comparison), provided in the main paper (Sec. 2); (ii) additional dataset visualizations and statistics (Sec. 3); (iii) additional experiments on trade-offs on data volume vs. diversity (Sec. 4); (iv) implementation details for all experiments, provided in the main paper (Sec. 5); (v) MOT20 benchmark results for each sequence (Sec. 6).

2. Dataset Comparison

Tab. 1 extends Tab. 1 from the main paper. In the table the most widely used publicly available datasets that contain annotation for the people class are reported. Compared to real world urban surveillance dataset, *MOTSynth* has one order of magnitude more clips, annotated frames and annotated instances. Besides JTA [13], *MOTSynth* is the only available dataset that provides 3D pose annotations. Additionally, *MOTSynth* also provides instance segmentation labels and depth maps. It is important to note that for autonomous driving datasets [14, 7, 30, 24] and TAO [10] the number of instances is relative to all the classes;

3. Dataset Visualizatons and Statistics

In Fig. 1 we show examples from the *MOTSynth* dataset to demonstrate its variation in terms of weather conditions, lighting conditions, viewpoints, and pedestrian density. We recorded sequences exhibiting nine different types

Figure 1: Examples from the *MOTSynth* dataset showing data variety in terms of weather conditions (first row), lighting condition (second row), viewpoints (third row) and number of people (fourth row). Best viewed on screen.

of weather: *clear, extra sunny, cloudy, overcast, rainy, thunder, smog, foggy, and blizzard*.

In addition, *MOTSynth* varies in terms of: (i) *lighting conditions*, resembling different day-time conditions, such as *sunrise, sunset, evening, dawn and night*; (ii) the *camera viewpoint*, ranging from ground plane position to bird's-eye view, and (iii) *density*, ranging from few pedestrians to hundreds of pedestrians.

We present a more detailed analysis of *MOTSynth* in Fig 2. In Fig. 2a we plot the distribution of the bounding box heights expressed in pixels. As can be seen, 50% of the bounding boxes are between 0 and 95 pixels. Only 2% of them are higher than 613 pixel. This clearly shows that *MOTSynth* has been designed specifically for surveillance applications.

In Fig. 2b we show the distribution of the number of bounding boxes per frame, ranging between 0 and 125 with a mean of 29.50 and a standard deviation of 17.12. The distribution is well balanced as peak values hardly reach a frequency of 2.5%.

In Fig. 2c, we plot the distance distribution of each pedestrian computed as the distance between the camera and the head joint expressed in meters. The average camera distance is 28.49 meters, while the standard deviation is 20.33 meters. Half of the annotations appear in 23m range from the camera. Again, the peaks of the distribution never exceed 3% showing good data balance.

In Fig. 2d, we plot pedestrian visibility distribution. It is calculated by counting the number of not occluded body joints, i.e., joints that are not obstructed by objects or other pedestrians and that are thus completely visible. *MOTSynth* provide the annotation for 22 body joint, thus, a person is completely visible only if all his 22 joints are not occluded. The plot clearly shows that *MOTSynth* is highly crowded as the percentage of completely visible pedestrians is less than 20%.

4. Data Volume and Diversity

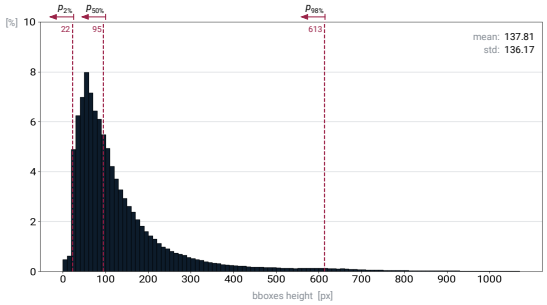
In the main paper, we discussed the impact of the data sampling rate based on the Faster R-CNN detector [23]. Here, we provide this analysis for all object detectors we experiment with in Tab. 2.

YOLOv3 requires 104k images to perform favourably w.r.t. COCO. Moreover, higher sampling rate is always beneficial both in term of AP and MODA. For CenterNet, the sampling rate does not impact the AP. For MODA, on the other hand, higher data volume seems to be beneficial.

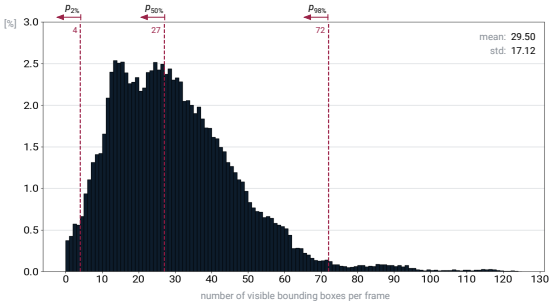
It is interesting to note that CenterNet is able to surpass COCO training with only 17k images. Moreover, it is clear that visual diversity is crucial as split 4 with 1/60 sampling rate (17k images) surpasses the split 3 with 1/10 sampling rate (52k images).

Results on Faster R-CNN are even more evident. With only 9k images we obtain higher AP w.r.t. real data training. However, results seem to saturate with bigger splits. For both YOLOv3 and Faster R-CNN, split 2 with 1/10 sampling rate (24k frames) and split 3 and 4 with 1/60 sampling rate (9k and 17k frames respectively) obtain similar performance.

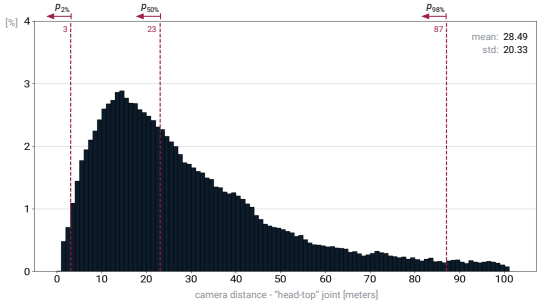
This shows that volume and diversity are equally important. In general, visual diversity and data volume are equally important to achieve competitive results as the best performance is always obtained when diversity and volume are maximized.



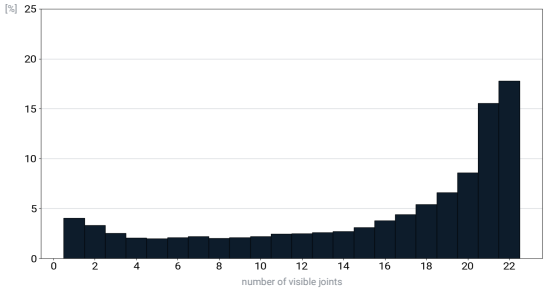
(a) Distribution of the bounding box heights over Full HD images



(b) Distribution of the number of bounding boxes per frame.



(c) Camera distance distribution of every annotated pedestrian relative to the head joint.



(d) Visibility distribution of every annotated pedestrian reported as the number of visible joints.

Figure 2: Additional statistics of the *MOTSynth* dataset. Each distribution is calculated considering all pedestrians with at least one visible joint.

5. Implementation Details

5.1. Object Detection Experiments

YOLOv3. For YOLOv3, we used Darknet backbone [22]. We trained our model on *MOTSynth* for $200,000$ iterations using the batch-size of 16. We resize input images to 608×608. We used the Ultralytics implementation [2] with default hyperparameters. For the evaluation, we used a confidence threshold of 0.4 when testing on MOT17 and MOT20.

CenterNet. For CenterNet, we used DLA-34 backbone [29] and used used the official implementation of CenterNet [1]. We trained on *MOTSynth* for $100,000$ iterations using batch size 32 (we used two GPUs). During the inference, we used a confidence threshold of 0.3 when testing on MOT17 and a confidence threshold of 0.1 when testing on MOT20.

Faster R-CNN and Mask R-CNN. For Faster R-CNN, we use a ResNet50 [16] backbone with FPN [20] (Detectron2 [15] implementation). We train models on *MOTSynth* for $35,000$ iterations and use default Detectron2 hyperparameters. To avoid overfitting, we freeze all the backbone blocks except for the last one. For fine-tuning, we follow [4] and train our models for 30 additional epochs on the respective dataset. Similarly, we follow the same training scheme and use the same hyperparameters for Mask R-CNN.

5.2. Person Re-Identification Experiments

For ReID, we follow [8]: we freeze all CNN layers and pre-train the fully connected layers for 5 epochs. We then train our entire models for 55 additional epochs using Adam optimizer (citation needed) and a learning rate of 0.004. We resize images to $128x56$ and use random cropping and flipping data augmentation techniques.

5.3. Multi-Object Tracking Experiments

MOT. For *CenterTrack* [32], we follow the training schemes explained in Section 4.4 of the main paper. We fine-tune our network for 30 epochs for MOT17 and 70 epochs for the MOT20 dataset for the fine-tuning experiments. We train and evaluate our models using the same hyperparameters as reported by [32]. For *Tracktor* [4], we follow the setting described for Faster R-CNN and ReID, as no additional training is required: *Tracktor* leverages bounding box regression head of Faster R-CNN detector, trained on static images.

	Dataset	Split	Sampling rate	frames	AP	MODA
YOLOv3	COCO	–	–	118k	69.76	62.02
	MOTSynth	1	1:60	2k	51.15	45.71
			1:10	13k	62.66	52.36
		2	1:60	4k	53.86	47.49
			1:10	24k	63.08	56.67
		3	1:60	9k	62.10	51.20
			1:10	52k	63.13	60.60
		4	1:60	17k	62.59	58.66
			1:10	104k	**71.90**	**64.51**
CenterNet	COCO	–	–	118k	67.01	44.38
	MOTSynth	1	1:60	2k	61.18	39.06
			1:10	13k	61.82	49.34
		2	1:60	4k	61.45	44.54
			1:10	24k	62.32	54.90
		3	1:60	9k	62.22	53.04
			1:10	52k	62.45	55.82
		4	1:60	17k	70.15	51.75
			1:10	104k	**70.68**	**57.39**
Faster R-CNN	COCO	–	–	118k	76.68	53.86
	MOTSynth	1	1:60	2k	70.00	42.90
			1:10	13k	76.80	39.02
		2	1:60	4k	70.27	44.54
			1:10	24k	77.47	50.62
		3	1:60	9k	77.32	51.46
			1:10	52k	78.30	49.75
		4	1:60	17k	77.78	53.72
			1:10	194k	**78.98**	**54.96**

Table 2: The effect of the density of sampled data. Sparser sampling increases the diversity. As can be seen, we can bridge the gap syn-to-real even when using smaller *MOTSynth* subsets if we ensure that training images are diverse.

MOTS. We adapt our Mask R-CNN model, trained on *MOTSynth*, by using bounding box regression mechanism for tracking and mask segmentation head provides segmentation masks (Mask R-CNN Tracktor (†)). For all experiments and the benchmark submission, we use the same ReID network and hyperparameters as reported in [4].

6. Detailed Benchmark Results

In Tab. 3 we present the detailed MOT20 benchmark results for each sequence and analyze how Tracktor and CenterTrack (trained only on *MOTSynth*) compare with the state-of-the-art trackers in extremely crowded scenes. In addition to published models, we train and evaluate CenterTrack on MOT20 (denoted with ‡), following the training procedure of [32].We are interested in comparing existing models trained on different datasets. Therefore, we use the default CenterTrack hyperparameters.

We observe that in sequence MOT20-04, Tracktor-*MOTSynth* and CenterTrack-*MOTSynth* are not on-par with Tracktorv2, MPNTrack and LPC. This is likely because the sequences with near-bird's-eye viewpoints (similar to MOT20-04) are rare in *MOTSynth* dataset. However, in all other MOT20 sequences, Tracktor and CenterTrack only trained on synthetic data outperform Tracktorv2 with a significant margin and are on-par with the state-of-the-art. Fine-tuning these models on MOT20 further improves their performance, as reported in Section 4.6 of the paper. These

	Method	MOTA ↑	MOTP ↑	IDF1 ↑	FP ↓	FN ↓	IDS ↓
MOT20-04	Tracktor-*MOTSynth*	50.7	75.5	42.6	7383	125803	1963
	CenterTrack-*MOTSynth*	41.7	74.5	38.7	15154	142557	2152
	CenterTrack‡ [32]	54.9	**81.1**	43.7	**2187**	118918	2641
	Tracktorv2 [4]	72.7	80.1	65.4	2855	71164	739
	MPNTrack [6]	**77.0**	79.6	71.2	7459	**55204**	**506**
	LPC [9]	75.7	80.6	**75.7**	4180	61864	648
	SORT20 [5]	59.5	81.0	56.7	3206	106117	1643
MOT20-06	Tracktor-*MOTSynth*	35.5	73.9	33.7	4594	80171	871
	CenterTrack-*MOTSynth*	**40.8**	71.8	35.3	12448	**64330**	1748
	CenterTrack‡ [32]	26.4	75.9	29.0	17481	78371	1881
	Tracktorv2 [4]	30.1	**78.8**	33.2	**1745**	90509	512
	MPNTrack [6]	36.0	77.1	39.8	4831	79649	**425**
	LPC [9]	35.3	77.7	**43.2**	3503	81891	499
	SORT20 [5]	23.7	73.1	29.5	12309	87352	1640
MOT20-07	Tracktor-*MOTSynth*	52.5	77.5	50.9	509	15009	194
	CenterTrack-*MOTSynth*	53.5	74.6	46.3	3082	**11785**	539
	CenterTrack‡ [32]	45.2	80.9	41.9	1101	16728	303
	Tracktorv2 [4]	50.1	**81.1**	49.6	252	16127	146
	MPNTrack [6]	**57.4**	79.5	**59.9**	906	13061	120
	LPC [9]	50.8	79.3	58.9	**229**	15921	**124**
	SORT20 [5]	48.5	77.6	47.3	1032	15666	360
MOT20-08	Tracktor-*MOTSynth*	**29.4**	73.2	33.5	3447	50831	439
	CenterTrack-*MOTSynth*	24.6	68.7	31.4	16382	**40602**	1433
	CenterTrack‡ [32]	9.0	73.9	25.8	19736	49828	948
	Tracktorv2 [4]	21.0	**78.8**	27.2	**2078**	58880	251
	MPNTrack [6]	25.9	77.3	36.1	3757	53470	**159**
	LPC [9]	25.8	76.3	**37.4**	3814	53380	291
	SORT20 [5]	13.1	70.9	24.2	10974	55559	827

Table 3: Per-sequence benchmark results on MOT20.

	Method	MOTA ↑	IDF1 ↑	FP ↓	FN ↓	IDS ↓
Public	Tracktor-*MOTSynth*	56.9	56.9	20852	220273	2012
	Tracktor-*MOTSynth* + FT	59.1	58.8	22231	206062	2323
	Tracktor [4]	53.5	52.3	12201	248047	2072
	Tracktorv2 [4]	56.3	55.1	**8866**	235449	1987
	CenterTrack-*MOTSynth*	59.7	52.0	39707	181471	6035
	CenterTrack-*MOTSynth* + FT	**65.1**	57.9	11521	**180901**	4377
	CenterTrack [32]	61.5	59.6	14076	200672	2583
	Lif_T [17]	60.5	65.6	14966	206619	1189
	LPC [9]	59.0	**66.8**	23102	206948	**1122**
	MPNTrack [6]	58.8	61.7	17413	213594	1185
Private	CorrTracker [26]	**76.5**	**73.6**	29808	**99510**	3369
	FairMOTv2 [31]	73.7	72.3	27507	117477	**3303**
	TraDeS [28]	69.1	63.9	**20892**	150060	3555

Table 4: Detailed Benchmark results on MOT17.

	Method	MOTA ↑	IDF1 ↑	FP ↓	FN ↓	IDS ↓
Public	Tracktor-*MOTSynth*	43.7	39.7	15933	271814	3467
	Tracktor-*MOTSynth* + FT	56.5	52.8	11377	211772	1995
	Tracktorv2 [4]	52.6	52.7	**6930**	236680	1648
	CenterTrack-*MOTSynth*	39.7	37.2	47066	259274	5872
	CenterTrack-*MOTSynth* + FT	41.9	38.2	36594	258874	5313
	MPNTrack [6]	**57.6**	59.1	16953	**201384**	**1210**
	LPC [9]	56.3	**62.5**	11726	213056	1562
	SORT20 [5]	42.7	45.1	27521	264694	4470
Private	JDMOTGNN [27]	**67.1**	67.5	**31913**	135409	**3131**
	CorrTracker [26]	65.2	**69.1**	79429	95855	5183
	FairMOTv2 [31]	61.8	67.3	103440	**88901**	5243

Table 5: Detailed Benchmark results on MOT20.

experiments indicate that top-performing tracking models can be trained on synthetic data even in extremely dense scenarios.

References

[1] Centernet. https://github.com/xingyizhou/CenterNet. Accessed: 2021-03-08. 3

[2] Yolov3. https://github.com/ultralytics/yolov3. Accessed: 2021-03-08. 3

[3] Mykhaylo Andriluka, Umar Iqbal, Eldar Insafutdinov, Leonid Pishchulin, Anton Milan, Juergen Gall, and Bernt Schiele. Posetrack: A benchmark for human pose estimation and tracking. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018. 1

[4] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixé. Tracking without bells and whistles. In *Int. Conf. Comput. Vis.*, 2019. 3, 4

[5] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *IEEE Int. Conf. Image Process.*, 2016. 4

[6] Guillem Brasó and Laura Leal-Taixé. Learning a neural solver for multiple object tracking. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020. 4

[7] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020. 1

[8] H. Chen, Y. Wang, Y. Shi, K. Yan, M. Geng, Y. Tian, and T. Xiang. Deep transfer learning for person re-identification. In *IEEE Int. Conf. Multimedia Big Data*, pages 1–5, 2018. 3

[9] Peng Dai, Renliang Weng, Wongun Choi, Changshui Zhang, Zhangping He, and Wei Ding. Learning a proposal classifier for multiple object tracking. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021. 4

[10] Achal Dave, Tarasha Khurana, Pavel Tokmakov, Cordelia Schmid, and Deva Ramanan. TAO: A large-scale benchmark for tracking any object. In *Eur. Conf. Comput. Vis.*, 2020. 1

[11] Patrick Dendorfer, Hamid Rezatofighi, Anton Milan, Javen Shi, Daniel Cremers, Ian Reid, Stefan Roth, Konrad Schindler, and Laura Leal-Taixé. Mot20: A benchmark for multi object tracking in crowded scenes. *arXiv preprint arXiv:2003.09003*, 2020. 1

[12] Matteo Fabbri, Fabio Lanzi, Simone Calderara, Andrea Palazzi, Roberto Vezzani, and Rita Cucchiara. Learning to detect and track visible and occluded body joints in a virtual world. In *Eur. Conf. Comput. Vis.*, 2018. 1

[13] Matteo Fabbri, Fabio Lanzi, Simone Calderara, Andrea Palazzi, Roberto Vezzani, and Rita Cucchiara. Learning to detect and track visible and occluded body joints in a virtual world. In *Eur. Conf. Comput. Vis.*, September 2018. 1

[14] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2012. 1

[15] Ross Girshick, Ilija Radosavovic, Georgia Gkioxari, Piotr Dollár, and Kaiming He. Detectron. https://github.com/facebookresearch/detectron. Accessed: 2021-03-08. 3

[16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016. 3

[17] Andrea Hornakova, Roberto Henschel, Bodo Rosenhahn, and Paul Swoboda. Lifted disjoint paths with application in multiple object tracking. In *IEEE Int. Conf. Mach. Learn.*, 2020. 4

[18] Dahun Kim, Sanghyun Woo, Joon-Young Lee, and In So Kweon. Video panoptic segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020. 1

[19] Philipp Krähenbühl. Free supervision from video games. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018. 1

[20] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, July 2017. 3

[21] Anton Milan, Laura Leal-Taixé, Ian Reid, Stefan Roth, and Konrad Schindler. Mot16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831*, 2016. 1

[22] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 779–788, 2016. 3

[23] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Adv. Neural Inform. Process. Syst.*, 2015. 2

[24] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020. 1

[25] Paul Voigtlaender, Michael Krause, Aljoša Ošep, Jonathon Luiten, B.B.G Sekar, Andreas Geiger, and Bastian Leibe. MOTS: Multi-object tracking and segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019. 1

[26] Qiang Wang, Yun Zheng, Pan Pan, and Yinghui Xu. Multiple object tracking with correlation learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3876–3886, June 2021. 4

[27] Yongxin Wang, Xinshuo Weng, and Kris Kitani. Joint detection and multi-object tracking with graph neural networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020. 4

[28] Jialian Wu, Jiale Cao, Liangchen Song, Yu Wang, Ming Yang, and Junsong Yuan. Track to detect and segment: An online multi-object tracker. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 12352–12361, June 2021. 4

[29] Fisher Yu, Dequan Wang, Evan Shelhamer, and Trevor Darrell. Deep layer aggregation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2403–2412, 2018. 3

[30] Wentao Yuan, Tejas Khot, David Held, Christoph Mertz, and Martial Hebert. Pcn: Point completion network. In *Int. Conf. 3D Vis.*, 2018. 1

[31] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. A simple baseline for multi-object tracking. *arXiv preprint arXiv:2004.01888*, 3(4):6, 2020. 4

[32] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. In *Eur. Conf. Comput. Vis.*, 2020. 3, 4