Supplementary material: PolarMOT: How far can geometric relations take us in 3D multi-object tracking?

Aleksandr Kim, Guillem Brasó, Aljoša Ošep, and Laura Leal-Taixé

Technical University of Munich, Germany
{aleksandr.kim,guillem.braso,aljosa.osep,leal.taixe}@tum.de

1 Implementation details

1.1 Training and augmentation

Training. As training data we use annotated training keyframes from the nuScenes [1] dataset. We represent labeled boxes as nodes in the graph. Edges in the graph are labeled positive if they connect nodes with the same track ID (across any number of frames) and negative otherwise. During model training, all input clips are processed individually and each edge is considered an independent sample that contributes to the total focal loss [4].

Data augmentation. To mimic noisy, real-world detectors, we rely on data augmentation. We add random bounding box detections at each frame before the graph construction to imitate false positive detections. For each frame, the number of added boxes is a fraction of the number of real boxes (between 0.7 and 0.9 for each class) plus a fixed number (between 1 and 3) to augment completely empty frames. Pose coordinates (position and orientation) of augmented boxes are sampled from uniform distributions whose parameters are the minimum and maximum values of corresponding coordinates to labeled boxes.

We also augment input graphs at each training iteration. To mimic occlusions and miss-classifications, we randomly drop nodes (between 40%-60%) at each frame as well as some complete frames from our graphs. To emulate imprecise detections, we further perturb each initial edge feature, representing differences in object poses, with noise vectors sampled from class-specific Gaussian distributions with zero mean. For each class, the standard deviation for distributions of distance noise (meters) is between 0.05 and 0.35, for polar angle noise (radians) between 0.1 and 0.25, and for orientation noise (radians) between 0.05 and 0.25. For further augmentation, we fully remove approximately 20% of all edges in the graph. These augmentations ensure that our model is robust to imperfect/noisy inputs that we obtain from real-world 3D object detectors.

Inference. In multi-object tracking, track IDs need to be distinct at each frame, *i.e.*, only one object detection can be assigned to each identity (track ID). In our problem setting, this means that every node in the graph can have at most one positive edge connecting it to each past and future frame. We do not impose

Table 1: Neural network architecture of *PolarMOT*. Each cell describes the output dimensionality of each layer in the fully-connected MLPs of our model

	Edge initial	Node initial	Edge model	Edge pres, past, fut	Node model	Final edge classifier
Input	4	48	80	80	96	16
1st layer output	16	64	64	64	128	64
2nd layer output	16	128	16	32	64	32
3rd layer output		32			32	16
4th layer output						1

this constraint on *PolarMOT* during the training (*i.e.*, we train our model as an unconstrained binary classifier). However, we do ensure this constraint during inference via a simple post-processing procedure.

In particular, given edge classification scores from our model, edges with a score higher than a certain threshold (between 0.5 and 0.8 for each class) are positive, others are negative. Then, positive edge labels are *greedily* assigned starting from the highest score. As soon as a positive edge is assigned between nodes o_i^k and o_j^m at frames k and m, all other edges between node o_i^k and frame m (and between node o_j^m and frame k) are ignored from further assignment. This greedy procedure ensures that positive edges with the highest confidence are assigned first and each node has at most one positive edge to each frame.

1.2 Network structure

In this section, we detail the architecture of our network, explained in Sec. 4.2 in the main paper. In Tab. 1, we outline our network architecture composed entirely of multi-layer perceptrons (MLPs) with fully-connected layers. For each MLP, we list the dimensionality of all of its layers: input, intermediary and output. Here, "Edge initial" and "Node initial" columns correspond to MLP_{edge_init} and MLP_{node_init} (Eq. 4 and 5 in the paper), which produce initial learned embeddings from the initial relative features. The "Edge model" column describes MLP_{edge} (Eq. 1 in the paper) that processes edge features at each message passing step. The columns "Edge pres., past&fut." describe the identical composition of MLP_{pres} and $\text{MLP}_{past}/\text{MLP}_{fut}$, which process intra-frame edges and two temporal directions of inter-frame edges (Eq. 2 in the paper). The "Node model" column outlines MLP_{node} that aggregates all edge embeddings and produces node features at each step (Eq. 3 in the paper). Finally, the last column denotes the structure of the MLP used to classify edges based on the latest edge embeddings. We use leaky ReLU [5] between all layers of the network.

In our experiments, we always perform $L = 4$ message passing steps. For offline inference, we process clips of 11 frames and for online tracking, we keep only the 3 latest detections for each track history.

Table 2: Ablation on parametrization of geometric relations among objects on the nuScenes validation set. Trained on the **full** training set

Localized polar	Normalized by time	IDs ↓ total	Recall ↑ average	AMOTA ↑ average	class-specific AMOTA ↑ car	ped	bicycle	bus	motor	trailer	truck
✓	✓	213	**75.14**	**71.14**	85.83	**81.70**	**54.10**	87.36	**72.32**	**48.67**	**68.03**
✓	✗	**182**	72.85	70.27	**86.12**	**81.70**	51.73	**87.79**	69.29	47.20	**68.07**
✗	✓	225	70.90	69.75	85.89	**81.72**	48.92	87.54	69.25	47.60	67.31

Table 3: Ablation on the impact of contextual aggregation in node updates on the nuScenes validation set

Node aggregation connections	IDs ↓ total	Recall ↑ average	AMOTA ↑ average	class-specific AMOTA ↑ car	ped	bicycle	bus	motor	trailer	truck
Past/Present/Future	**213**	**75.14**	**71.14**	**85.83**	**81.70**	**54.10**	**87.36**	**72.32**	**48.67**	**68.03**
Spatial/Temporal	968	60.29	55.83	81.48	77.46	48.76	61.09	30.23	29.22	62.56
All together	765	26.75	23.30	0	72.56	40.75	19.80	0	0	30.04

2 Experimental evaluation

2.1 Edge parametrization ablation on the full training set

In the main paper, we ablated the impact of our proposed feature representation (time-normalized localized polar coordinates) by comparing models trained on the official nuScenes mini split. The main advantage of our representation is the inductive bias, which helps the model better understand long trajectories, turns and non-holonomic motion in general. The benefits of this inductive bias are best demonstrated when the amount of training data is low, which is why we used the mini split in our ablation.

For completeness, in Tab. 2, we provide the same ablation when the full training set is used. Unsurprisingly, with enough data (*e.g.* car, pedestrian and bus classes), different feature representations perform similarly because there are enough samples to learn motion bias directly from data. On the other hand, for rarely-observed classes, such as bicycles, motorcycles and trailers, using a better feature representation is clearly beneficial, *e.g. bicycle* AMOTA rises from 48.92 to 54.10. This aligns with our main ablation results, where our parametrization outperforms standard representation in low data regimes.

2.2 Contextual node aggregation

During message passing, our node aggregation step processes messages from past, present and future separately to maintain contextual awareness. To ablate the importance of this technique, we evaluate 3 versions of our trained model with different aggregation logic and show the results in Tab. 3.

When both temporal messages (past and future) are aggregated together, the model loses its time-awareness and its tracking performance significantly declines −15.31 avg. AMOTA (from 71.14 down to 55.83). Moreover, if spatial messages

Table 4: CenterPoint (CP) [6] and our method when trained and evaluating only on one city

Train city → eval city	Tracking model	IDs ↓ total	Recall ↑ average	AMOTA ↑ average	class-specific AMOTA ↑ car	ped	bicycle	bus	motor	trailer	truck
Singapore → Singapore	Ours	**145**	**77.82**	**75.79**	**85.42**	**77.17**	**51.41**	**85.76**	**74.23**	0	**80.73**
	CP	351	72.97	70.22	82.66	71.97	39.71	85.17	62.36	0	79.44
Boston → Boston	Ours	**48**	**70.94**	**70.17**	**85.80**	**85.16**	**58.57**	**83.45**	**60.89**	**52.75**	**64.58**
	CP	246	67.78	65.96	83.54	81.37	50.00	82.47	51.63	48.61	64.10

Table 5: Extended results of state-of-the-art methods for 3D multi-object tracking on the NuScenes test set benchmark. Legend: L – lidar, P – ego poses, B – 3D boxes

Method name	Input modality	MT ↑ total	ML ↓ total	Frag ↓ total	TID ↓ average	LGD ↓ average
Ours	3D (B)	**5701**	**1686**	**332**	0.444	**0.657**
OGR3MOT [7]	3D (B + P)	5278	2094	371	0.575	0.782
CenterPoint [6]	3D (L + P)	5399	1818	553	**0.415**	0.720
IPRL-TRI [2]	3D (B + P)	4294	2184	776	0.960	1.376
AlphaTrack[7]	3D + 2D + P	5560	1744	480	0.409	0.755
EagerMOT [3]	3D + 2D + P	5303	1842	601	0.448	0.801

are also aggregated in the same single group, the model is completely unaware of scene context and geometric scene composition, so its avg. AMOTA falls further by −32.53 (a total decline of 47.84 from our default contextual aggregation).

2.3 Cross-city generalization oracle results

In the main paper, we demonstrated the ability of our model to generalize across different cities by training it in Boston and evaluating in Singapore (and vice versa). To provide a better baseline and show how well the model would normally perform in each of the cities, we present Tab. 4 where models are trained and evaluated on the same single city, *i.e.* only Boston or only Singapore.

Since *PolarMOT* demonstrates better performance than CenterPoint [6] on the full nuScenes [1] validation set (see Tab. 2 in the main paper), it is unsurprising that its results on individual cities are also better.

2.4 Extended evaluation results

In this section, we present extended versions of the experimental evaluations detailed in the main paper. These tables include additional tracking metrics reported by the nuScenes benchmark [1], which we provide for completeness:

- MT (*mostly tracked*): percentage of tracks tracked correctly for at least 80% of their life span
- ML (*mostly lost*) percentage of tracks tracked correctly for at most 20% of their life span

Table 6: Extended results for online vs. offline tracking on the nuScenes val set [1]

Method name	Input modality	MT ↑ total	ML ↑ total	Frag ↓ total	TID ↓ average	LGD ↓ average
Ours *offl.*	3D	**4524**	**1452**	332	**0.379**	**0.672**
Ours *onl.*	3D	4262	1545	**285**	0.636	0.901
CenterPoint *onl.*	3D	4405	1508	445	0.516	0.956

Table 7: Extended results for CenterPoint (CP) [6] and our method when trained on training data from one city, and evaluated on the validation data from another

Train city → eval city	Tracking model	MT ↑ total	ML ↓ total	Frag ↓ total	TID ↓ average	LGD ↓ average
Boston → Singapore	Ours	**1595**	**765**	**139**	**0.461**	**0.773**
	CP	1464	861	224	0.584	1.037
Singapore → Boston	Ours	2274	1177	**147**	0.850	**1.171**
	CP	**2282**	**1147**	253	**0.715**	1.209

– Frag. (*fragmentations*): the number of times a trajectory is interrupted during tracking.
– TID (*track initialization in seconds*): time until the first detection of the track is successfully tracked.
– LGD (*longest gap duration in seconds*): time an object instance has been incorrectly tracked.

In Tab. 5 we show extended results on the nuScenes test set benchmark (Tab. 9 in the main paper).

Tab. 6 extends Tab. 2 in the main paper and details offline and online model evaluations on the nuScenes validation set.

Tab. 7 extends Tab. 7 in the main paper where *PolarMOT* and CenterPoint [6] are trained on one city and evaluated on the other. For evaluations of our method, we use detections produced by the corresponding CP model to make sure each pair of trackers uses the same set of detections.

These extended evaluations along with our code, models and experimental data will be published at `polarmot.github.io` **.**

References

1. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuScenes: A multimodal dataset for autonomous driving. In: CVPR (2020) 1, 4, 5
2. Chiu, H.k., Prioletti, A., Li, J., Bohg, J.: Probabilistic 3d multi-object tracking for autonomous driving. In: ICRA (2021) 4
3. Kim, A., Ošep, A., Leal-Taixé, L.: Eagermot: 3d multi-object tracking via sensor fusion. In: ICRA (2021) 4
4. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: ICCV (2017) 1
5. Maas, A.L., Hannun, A.Y., Ng, A.Y., et al.: Rectifier nonlinearities improve neural network acoustic models. In: Proc. icml. vol. 30, p. 3. Citeseer (2013) 2
6. Yin, T., Zhou, X., Krähenbühl, P.: Center-based 3d object detection and tracking. In: CVPR (2021) 4, 5
7. Zaech, J.N., Liniger, A., Dai, D., Danelljan, M., Van Gool, L.: Learnable online graph representations for 3d multi-object tracking. IEEE R-AL (2022) 4