Chair of Robotics, Artificial Intelligence and Real-time Systems
Department of Informatics
Technical University of Munich

Spatio-Temporal Graph Neural Networks for Multiple Object Tracking

Final Presentation, Master's Thesis

Maximilian Listl

June 24, 2022

Supervisor: *Emeç Ercelik, M.Sc.*
Examiner: *Prof. Dr.-Ing. habil. Alois C. Knoll*

Table of Content

What is Multiple Object Tracking?

Tracking-by-Detections paradigm:

- Given: Object detections $\mathcal{O} = \{o_1, ..., o_n\}$ are given by object detection method
- Goal:
  - Associated detections to determine the trajectory of objects over time.
  - Tracklet $T_i = \{o_{i_1}, ..., o_{i_{n_i}}\}$
  - Find a set of tracklets $\mathcal{T}_* = \{T_1, \ldots, T_m\}$ which explains the detections in a coherent way.

What is Multiple Object Tracking?

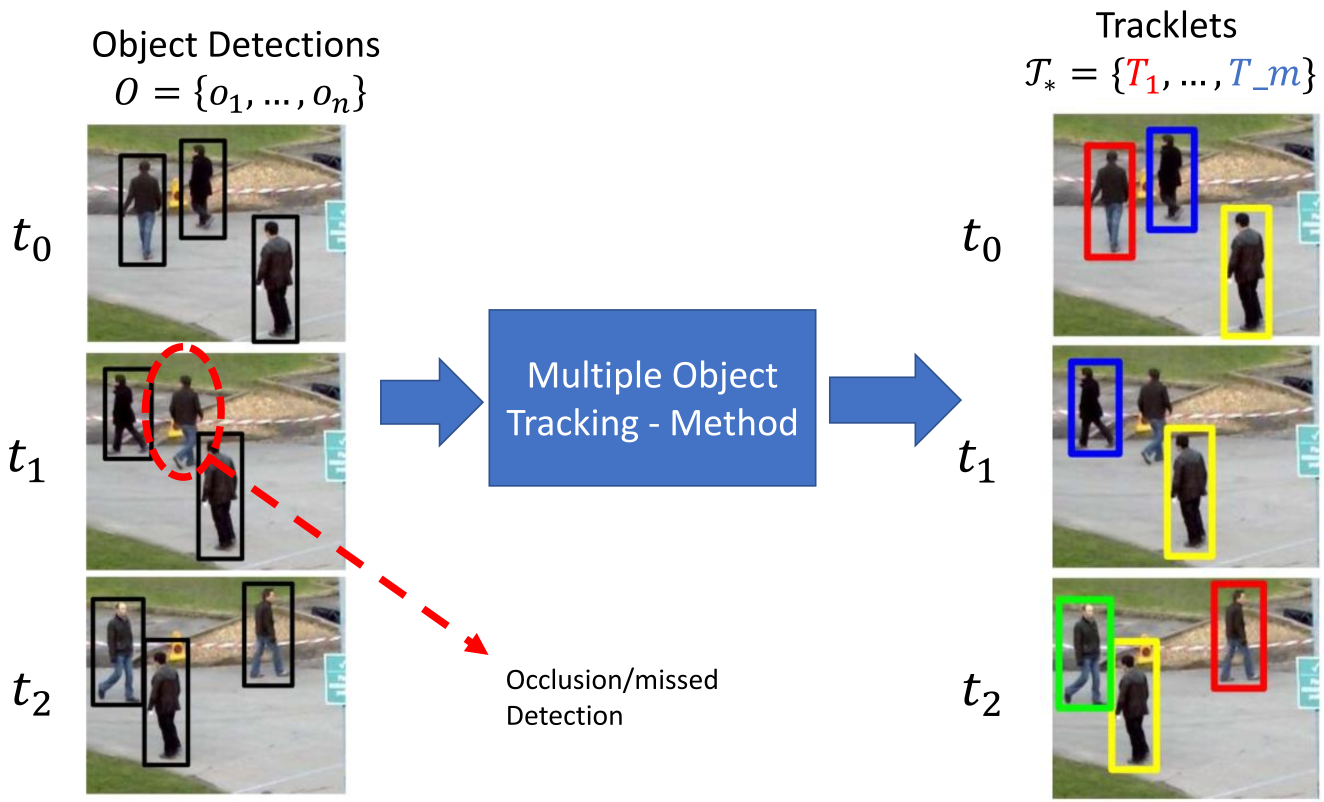


Figure: Visualization of multiple object tracking. Images from [BL20].

Common Approaches

- Kalman Filters and Association Metrics:

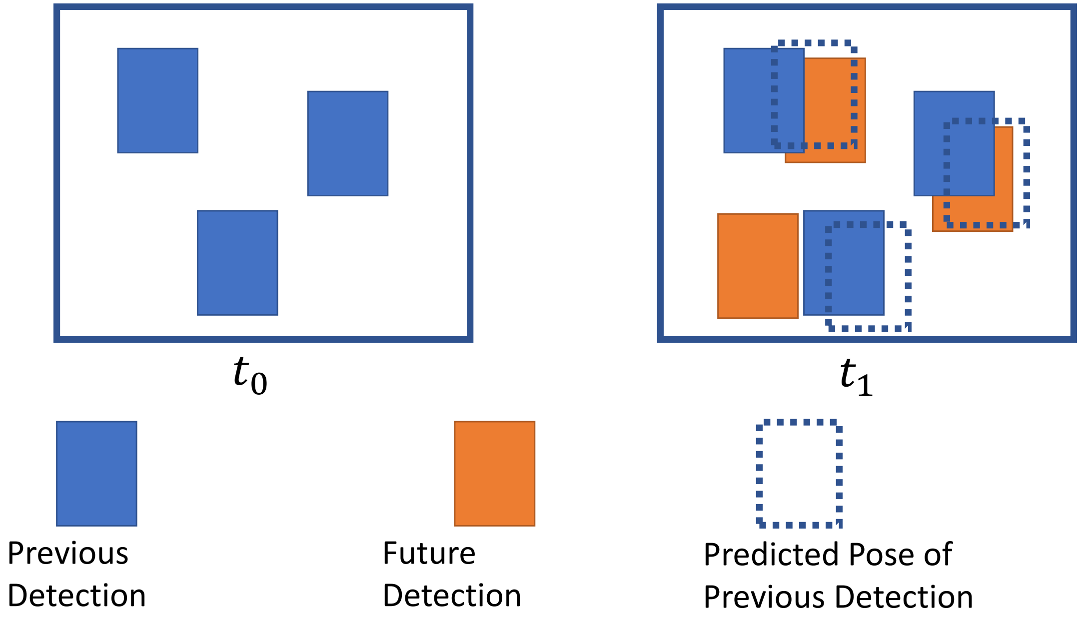
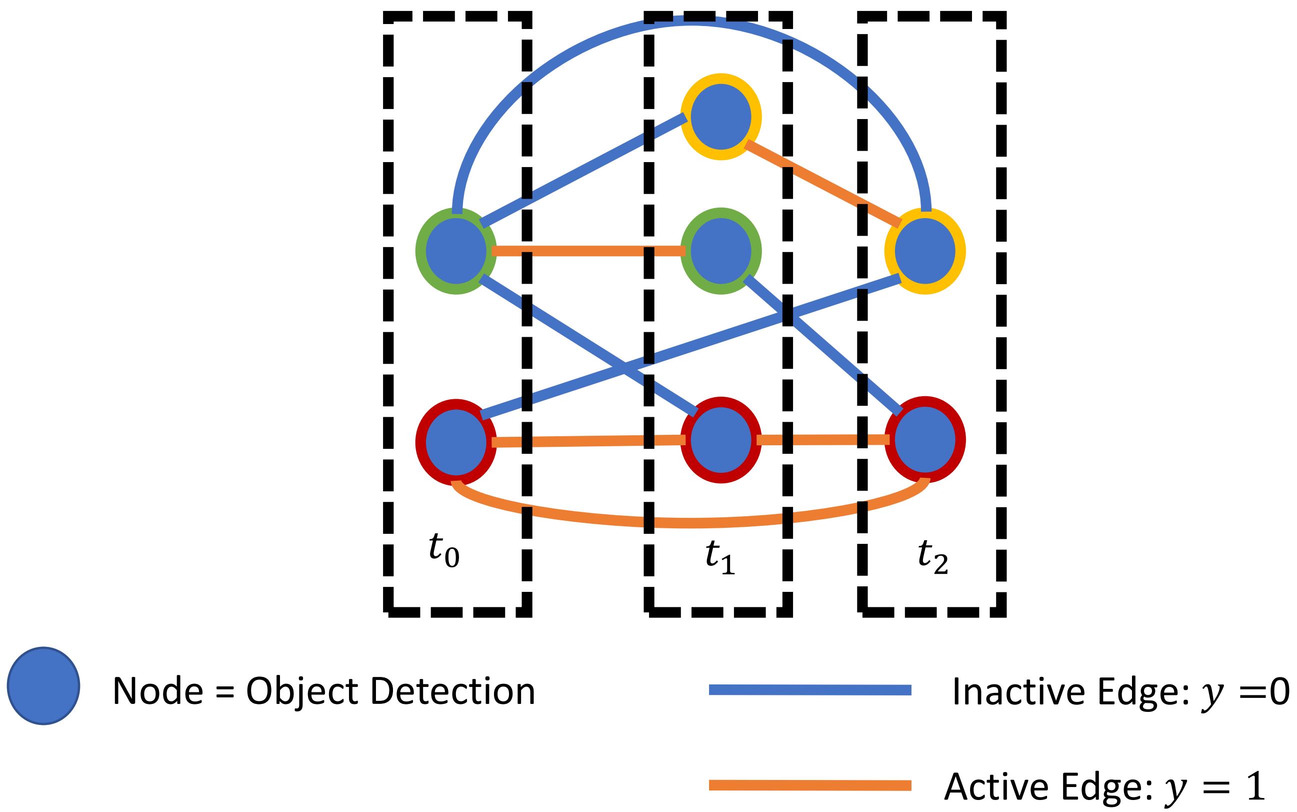


Figure: Visualization of multiple object tracking with Kalman Filters

Multiple Object Tracking - Graph representation

Direct Matching over Edge Classification: 2D MOT [BL20], 3D MOT [Zae+22]

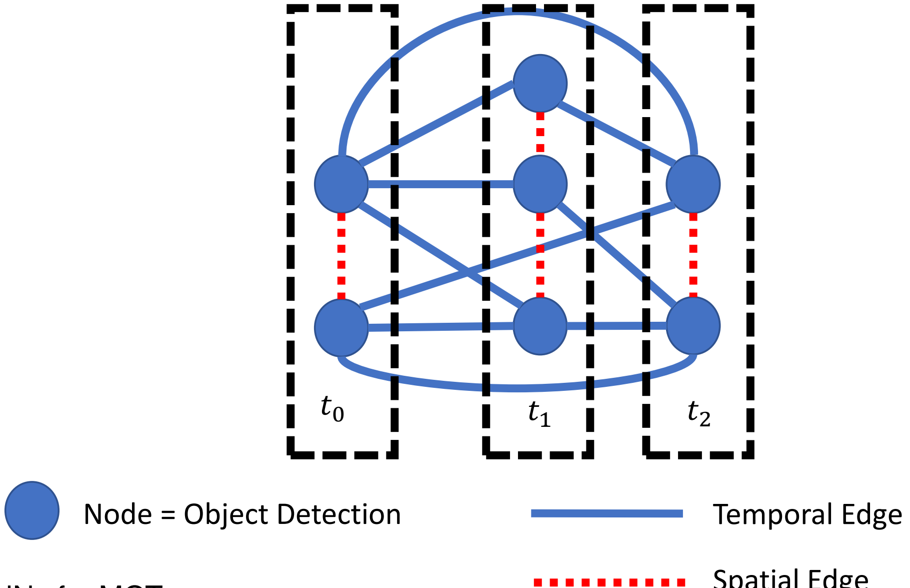


Motivation

Our Idea: Spatio-Temporal Approach for 3D MOT

Take spatial relationships between object detections into account to leverage tracking performance (include spatial information)

Allow matching between object detections across multiple time frames (multi-frame approach)



Method

Pipeline

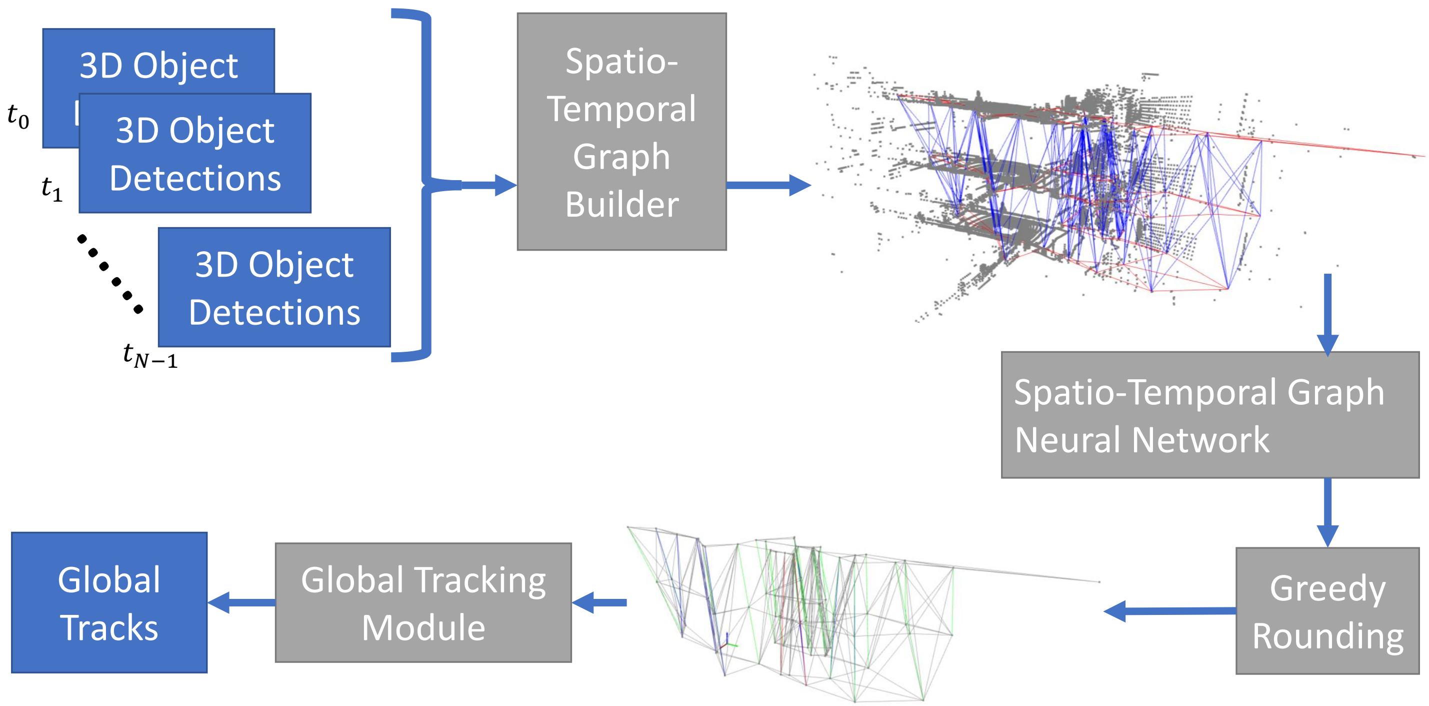
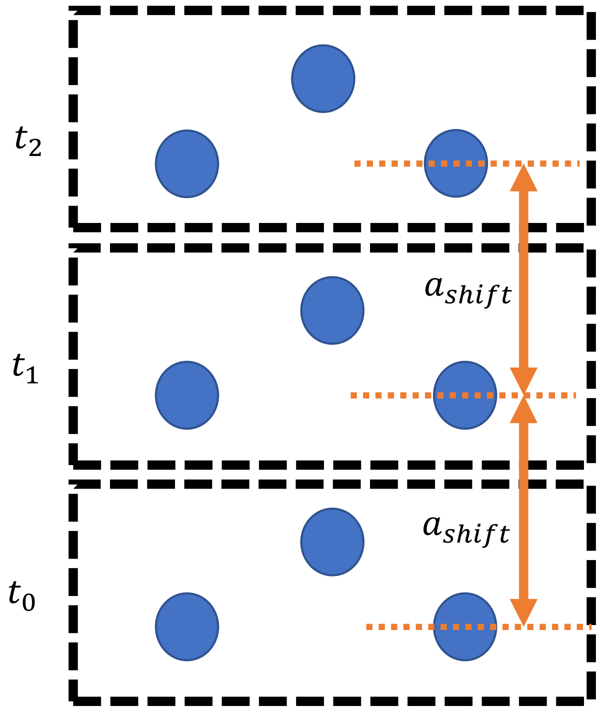
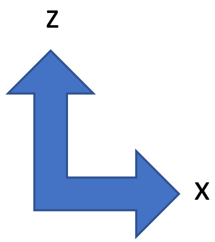


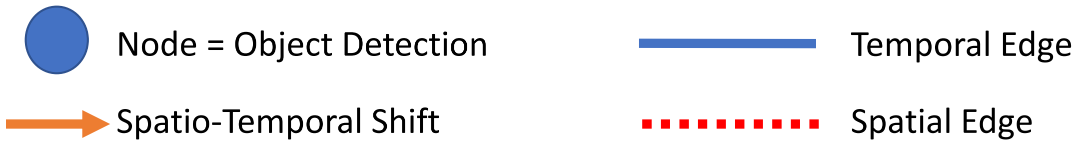
Figure: Visualization of pipeline.

Spatio-Temporal Graph Builder

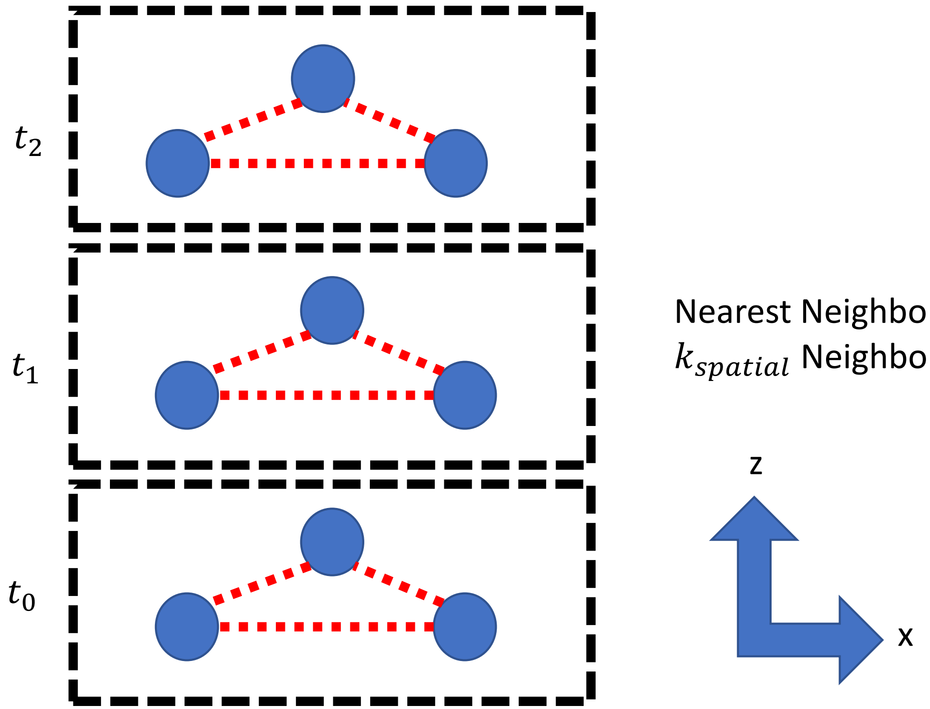


$$x_i' = x_i + \begin{bmatrix} 0 \\ 0 \\ a_{shift} * t_i \end{bmatrix}$$

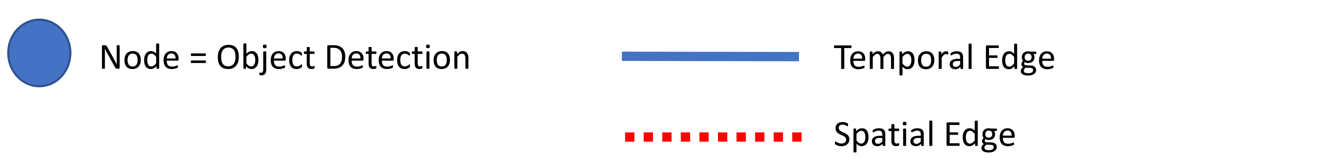




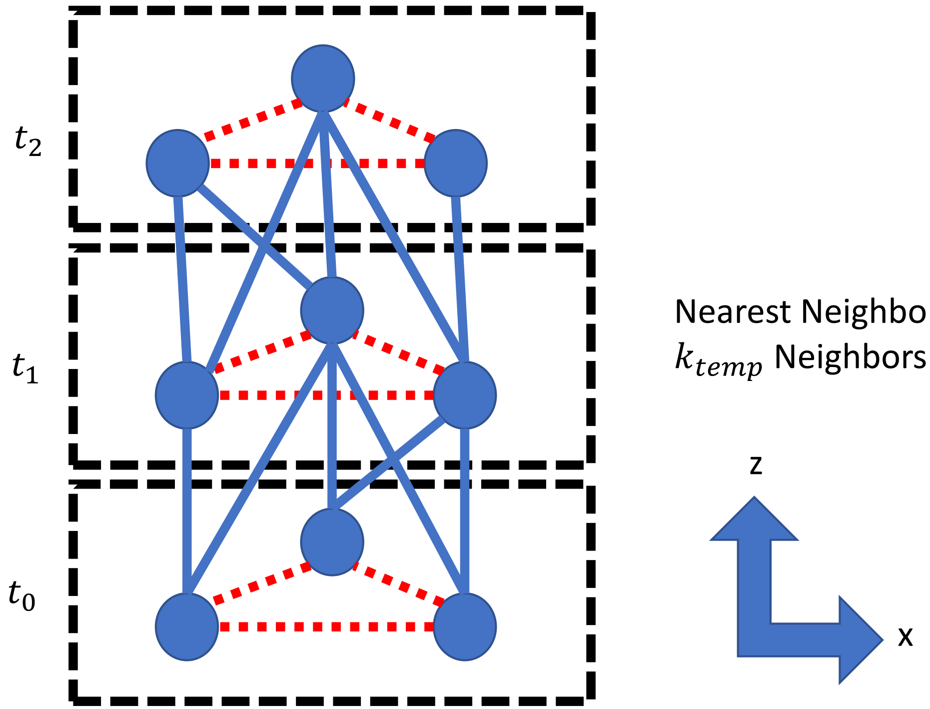
Spatio-Temporal Graph Builder



Nearest Neighbor Method
$k_{spatial}$ Neighbors



Spatio-Temporal Graph Builder



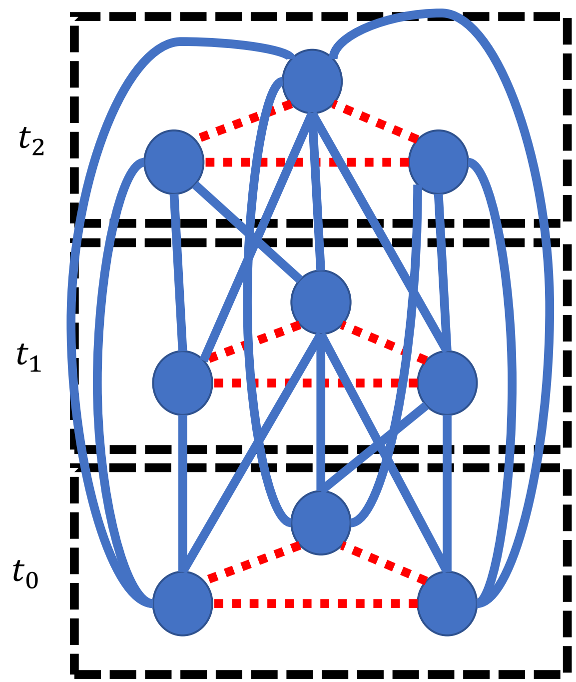
Nearest Neighbor Method
k_{temp} Neighbors

Node = Object Detection

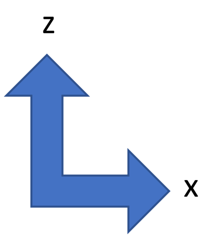
Temporal Edge

Spatial Edge

Spatio-Temporal Graph Builder



- Maximum Temporal Edge Length
- With $\beta \geq (t_j - t_i)$ and $t_i < t_j$



Node = Object Detection

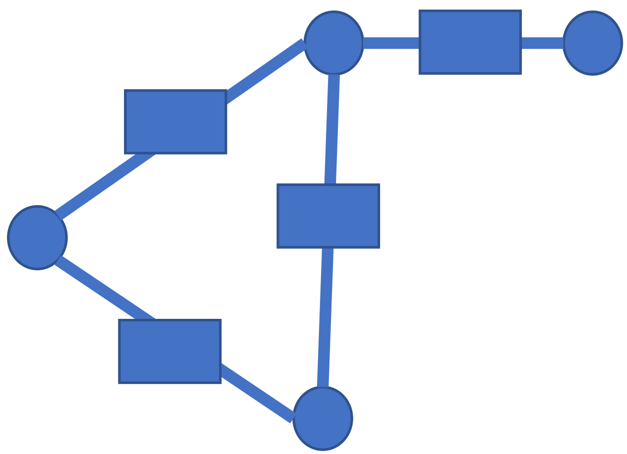
Temporal Edge

Spatial Edge

Method - Message Passing Network

Initial Graph

- Initial node features $h_i^{(0)}$ for node i
- Initial edge features $h_{(i,j)}^{(0)}$ for edge i,j



Initial Node Feature: $h_i^{(0)}$

Initial Edge Feature: $h_{(i,j)}^{(0)}$

Method - Message Passing Network

Edge Update:

$$h_{(i,j)}^{(l)} = \mathcal{N}_e\left(\left[h_i^{(l-1)}, h_j^{(l-1)}, h_{(i,j)}^{(l-1)}\right]\right) \quad (1)$$

Node Update:

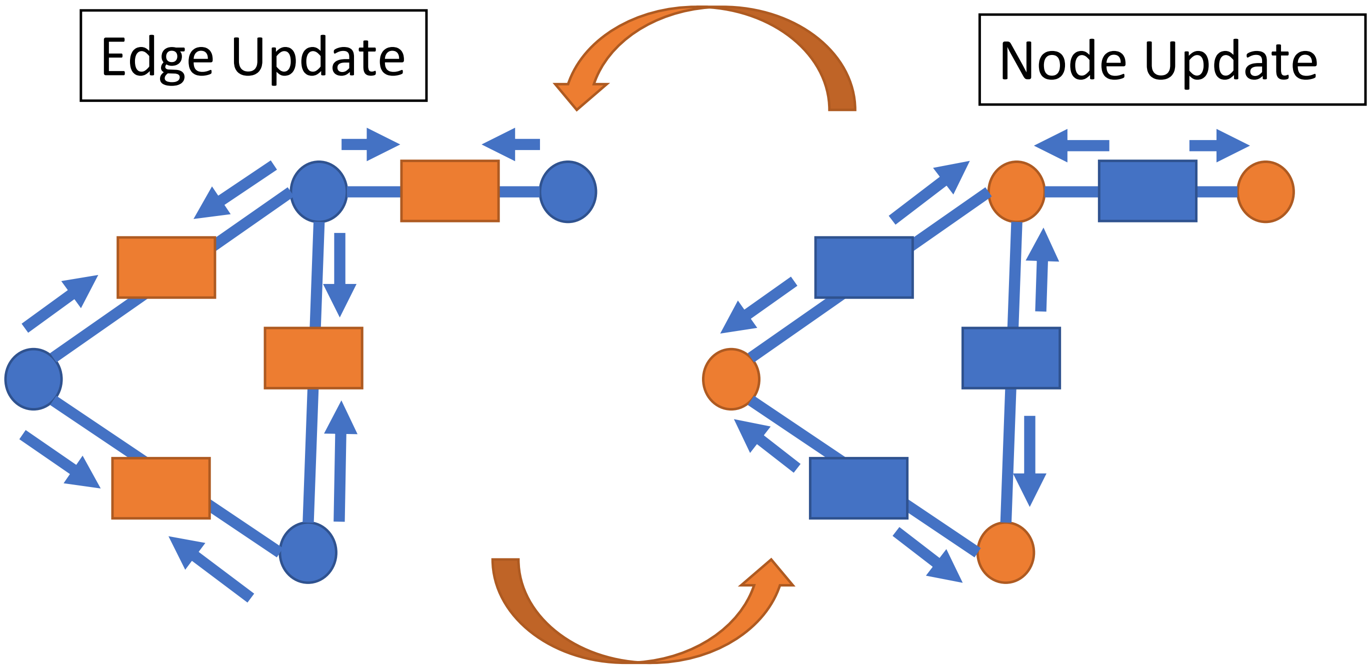
Computing Neural Messages:

$$m_{(i,j)}^{(l)} = \mathcal{N}_v\left(\left[h_i^{(l-1)}, h_{(i,j)}^{(l-1)}\right]\right) \quad (2)$$

Aggregating Messages:

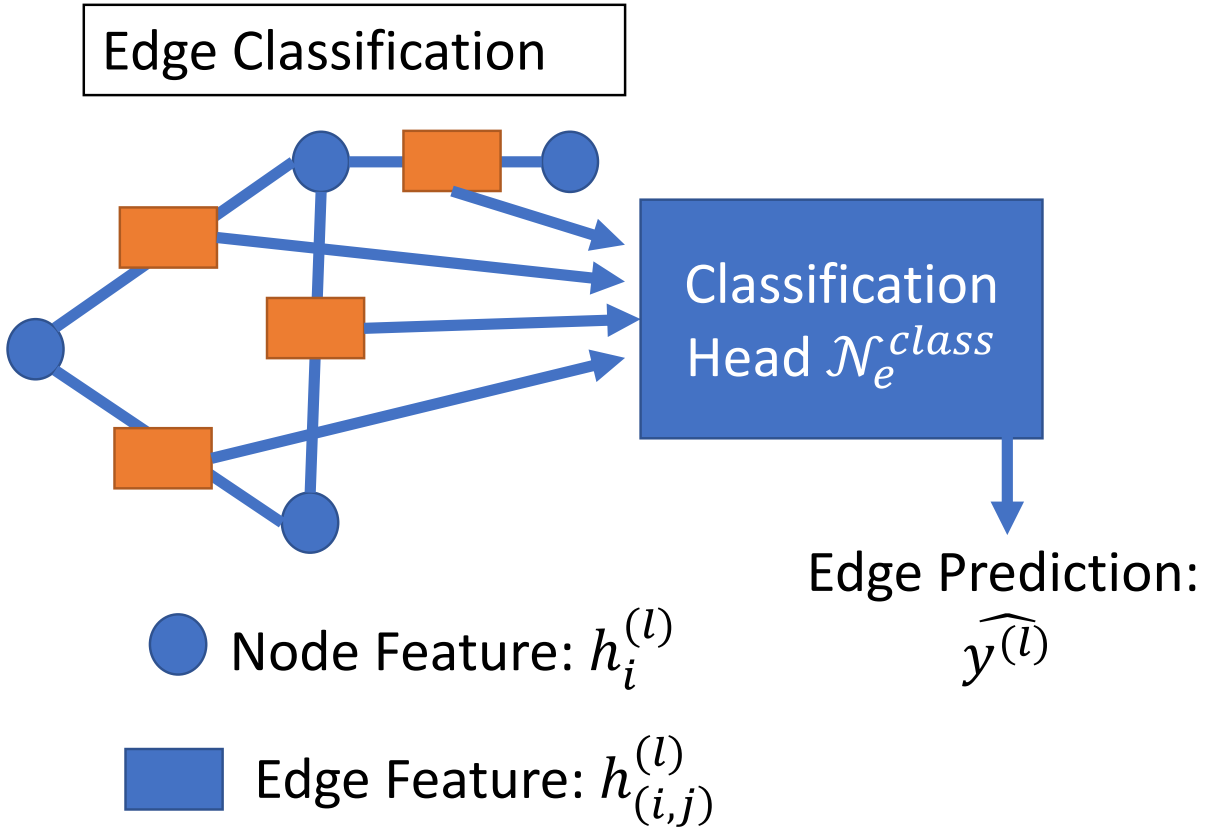
$$h_i^{(l)} = \Phi\left(\left\{m_{(i,j)}^{(l)}\right\}_{j\in\mathbb{N}_i}\right) \quad (3)$$

Method - Message Passing Network



Node Feature: $h_i^{(l)}$ Edge Feature: $h_{(i,j)}^{(l)}$

Method - Message Passing Network



Method

Greedy Rounding

Why?:

GNN can assign more than one incoming or outgoing active edges for each node i.

Adopt method from [BL20].

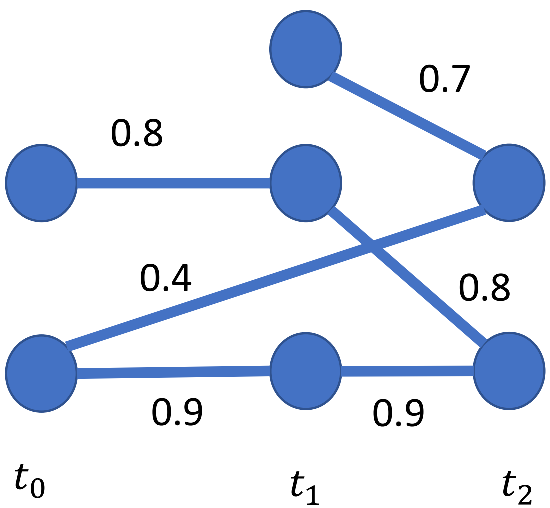


Figure: Example of inconsistent assignment of active edges

Method

Greedy Rounding

Set all edges with $\hat{y}^{(l)}_{(i,j)} < 0.5$ inactive

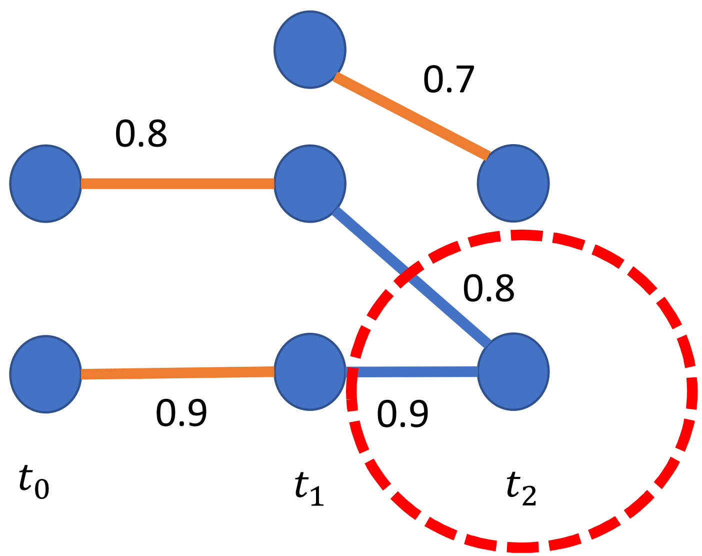


Figure: Example of first rounding step

Method

Greedy Rounding

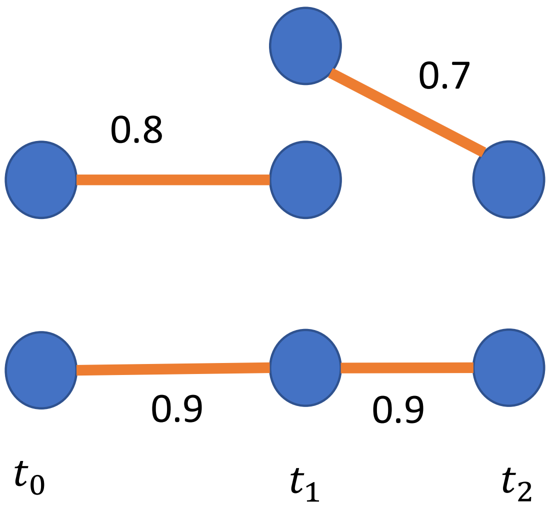


Figure: Example of second rounding step

Method

Greedy Rounding

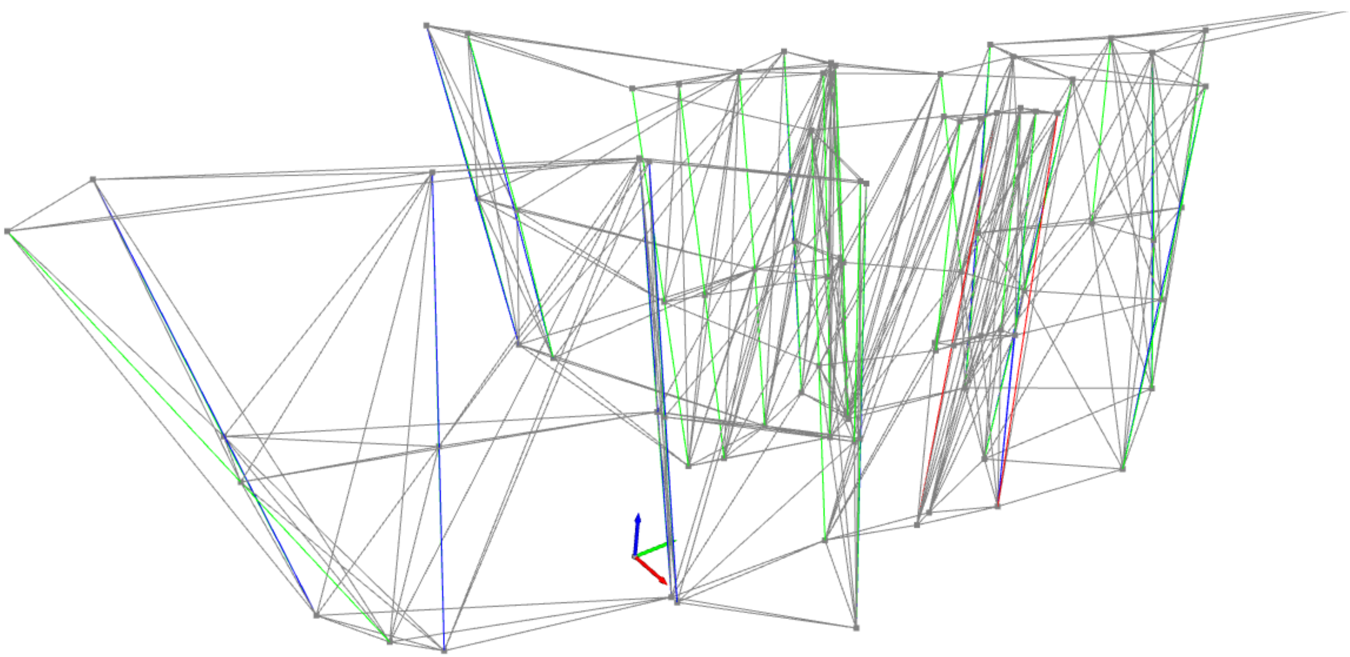


Figure: Resulting prediction after greedy rounding

Method

Tracker Routine

- Given: Local tracking IDs for each node in one graph
- Goal: Global tracking IDs to for each node within a whole scene

Method

Tracker Routine

- Solution: Window shifting

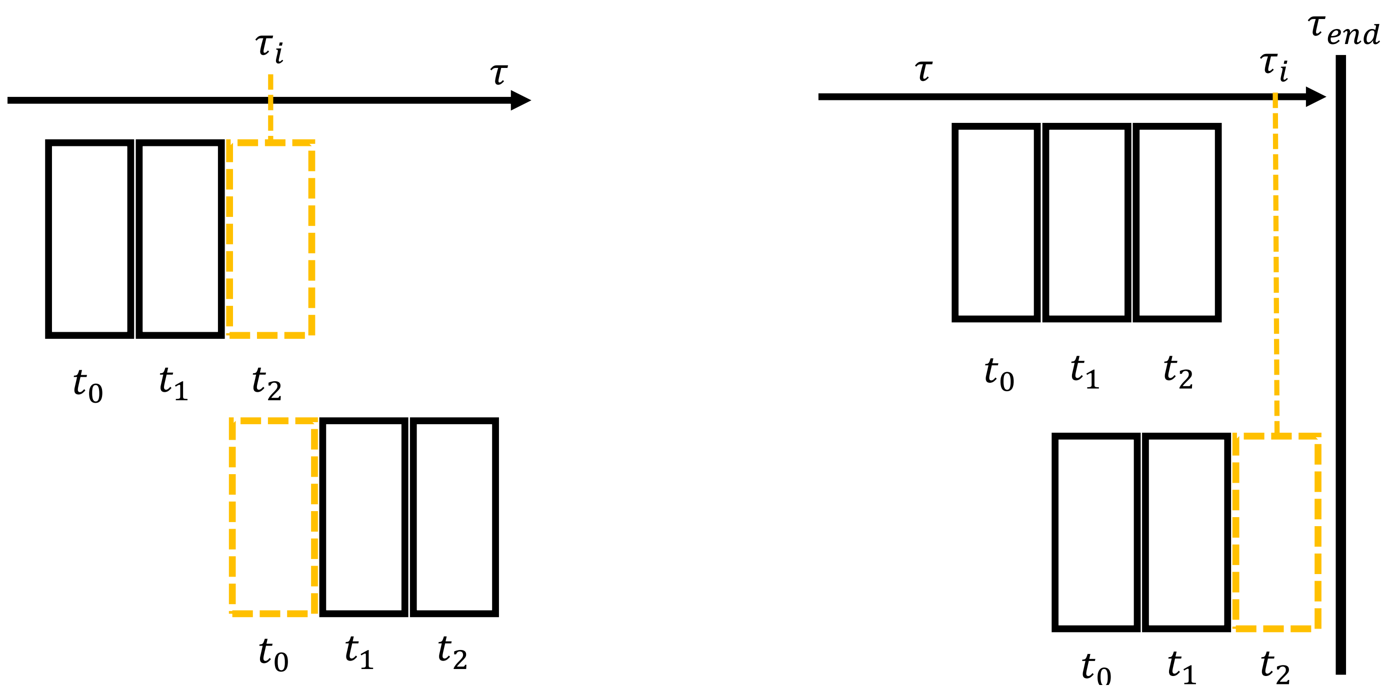


Figure: Visualization of window shifting of tracker.

Experiments - Implementation Details

Nuscenes Dataset [Cae+20]:

- LIDAR measurements, 20Hz capture frequency, 32 channels
- 1,000 scenes, each scenes is 20 s long
- 23 object classes with accurate 3D bounding boxes at 2Hz
- Tracking Challenge:
  - cars, pedestrians, buses, motorcycles, trailers, trucks, bicycles

Experiments - Implementation Details

Initial Features

$$h_i^{(0)} = \left[x_i', t_i\right] \in \mathrm{R}^4 \quad (4)$$

$$h_{(i,j)}^{(0)} = \left[q_{i,j}, q_{i,j}\right] \in \mathrm{R}^2 \quad \text{with} \quad (5)$$

$$q_{(i,j)} = \begin{cases} 0, & \text{if} \quad (i,j) \in E_{spatial} \\ 1, & \text{if} \quad (i,j) \in E_{temp} \end{cases} \quad (6)$$

Experiments - MOT Metrics

MOT metrics introduced by [Wen+20]:

- Average multiple object tracking accuracy (AMOTA)

$$MOTA_r = max(0, 1 - \frac{FN_r + FP_r + IDS_r - (1-r) \cdot P}{r \cdot P}) \tag{7}$$

$$AMOTA = \frac{1}{n-1} \sum_{r \in \{\frac{1}{n-1}, \frac{2}{n-1}, \dots, 1\}} MOTA_r \tag{8}$$

- Average multiple object tracking precision (AMOTP)

$$MOTP = \frac{\sum_{i,t} d_{i,t}}{\sum_t TP_t} \tag{9}$$

$$AMOTP = \frac{1}{n-1} \sum_{r \in \{\frac{1}{n-1}, \frac{2}{n-1}, \dots, 1\}} MOTP \tag{10}$$

Experiments - Feasibility Study

- Does our tracking approach work theoretically?
- Perform a sanity check:
  - Ground truth (GT) annotations as detections (Validation Set)
  - GT edge labels $y = \hat{y}$ as edge predictions (Do not use GNN for edge classification)

Experiments - Feasibility Study

Analyze tracking performance under influence of:

- number of frames per graph N
- the temporal KNN parameter k_{temp}
- maximum temporal edge length β

Experiments - Feasibility Study

Influence of Temporal Edges

k_{temp}	AMOTA↑	AMOTP↓	RECALL↑	TP↑	FP↓	FN↓	IDS↓
3	0.9201	0.1268	0.9954	96,347	2,729	738	4,812
6	0.9487	0.0699	0.9954	99,766	2,717	738	1,393
12	**0.9558**	**0.0558**	0.9954	**100,780**	**2,714**	**737**	**380**

Table: MOT-results after varying k_{temp}. Using GT annotations from Nuscenes validation split and GT edge labels as predictions.

- increased k_{temp} leads to higher connectivity.
- higher likelihood to connect nodes from same trajectory.

Experiments - Learned Tracker Results

- Perform MOT on GNN
  - Ground truth (GT) annotations as detections (Validation Set)

Experiments - Learned Tracker Results

Influence of Spatial Edges

$k_{spatial}$	AMOTA↑	AMOTP↓	RECALL↑	TP↑	FP↓	FN↓	IDS↓
3	0.8581	0.2244	0.9933	**16.702**	92,278	5,024	8,752
6	**0.8657**	**0.2149**	0.9933	14.803	92,392	4,484	**8,646**
12	0.8617	0.2239	0.9933	13.711	**91,622**	**4,194**	9,423

Table: MOT-results after varying $k_{spatial}$. Using GT annotations from Nuscenes validation split but GNN edge predictions.

- Increase in $k_{spatial}$ improves AMOTA and AMOTP
- However, effect is small, probably due to small feature space of edge embeddings
- Shows that spatial information help to leverage MOT performance

Experiments - Learned Tracker Results

Influence of Temporal Edges and Spatial Edges

k_{temp}	$k_{spatial}$	AMOTA↑	AMOTP↓	RECALL↑	TP↑	FP↓	FN↓	IDS↓
3	3	0.8581	0.2244	0.9933	92,278	5,024	867	8,752
6	3	0.8972	0.1609	0.9933	94,814	4,074	848	6,235
12	3	0.9060	0.1489	0.9934	96,454	3,456	820	4,623
6	6	0.9090	0.1418	**0.9952**	96,157	3,673	**811**	4,929
12	6	**0.9195**	**0.1244**	0.9950	**96,464**	**3,222**	845	**4,588**

Table: MOT-results after varying jointly k_{temp} and $k_{spatial}$. Using GT annotations from Nuscenes validation split but GNN edge predictions.

- Combined increase of $k_{spatial}$ and k_{temp} leads to best performing configuration.
- Higher temporal connectivity increases likelihood of connecting nodes which belong to the same trajectory.
- Higher spatial connectivity increases awareness of spatial context for each node

Experiments - Learned Tracker Results

Benchmark

How well performs our method against other methods?

- Perform Benchmark
- Use Centerpoint Detections [YZK21] (Nuscenes - validation set)
- Comparison with CBMOT [BSZ21] and EagerMot [KOL21]

Experiments - Learned Tracker Results

Benchmark

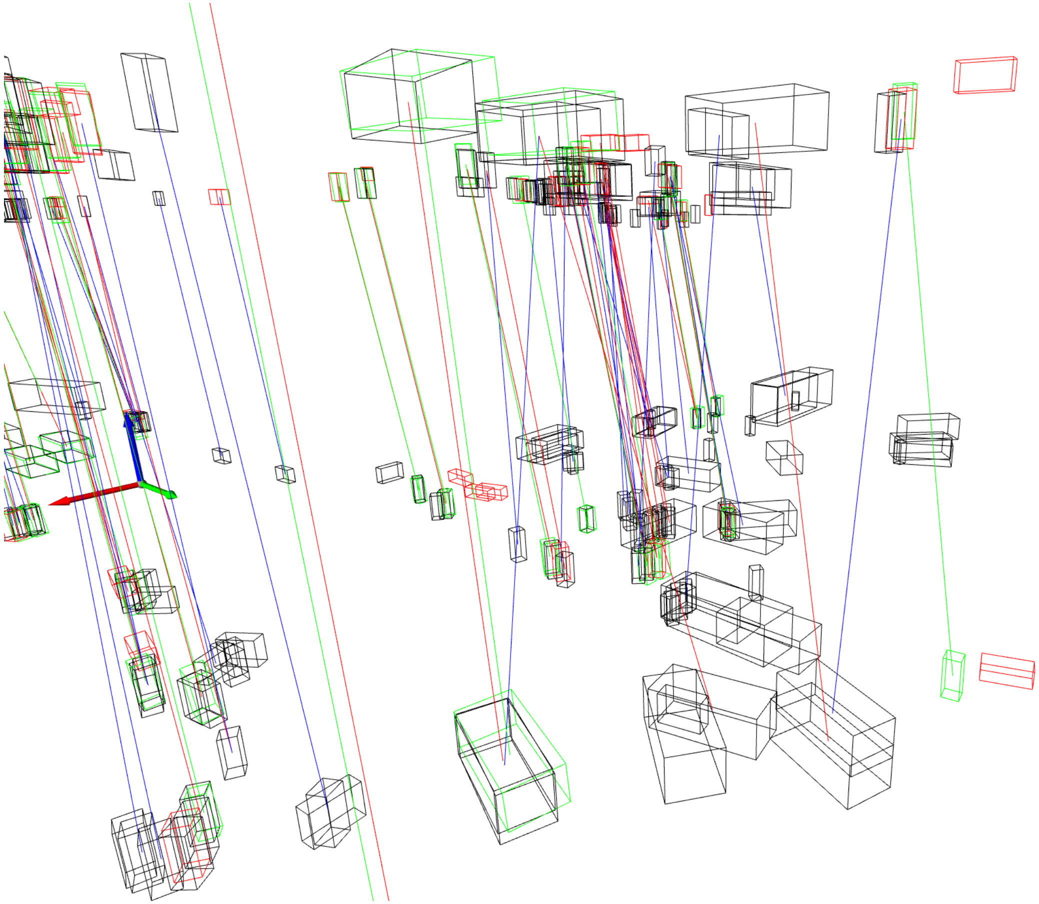
Method	k_{temp}	$k_{spatial}$	AMOTA↑	AMOTP↓	RECALL↑
CBMOT* [BSZ21]	-	-	**0.7219**	0.5337	**0.73784300**
CBMOT^ [BSZ21]	-	-	0.7196	**0.4869**	0.73473442
ours	3	3	0.2101	1.1948	0.42755481
ours	12	6	0.2021	1.3185	0.43722993

Table: MOT-results comparing our method with CBMOT[BSZ21] (Part 1). Using CenterPoint detections (centerpoint_voxel_1440) [YZK21] from Nuscenes validation split. CBMOT* uses 2D and 3D tracklets and uses a multiplication for its score update. CBMOT^ also uses 2D and 3D tracklets. However, it uses a neural network for its score update.

- Our method performs worse than CBMOT, given the Centerpoint detections.
- Probably caused by our methods inability to filter out false positive detections

Experiments - Learned Tracker Results

Comparison with CBMOT



Experiments - Learned Tracker Results

Benchmark

Methods	k_{temp}	$k_{spatial}$	AMOTA↑	AMOTP↓	RECALL↑
EagerMot*	-	-	**0.7145**	**0.5664**	**0.76163287**
EagerMot^	-	-	0.7120	0.5690	0.75200000
Ours	3	3	0.2328	1.1631	0.45707547
Ours	12	6	0.2145	1.2831	0.44252695

Table: MOT-results comparing our method with EagerMOT[KOL21] (Part 1). Using CenterPoint detections (centerpoint_voxel_1440_dcn(flip)) [YZK21] from Nuscenes validation split. EagerMOT* represents the results evaluated by our own computation. EagerMOT^ represents the results from the paper.

Conclusion

1. Our approach work well for detection inputs without any false positive object detections (ground truth annotations)

2. Spatio-temporal graph structure allows association between nodes over time, while including the spatial context for edge classification

3. However, our method cannot handle false positive object detections. Therefore it fails to perform on state-of-the-art levels.

Future Work

1. Additional strategy for handling false positive detections:
   1.1 Classification of nodes, as in [Zae+22]
   1.2 Randomly add false positive nodes to graph (augmentation method)
2. Random removal of nodes, while training. Improve Robustness of GNN towards occlusions. (augmentation method)
3. Graph construction: Increase robustness by including appearance features and orientation features to K-Nearest Neighbor methods
4. Initial feature selection:
   4.1 $h_i^{(0)}$ with appearance features and class-id
   4.2 $h_{(i,j)}^{(0)}$ contains difference in bounding box dimensions

References I

[BSZ21] Benbarka, N., Schroder, J., and Zell, A. "Score refinement for confidence-based 3D multi-object tracking". In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 27.09.2021 - 01.10.2021, pp. 8083–8090. ISBN: 978-1-6654-1714-3. DOI: 10.1109/IROS51168.2021.9636032.

[BL20] Braso, G. and Leal-Taixe, L. "Learning a Neural Solver for Multiple Object Tracking". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.

[Cae+20] Caesar, H., Bankiti, V., Lang, A. H., Vora, S., Liong, V. E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., and Beijbom, O. "nuScenes: A multimodal dataset for autonomous driving". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020. URL: http://arxiv.org/pdf/1903.11027v5.

[KOL21] Kim, A., Ošep, A., and Leal-Taixé, L. "EagerMOT: 3D Multi-Object Tracking via Sensor Fusion". In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021. DOI: 10.1109/ICRA48506.2021.9562072. URL: http://arxiv.org/pdf/2104.14682v1.

[Wen+20] Weng, X., Wang, Y., Man, Y., and Kitani, K. "GNN3DMOT: Graph Neural Network for 3D Multi-Object Tracking with Multi-Feature Learning". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2020. ISBN: 978-1-7281-7168-5. URL: http://arxiv.org/pdf/2006.07327v1.

References II

[YZK21] Yin, T., Zhou, X., and Krahenbuhl, P. “Center-based 3D Object Detection and Tracking”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 11784–11793.

[Zae+22] Zaech, J.-N., Liniger, A., Dai, D., Danelljan, M., and van Gool, L. “Learnable Online Graph Representations for 3D Multi-Object Tracking”. In: *IEEE Robotics and Automation Letters* (2022), p. 1. DOI: 10.1109/LRA.2022.3145952.

Chair of Robotics, Artificial Intelligence and Real-time Systems
Department of Informatics
Technical University of Munich

Spatio-Temporal Graph Neural Networks for Multiple Object Tracking

Final Presentation, Master's Thesis

Maximilian Listl

June 24, 2022

Supervisor: *Emeç Ercelik, M.Sc.*
Examiner: *Prof. Dr.-Ing. habil. Alois C. Knoll*

Motivation

Graph Neural Networks - based

- Feature fusion for learned association metric [Wen+20]
- Direct Matching over Edge Classification [Zae+22]

Method

Spatio-Temporal Graph Builder

Spatio-Temporal Shift:

constant shift value a_{shift}

$$x_i' = x_i + \begin{bmatrix} 0 \\ 0 \\ a_{shift} \cdot t_i \end{bmatrix} \in \mathbb{R}^3 \tag{11}$$

with $\quad t_i = \{0, ..., N-1\}$

Spatial Edge Construction:

K-Nearest Neighbor with $k_{spatial}$

Temporal Edge Construction:

K-Nearest Neighbor with k_{temp}

Connect time frames with a time difference of up to β (maximum temporal edge length)

Method

Spatio-Temporal Graph Builder

Algorithm 1 Algorithm to build temporal edges

Require: $\beta \geq (t_j - t_i)$ and $t_i < t_j$

Given: k_{temp}

Given: $X'_{t_i} \leftarrow \left\{x'_{t_{i_1}}, ..., x'_{t_{i_p}}\right\}$

Given: $X'_{t_j} \leftarrow \left\{x'_{t_{j_1}}, ..., x'_{t_{j_q}}\right\}$

$A \leftarrow \{\}$ ▷ A is the empty set initially

for $x'_{t_{i_g}} \in X'_{t_i}$ **do**

$a_i \leftarrow knn\_method(X'_{t_j}, x'_{t_{i_g}}, k_{temp})$ ▷ a_i contains index pairs of the k_{temp} neighbors of detection o_i

$A \leftarrow A \cup a_i$

end for

return A ▷ Contains the index pairs of all k_{temp} neighbors for all detection from O_{t_i}. They represent the temporal edges.

Figure: Input Graph

Method

Message Passing Network

We adopt the time-aware node update step from [BL20]

Node Update:

Computing Neural Messages:

$$m_{(i,j)}^{(l)} = \begin{cases} \mathcal{N}_v^{past}\left(\left[h_i^{(l-1)}, h_{(i,j)}^{(l-1)}, h_i^{(0)}\right]\right), & \text{if } j \in \left\{\mathbb{N}_{i,temp}^{past} \cup \mathbb{N}_{i,spatial}^{flow\_in}\right\} \\ \mathcal{N}_v^{fut}\left(\left[h_i^{(l-1)}, h_{(i,j)}^{(l-1)}, h_i^{(0)}\right]\right), & \text{if } j \in \left\{\mathbb{N}_{i,temp}^{fut} \cup \mathbb{N}_{i,spatial}^{flow\_out}\right\} \end{cases} \tag{12}$$

Method

Message Passing Network

We adopt the time-aware node update step from [BL20]

Node Update:

Aggregating Messages:

$$h_{i,past}^{(l)} = \sum_{j \in \left\{ \mathbb{N}_{i,temp}^{past} \cup \mathbb{N}_{i,spatial}^{flow\_in} \right\}} m_{(i,j)}^{(l)} \tag{13}$$

$$h_{i,fut}^{(l)} = \sum_{j \in \left\{ \mathbb{N}_{i,temp}^{fut} \cup \mathbb{N}_{i,spatial}^{flow\_out} \right\}} m_{(i,j)}^{(l)} \tag{14}$$

$$h_{i}^{(l)} = \mathcal{N}_{v}\left(\left[h_{i,past}^{(l)}, h_{i,fut}^{(l)}\right]\right) \tag{15}$$

Method

Message Passing Network

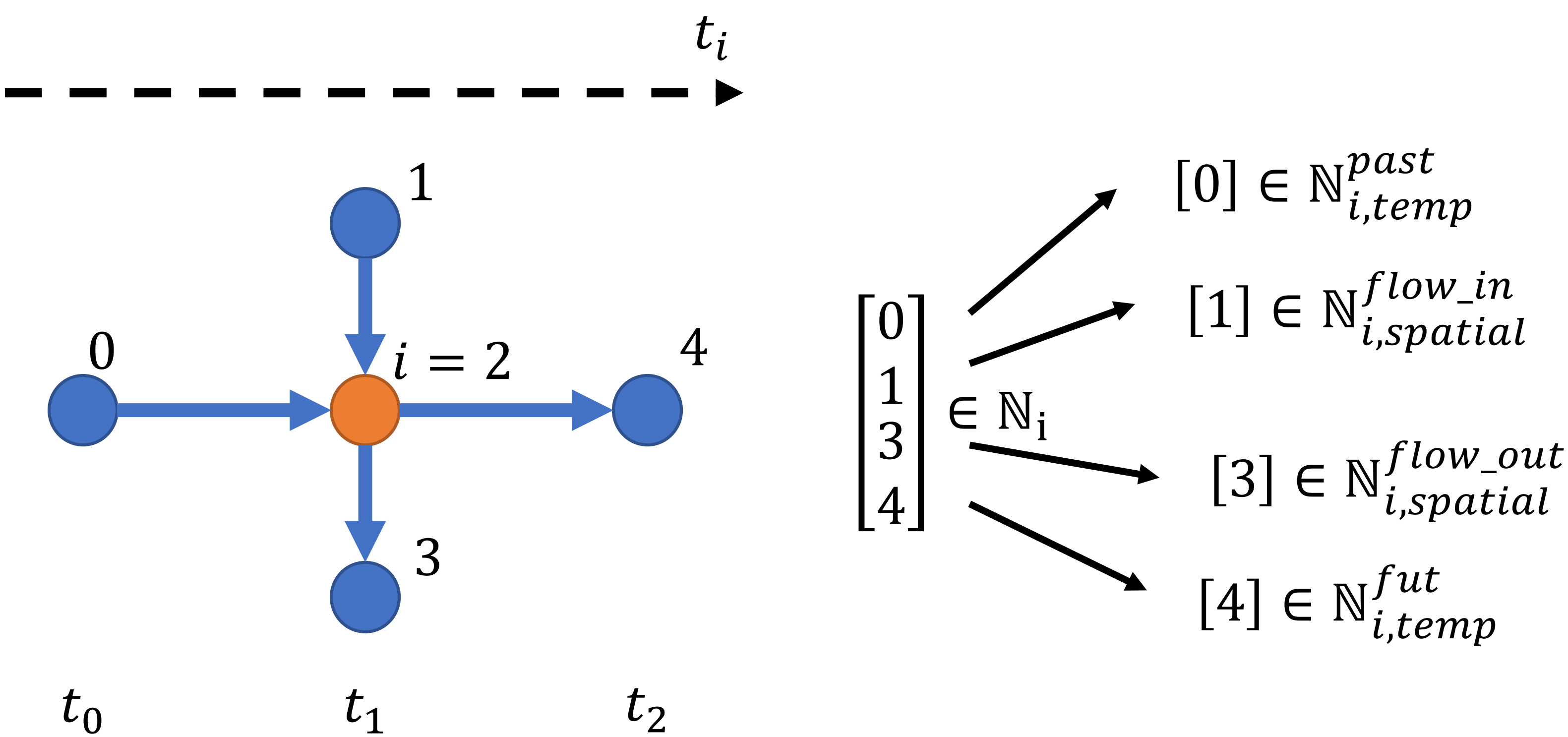


Figure: Visualization of indexing problem. Given node i with $i = 2$ and its adjacent nodes j with $j \in \mathbb{N}_i$.

Method

Message Passing Network

Feature selection:

$$h_i^{(0)} = \left[x_i', t_i\right] \in \mathrm{R}^4 \tag{16}$$

$$h_{(i,j)}^{(0)} = \left[q_{i,j}, q_{i,j}\right] \in \mathrm{R}^2 \tag{17}$$

with

$$q_{(i,j)} = \begin{cases} 0, & \text{if} \quad (i,j) \in E_{spatial} \\ 1, & \text{if} \quad (i,j) \in E_{temp} \end{cases} \tag{18}$$

Method

Message Passing Network

Training and Inference:

Edge classification:

$$\hat{y}_{(i,j)}^{(l)} = \sigma\left(\mathcal{N}_e^{class}\left(h_{(i,j)}^{(l)}\right)\right) \in [0,1] \tag{19}$$

Loss:

$$\mathcal{L} = \frac{-1}{|E|}\sum_{l=l_0}^{L}\sum_{(i,j)\in E} w \cdot y_{(i,j)} \log\left(\hat{y}_{(i,j)}^{(l)}\right) + (1 - y_{(i,j)}) \log\left(1 - \hat{y}_{(i,j)}^{(l)}\right) \tag{20}$$

Chair of Robotics, Artificial Intelligence and Real-time Systems
Department of Informatics
Technical University of Munich

Method

Greedy Rounding

Algorithm 2 Algorithm for Greedy Rounding based on Greedy Projection from [BL20]

Given: Graph $G = (V, E_{temp})$
Given: Edge Predictions $\hat{y}^L$
Return: feasible flow solution y
for $(i,j) \in E_{temp}$ **do**
  if $\hat{y}^L_{(i,j)} > 0.5$ **then**
    $y_{(i,j)} \leftarrow 1$
  else
    $y_{(i,j)} \leftarrow 0$
  end if
end for
for $i \in V$ **do**
  if Constraint 3.4 is violated **then**
    $j* \leftarrow \text{argmax}_{j \in \mathbb{N}^{past}_{i,temp}} \hat{y}^L_{(i,j)}$
    for $j \in \mathbb{N}^{past}_{i,temp} \backslash \{j*\}$ **do**
      $y_{(i,j)} \leftarrow 0$
    end for
  end if
  if Constraint 3.5 is violated **then**
    $j* \leftarrow \text{argmax}_{j \in \mathbb{N}^{fut}_{i,temp}} \hat{y}^L_{(i,j)}$
    for $j \in \mathbb{N}^{fut}_{i,temp} \backslash \{j*\}$ **do**
      $y_{(i,j)} \leftarrow 0$
    end for
  end if
end for

Figure: Greedy rounding method from [BL20]

Method

Tracker Routine

- Given: Local tracking IDs for each node in one graph
- Goal: Global tracking IDs to for each node within a whole scene

Method

Tracker Routine

- Solution: Window shifting

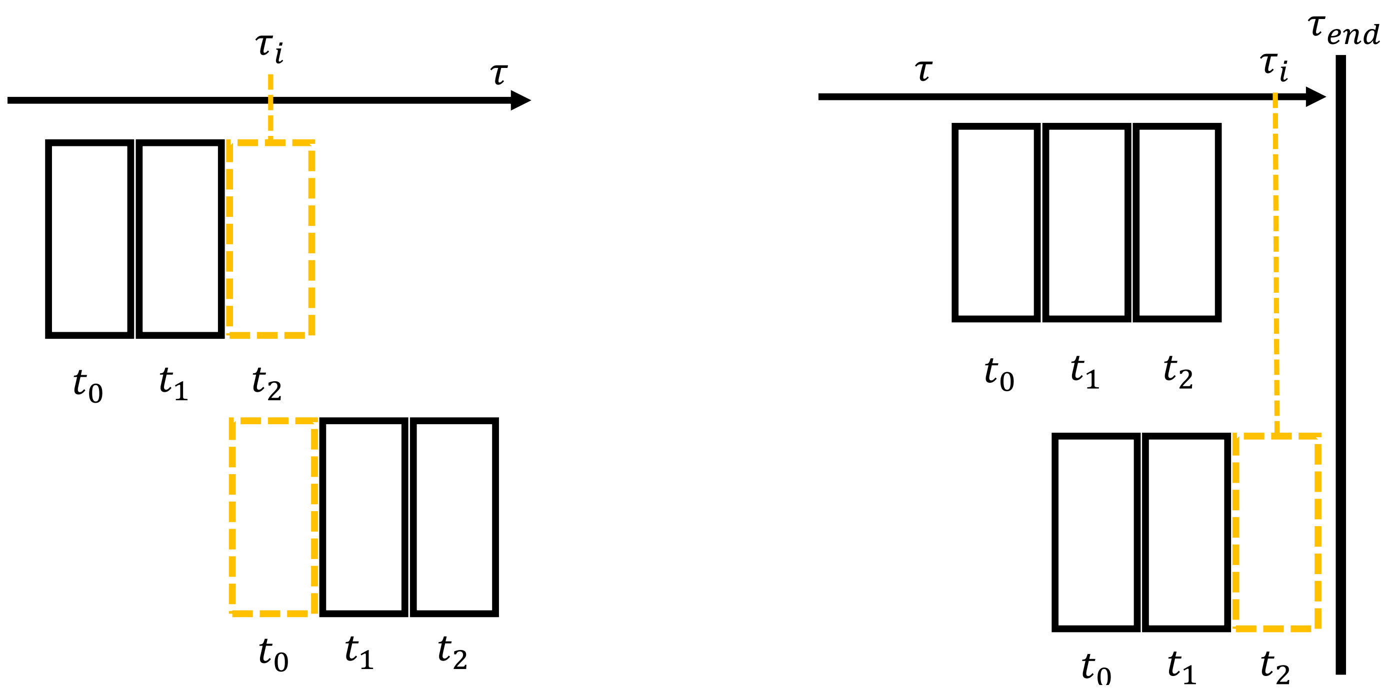


Figure: Visualization of window shifting of tracker.

Experiments

Feasibility Study

Influence of spatial edges:

$k_{spatial}$	AMOTA↑	AMOTP↓	RECALL↑	TP↑	FP↓	FN↓	IDS↓
6	0.9487	0.0699	0.9954	99,766	2,717	738	1,393
3	0.9487	0.0699	0.9954	99,766	2,717	738	1,393

Table: MOT-results after varying the number of frames $k_{spatial}$. Using GT annotations from Nuscenes validation split and GT edge labels as predictions.

Experiments - Feasibility Study

Influence of Frames per Graph

N	AMOTA↑	AMOTP↓	RECALL↑	MOTA↑	TP↑	FP↓	FN↓	IDS↓
3	**0.9201**	**0.1268**	**0.995**	**0.9378**	**96,347**	2,729	**738**	4,812
6	0.9129	0.1411	0.986	0.9294	95,643	2,741	1,506	4,748
12	0.7672	0.4403	0.847	0.7985	82,608	**2,367**	15,048	**4,241**

Table: MOT-results after varying the number of frames N. Using GT annotations from Nuscenes validation split and GT edge labels as predictions.

- Increased N leads to worse AMOTA and AMOTP for constant $k_{temp} = 3$
- Low k_{temp} at high N probably does not allow to connection between nodes belonging to same object

Experiments - Feasibility Study

Influence of Maximum Temporal Edge Length

β	AMOTA↑	AMOTP↓	RECALL↑	TP↑	FP↓	FN↓	IDS↓
2	**0.7672**	0.4403	0.8472	82,608	**2,367**	15,048	4,241
6	0.7668	0.4403	**0.8477**	82,759	2,439	14,957	4,181
11	0.7662	0.4403	**0.8477**	**82,761**	2,464	**14,956**	**4,180**

Table: MOT-results after varying β. Using GT annotations from Nuscenes validation split and GT edge labels as predictions.

- Increased β minimally affects AMOTA and AMOTP for constant $k_{temp} = 3$
- Low k_{temp} at high β probably does not allow to connection between nodes belonging to same object