

Ming Liang\*, Bin Yang\*, Wenyuan Zeng,

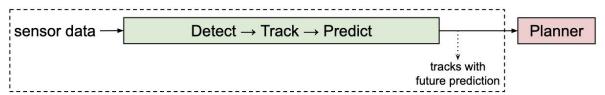
Yun Chen, Rui Hu, Sergio Casas, Raquel Urtasun





## PnPNet: Tracking in-the-loop

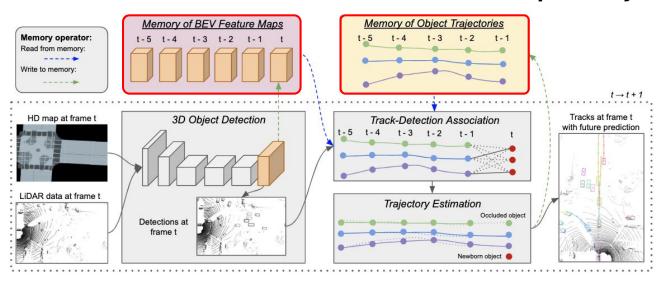
End-to-End Perception & Prediction, tracking in-the-loop



- Performs discrete-continuous tracking between detection and prediction
- Explicit memory for past tracks and their features
- Exploits long history information with a new trajectory representation
- End-to-end optimization of multi-tasks

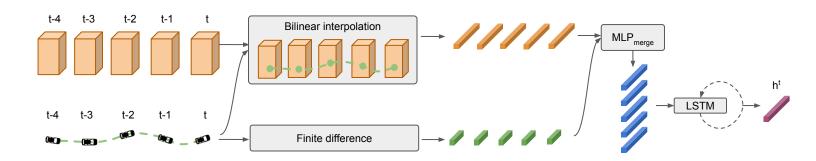
#### Model Architecture

- We start with a joint perception & prediction architecture
- We add a discrete tracker that links detections across time
- We smooth the updated trajectories in continuous space
- We perform motion forecasting from trajectory-level object feature
- The model runs in a recurrent fashion, and memorizes past trajectories



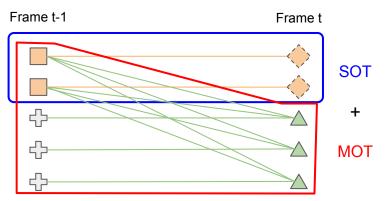
# A New Object Trajectory Representation

- History observation feature: per-frame feature extraction given the trajectory
- **History motion feature:** location displacement over the trajectory
- Feature fusion and temporal modelling



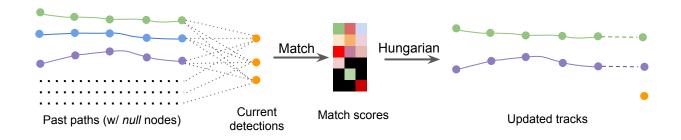
### Discrete Tracking

- For each <u>past track</u>:
  - It's associated with a current detection Multi-Object Tracking
  - It's unassociated, therefore need to "hallucinate" its current state → Single-Object Tracking
- For each *current detection*:
  - It's associated with a past track
     Multi-Object Tracking
  - It's unassociated, therefore need to "birth" a new track
    Multi-Object Tracking



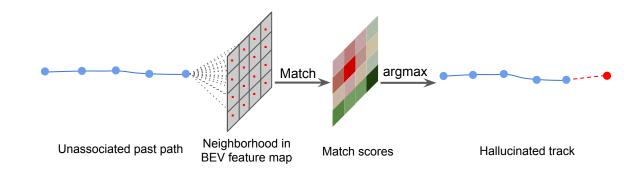
# Multi-Object Tracking

- Handle "newborn" objects by adding <u>null</u> nodes at past tracks side
- Learnable matching function
- Hungarian algorithm for optimal assignment



# Single-Object Tracking

- Perform on unassociated tracks
- Inherit the spirit from Siamese tracker [2], but replaces correlation with a learnable match function
- Produce more accurate estimations by exploiting observations

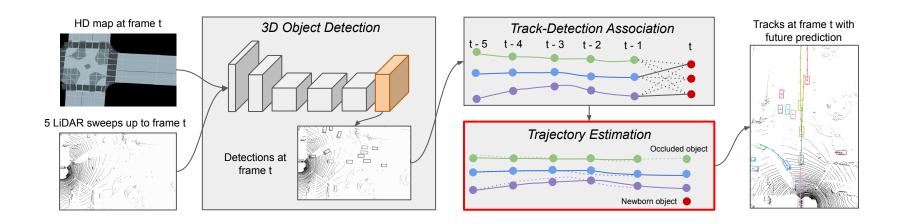


<sup>[1]</sup> PnPNet: End-to-End Perception and Prediction with Tracking in the Loop [Liang, M. et al., CVPR 2020]

<sup>[2]</sup> Fully-Convolutional Siamese Networks for Object Tracking [Bertinetto, L. et al., ECCV 2016]

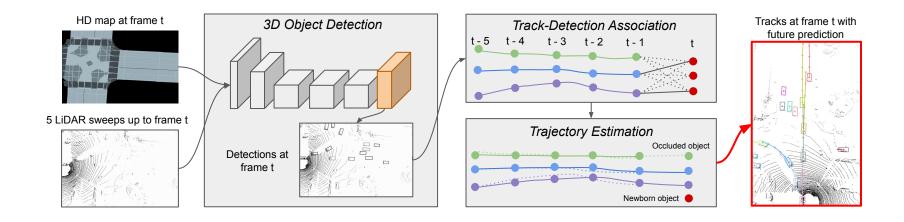
# Continuous Tracking

- Classification: re-estimate the object confidence
- Regression: Smooth the past trajectory
- Post-Process: NMS, keep top-50 confident objects



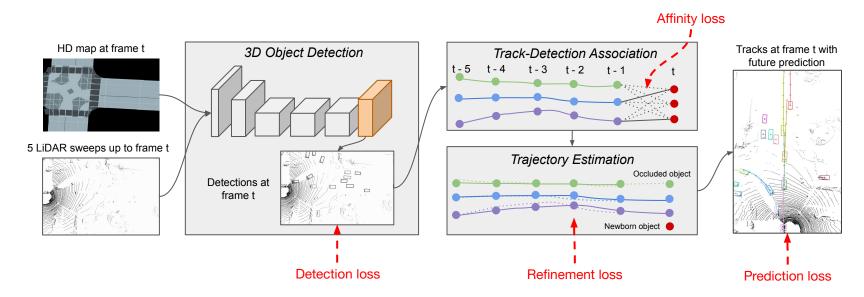
## Motion Forecasting

- Prediction header: simple regression based prediction
- Input features: trajectory features after tracking



## End-to-End Learning

- We adopt multi-task loss for detection, tracking and prediction
- Video-centric training, with online estimations from previous modules & time steps



#### 3D Detection Results on nuScenes

The detection module of PnPNet achieves state-of-the-art performance

Methods	AP↑	AP@0.5m	@1m	@2m	@4m
Mapillary [40]	47.9	10.2	36.2	64.9	80.1
PointPillars [22]	70.5	55.5	71.8	76.1	78.6
Megvii [55]	82.3	72.9	82.5	85.9	<b>87.7</b>
PnPNet, det only	82.7	73.7	83.3	86.2	87.5

## Multi-Object Tracking Results on nuScenes

- Compared with state-of-the-art, PnPNet achieves 8.0% gain in AMOTA
- Compared with a Kalman Filter based tracker, PnPNet achieves 4.6% gain in AMOTA
- PnPNet also produces more complete trajectories

Methods	<b>AMOTA</b> ↑	AMOTP↓	RECALL↑	MOTA↑	MOTP↓	MT↑	ML↓	FP↓	IDS↓	FRAG↓	TID↓	LGD↓
StanfordIPRL-TRI [13]	73.5%	0.53	73.8%	62.3%	0.26	1978	1053	6340	367	341	0.79	1.08
PnPNet, KF tracker	76.1%	0.52	79.1%	64.8%	0.24	2351	745	7555	802	628	0.51	0.97
PnPNet	81.5%	0.44	81.6%	69.7%	0.26	2518	804	6771	152	310	0.30	0.57

AMOTA/AMOTP: MOTA/MOTP averaged over different recall thresholds:

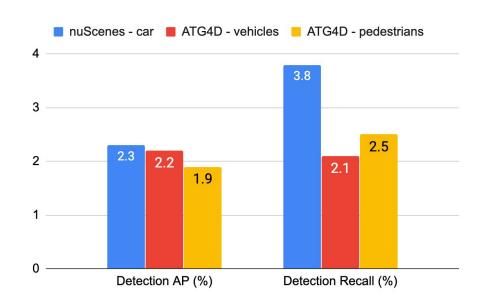
**TID:** average track initialization duration in seconds;

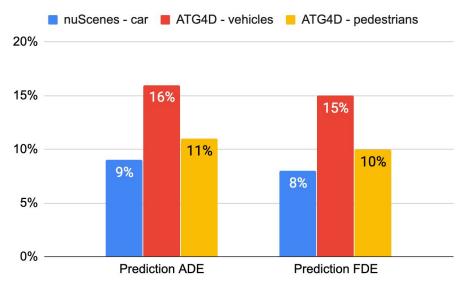
**LGD:** average longest gap duration in seconds.

#### Joint P&P Results on nuScenes & ATG4D

#### Absolute gain in **perception** metrics

#### Relative error reduction in **prediction** metrics





[1] PnPNet: End-to-End Perception and Prediction with Tracking in the Loop [Liang, M. et al., CVPR 2020]

# Qualitative Results

