Deep Learning-based Trajectory Prediction using Improved HiVT

System Control Lab.
44211032-5 SUN Honglin

2022/2/6

早稲田大学大学院 情報生産システム研究科
WASEDA University
Graduate School of Information, Production and Systems
WASEDA UNIVERSITY

Contents

WASEDA UNIVERSITY

1.2 Issue & Idea

Issue

- Most learning-based methods require heavy computation
  → Hard to apply to vehicles
- Long latency will make the prediction meaningless

Idea

Perform a two-stage modification on HiVT-64:

- Improve the computational efficiency
  - ~~Cut down parameters and layers~~
  - ~~Modify or remove some operations~~
  - Implement a new local encoder
- Improve the prediction accuracy
  - Refine the network structure
  - Introduce more useful features
  - ~~Improve the training strategy~~

TABLE I: Comparison of some deep learning-based methods (LaneGCN as the baseline of accuracy)

Method	**Accuracy**	**#Param**	**Device (GPU)**
LaneGCN	100.00%	3,701K	GTX TITAN X
Scene Transformer	110.71%	15,296K	Tesla V100
GOHOME	94.05%	400K	RTX 2080 Ti
HiVT-64 [4]	99.77%	653K	RTX 2080 Ti

TABLE II: Comparison of some devices (RTX 2080 Ti as the baseline of performance)

Device	**Performance**	**Power**
RTX 2080 Ti	100.0%	250 W
GTX 1070 Max-Q	42.0%	115 W
Jetson AGX Xavier	10.5%	30 W
Jetson AGX Orin	39.6%	60 W

Onboard AI chips

[4] Zhou, Zikang, et al. "HiVT: Hierarchical Vector Transformer for Multi-Agent Motion Prediction." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022.

Contents

WASEDA UNIVERSITY

2.0 Intro to HiVT (**Hi**erarchical **V**ector **T**ransformers)

Local regions

- Small circular regions centered at each agent

Modules inside the model

(4 stacked Transformers)

a. <u>Agent-Agent interaction</u> × T
   Encode agents' interaction in local region at each time step

b. <u>Temporal transformer</u>
   Capture temporal information

c. <u>Agent-Lane interaction</u>
   Encode the map information

d. <u>Global interaction</u> × 3
   Encode the relationship between local regions
   (Perform 3 times)

Local feature matrix: Spatio-temporal information of the local region

Global feature matrix: Spatio-temporal information and geometric relationship with other local regions

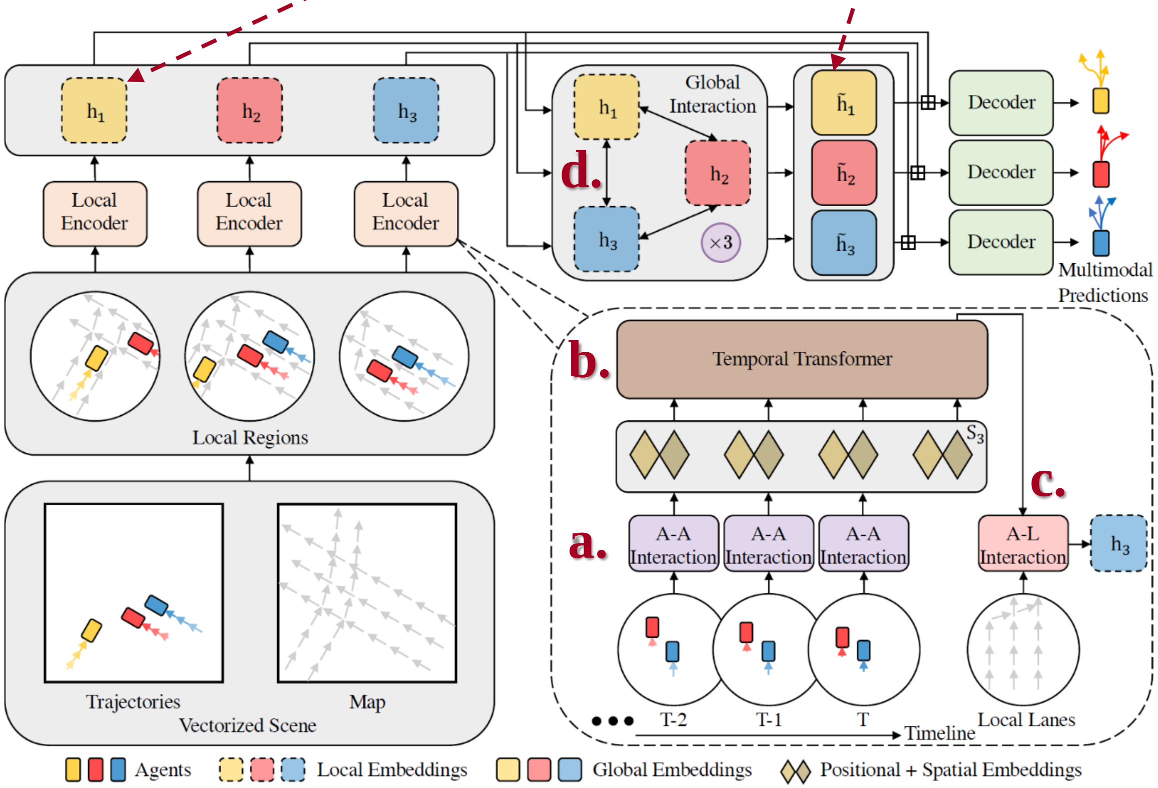


Fig. 4: Overview of HiVT

<u>Agent</u>: Traffic participants

2.1 Computational Efficiency Improvement

Implement a new local encoder

- ☐ <u>Original local encoder</u>
1. Agent-Agent Interaction $\times\ T$
2. Temporal Transformer
3. Agent-Lane Interaction

- ☐ A-A interaction module: time-consuming
  - ■ Not all agents have interaction with each other

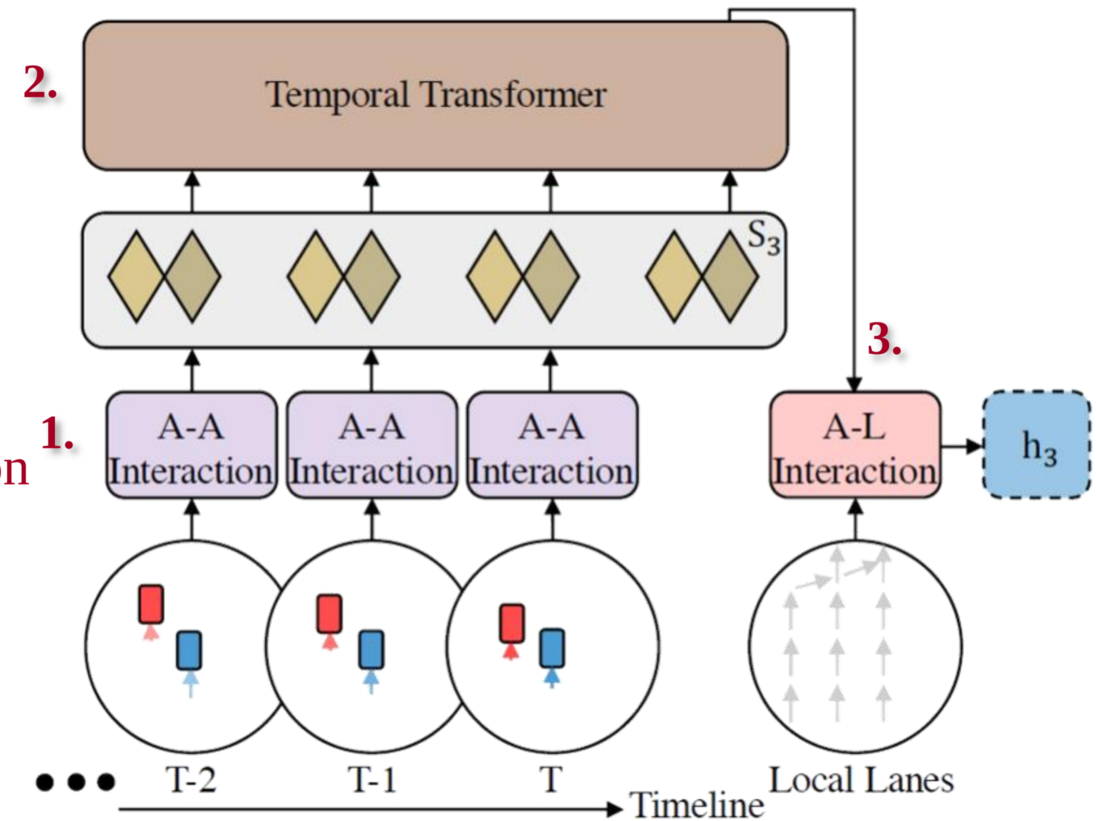


Fig. 12: Original local encoder

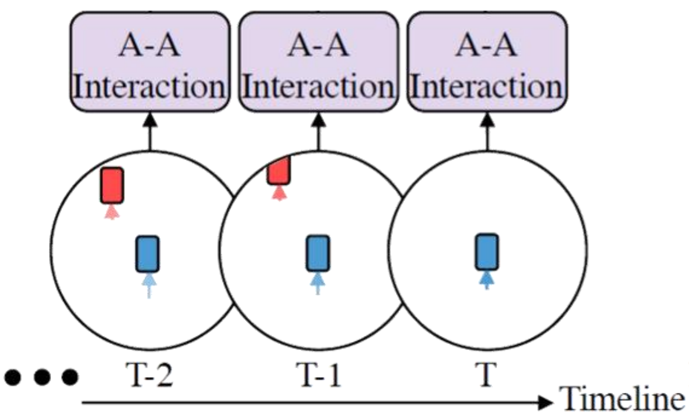


Fig. 13: Issue of A-A interaction

WASEDA UNIVERSITY

2.1 Computational Efficiency Improvement

Implement a new local encoder

☐ Original local encoder

1. Agent-Agent Interaction $\times T$
2. Temporal Transformer
3. Agent-Lane Interaction

☐ Proposed local encoder

a. Agent Encoder $\times T$

b. Temporal Transformer

c. Agent-Lane Interaction

d. Agent-Agent Interaction $\times$ 1

A-L before A-A: Enrich the features of each agent

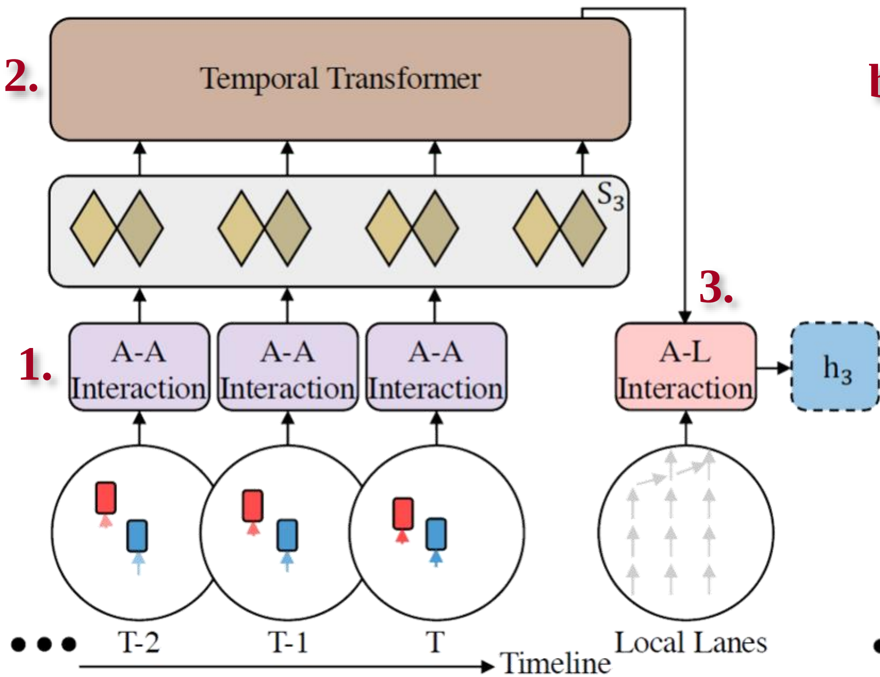


Fig. 12: Original local encoder

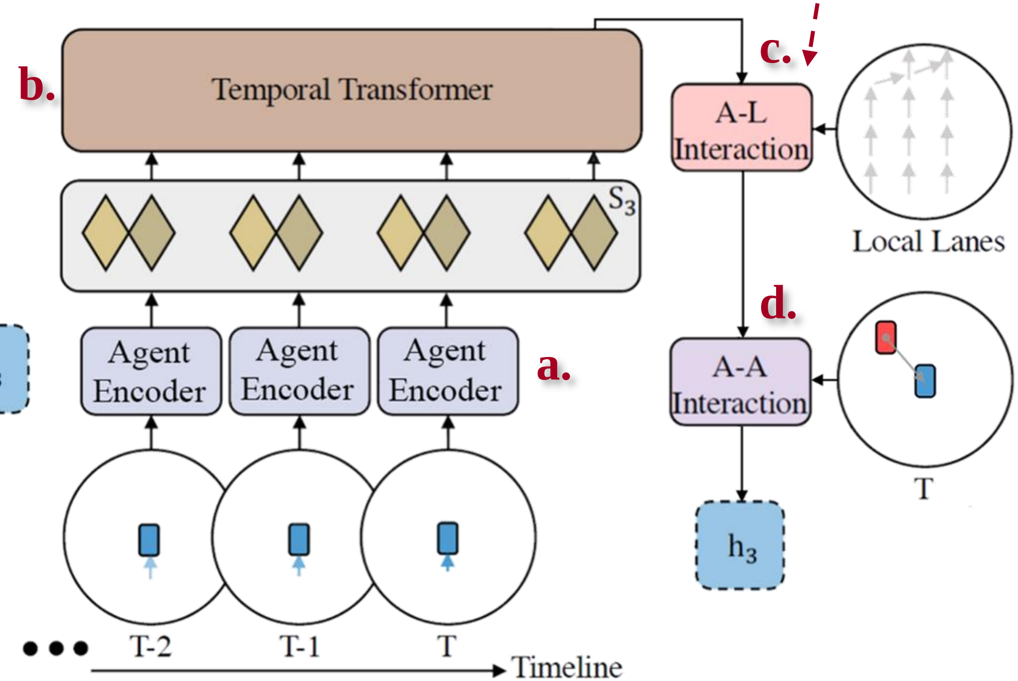
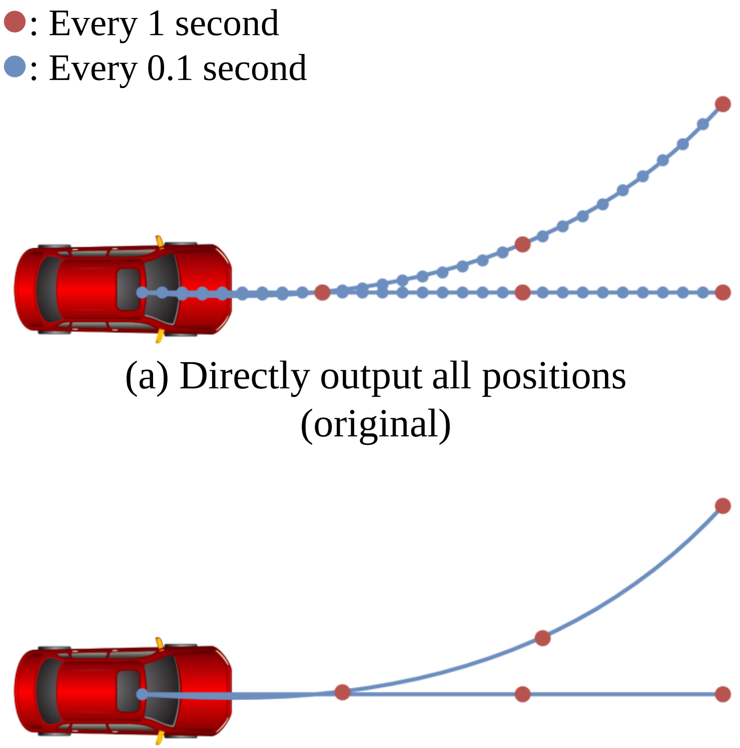


Fig. 14: Proposed local encoder

2.2 Prediction Accuracy Improvement

Refine the network structure

- Modify the trajectory generation
  - Change the decoder output size:
    30-position dense output (0.1s × 30)
    → 3-position sparse output (1s × 3)
    - Fewer data need to learn: 30 × 2 → 3 × 2
    - Makes the model easier to converge
  - Use cubic B-spline interpolation to complete the full trajectory
    - Driving purpose with 1-second interval is almost deterministic
      → Enough for the mathematical method to interpolate an accurate trajectory



(a) Directly output all positions (original)

(b) Sparse output with interpolation (proposed)

Fig. 15: Comparison of different output

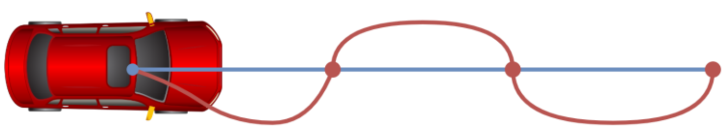


Fig. 16: Example of a driving purpose (go straight)

2.2 Prediction Accuracy Improvement

Refine the network structure

- Modify the fusion method of edge attributes (e) inside global interaction module
  - Original

    $$\boldsymbol{e}_{ij} = \phi_{\mathrm{MLP}}([R_{ij}\boldsymbol{p}_{ij}, \sin(\theta_{ij}), \cos(\theta_{ij})])$$
    $$\boldsymbol{h}_j = \boldsymbol{h}_j$$
    $$\rightarrow Q = W_{Q_{\mathrm{node}}}\boldsymbol{h}_i$$
    $$K = W_{K_{\mathrm{node}}}\boldsymbol{h}_j + W_{K_{\mathrm{edge}}}\boldsymbol{e}_{ij}$$
    $$V = W_{V_{\mathrm{node}}}\boldsymbol{h}_j + W_{V_{\mathrm{edge}}}\boldsymbol{e}_{ij}$$

  - Proposed

    $$\boldsymbol{e}_{ij} = [R_{ij}\boldsymbol{p}_{ij}, \sin(\theta_{ij}), \cos(\theta_{ij})]$$
    $$\boldsymbol{h}_j = \phi_{\mathrm{MLP}}([\boldsymbol{h}_j, \boldsymbol{e}_{ij}])$$
    $$\rightarrow Q = W_{Q_{\mathrm{node}}}\boldsymbol{h}_i$$
    $$K = W_{K_{\mathrm{node}}}\boldsymbol{h}_j$$
    $$V = W_{V_{\mathrm{node}}}\boldsymbol{h}_j$$

    Retains more node information

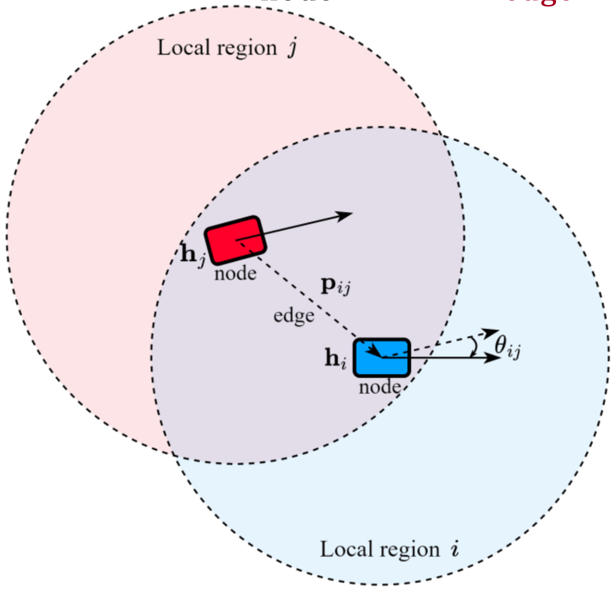


Fig. 17: Illustration of edge attributes

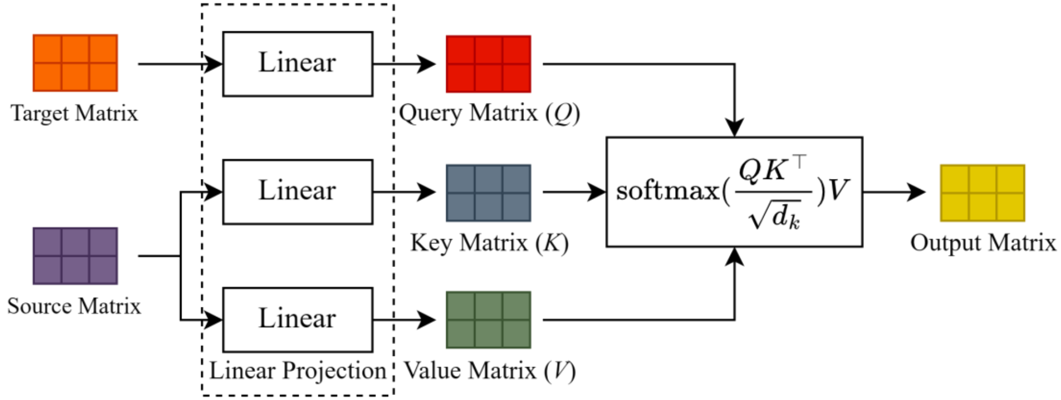


Fig. 7: Scaled-dot product attention inside the transformers

WASEDA UNIVERSITY

2.2 Prediction Accuracy Improvement

Introduce more useful features

- ☐ Add learnable time weights to the temporal transformer

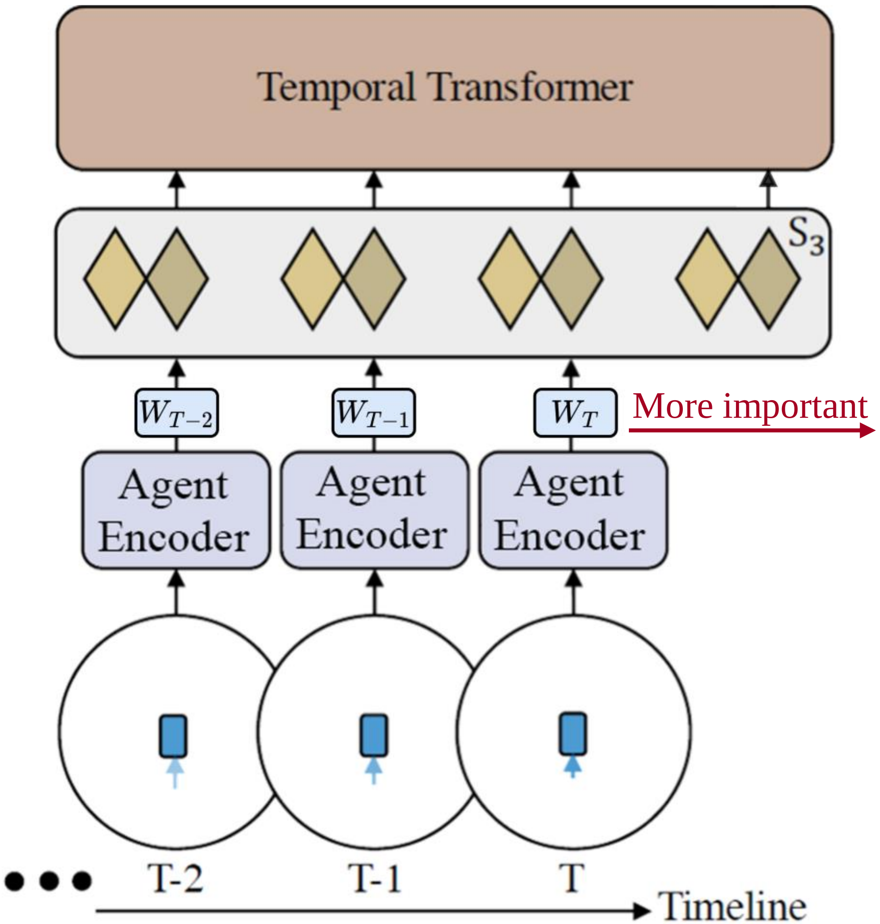


Fig. 21: Time weights

- ☐ Fusing lane adjacency relationship as semantic information

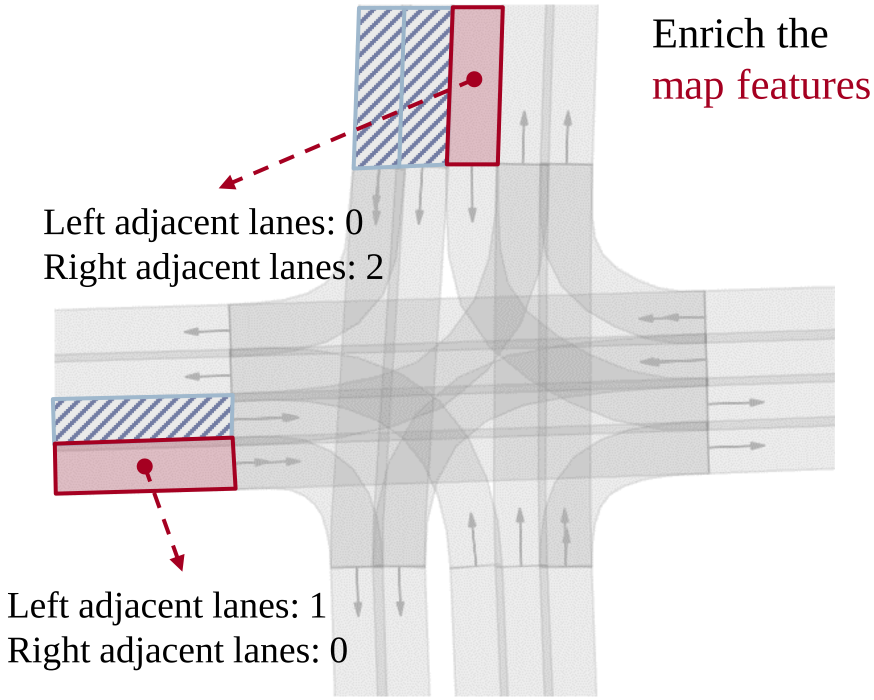
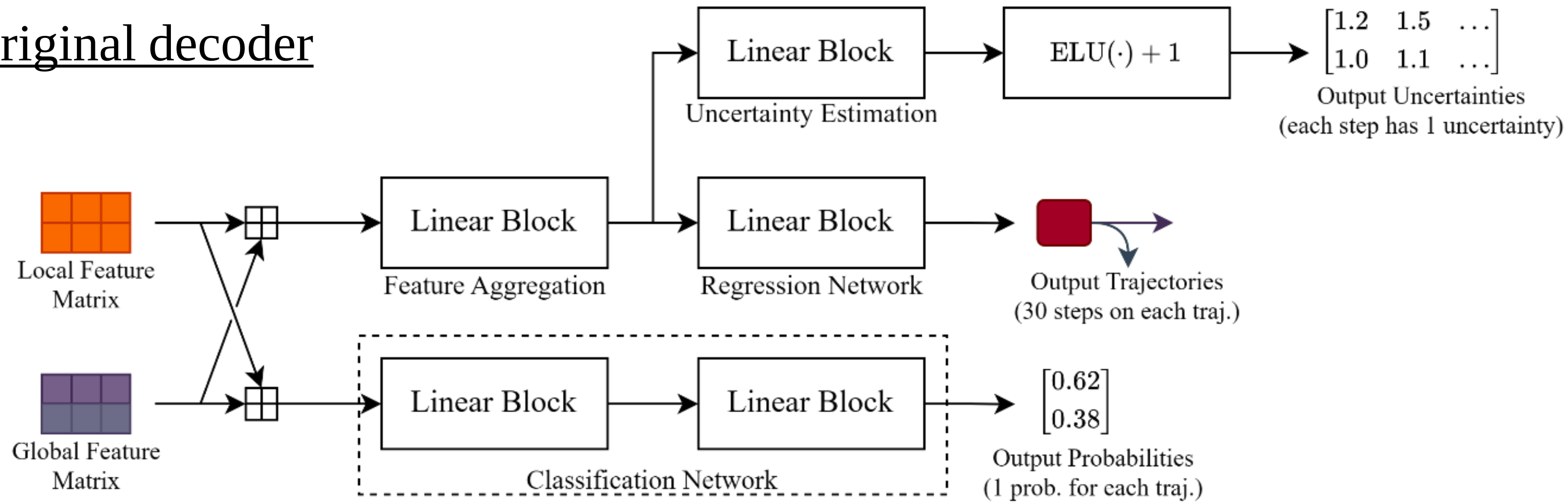


Fig. 22: Example usage of lane adjacency relationship

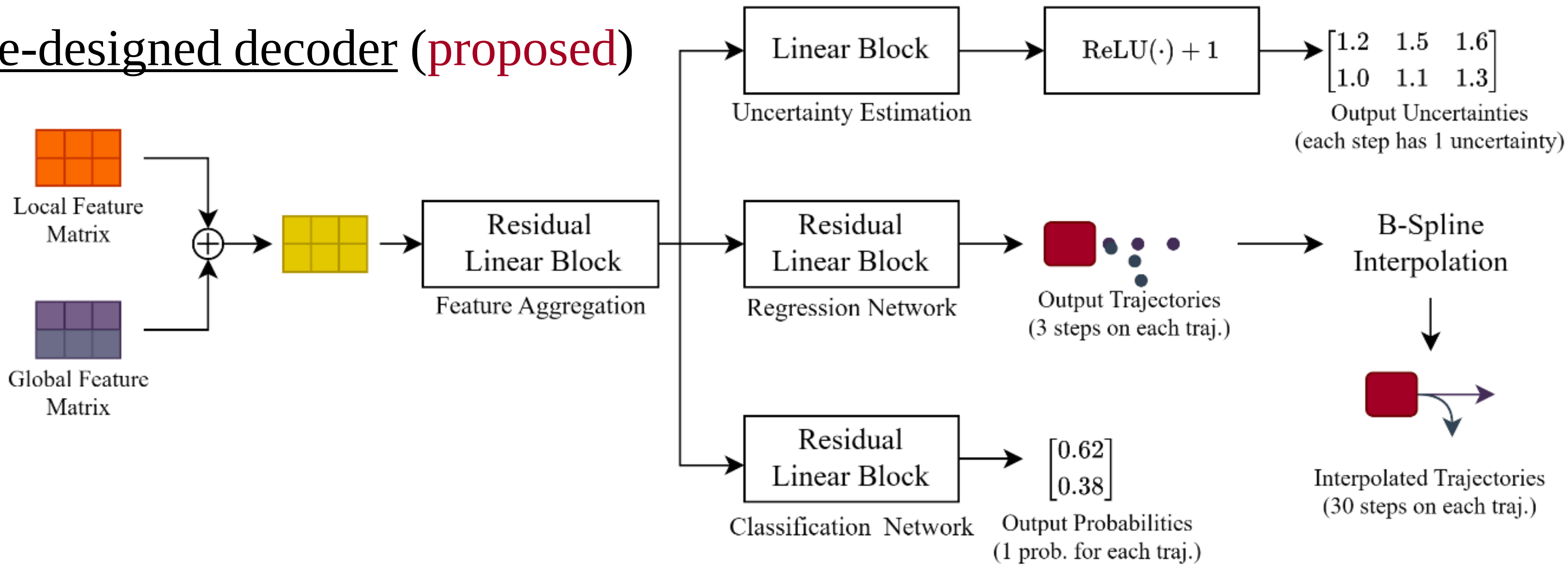
2.2 Prediction Accuracy Improvement

Overall comparison of the decoder structure

- Original decoder

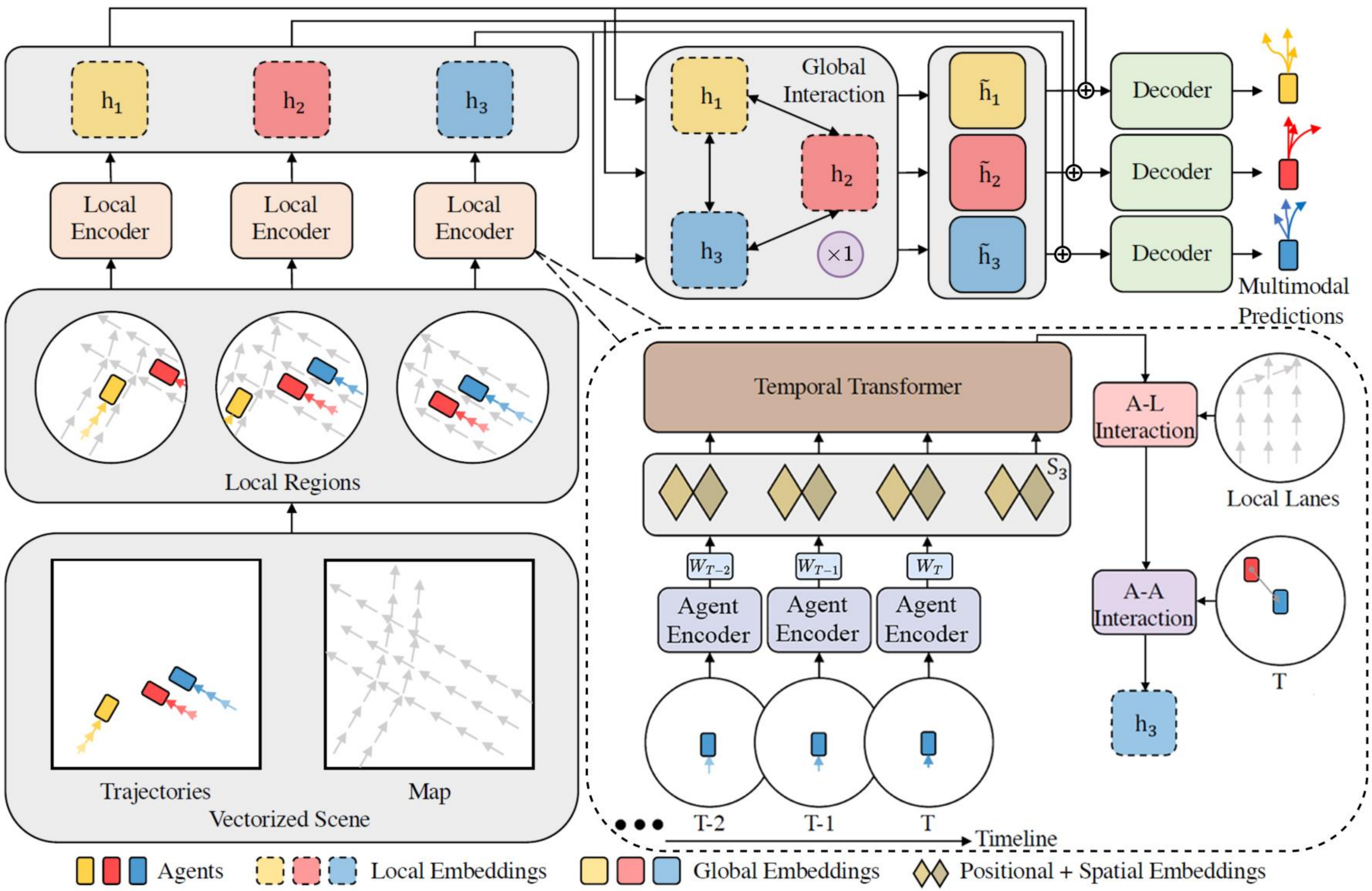


- Re-designed decoder (proposed)



2.2 Prediction Accuracy Improvement

Overall structure of the modified HiVT



Contents

Hardware: i7-8750H & GTX 1070 Max-Q
Software: Ubuntu 20.04 LTS & PyTorch 1.12
Training: 131 epochs, max learning rate 5e-4

3.1 Simulation Setup

Argoverse 1 Motion Forecasting Dataset [6]

- The same as used in HiVT and other papers
- Data was collected in Miami and Pittsburgh
- 324,557 scenarios in the dataset, split into:
  - Training set: 205,942 scenarios
  - Validation set: 39,472 scenarios
  - Test set: 78,143 scenarios
- 5 seconds of data for each scenario
  - 10 Hz sampling rate
- Task
  - 2-second historical trajectory + Map

    ↓ Predict
  - 3-second possible future trajectories (30 positions × number of trajs.)

```
lanes[lane_id] = {
  centerline = [X,Y]
  is_intersection = True
  turn_direction = 'LEFT'
  traffic_controls = True
  left_neighbor = None
  right_neighbor = lane_id
  predecessors = [lane_ids]
  successors = [lane_ids]
}
```

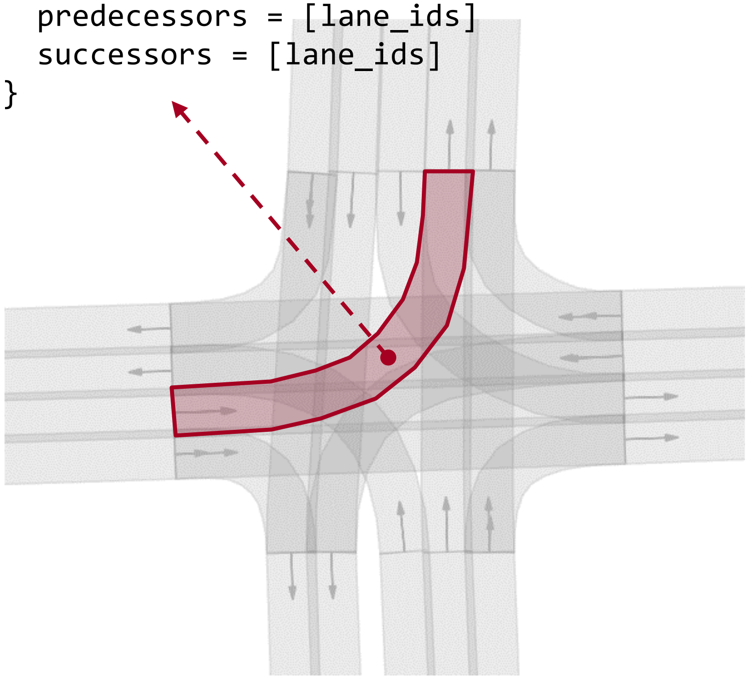


Fig. 24: Example of map data structure

[6] Chang, Ming-Fang, et al. "Argoverse: 3D Tracking and Forecasting with Rich Maps." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019.

3.1 Simulation Setup

K=1: Consider the highest prob. trajectory
K=6: Consider all the 6 trajectories

Visualization

- Predict 6 possible future trajectories
  - Yellow line: Historical trajectory (2s)
  - Red line: True future trajectory (3s)
  - Green lines: Predicted trajectories, probability declines from 1 to 6
  - Cyan dots: Final positions of other agents
  - Gray polygons: Lane segments
  - Gray arrows: Lane directions

- Good prediction: Close to the red line
- Plot one vehicle's prediction result
- Darker green indicates a higher probability

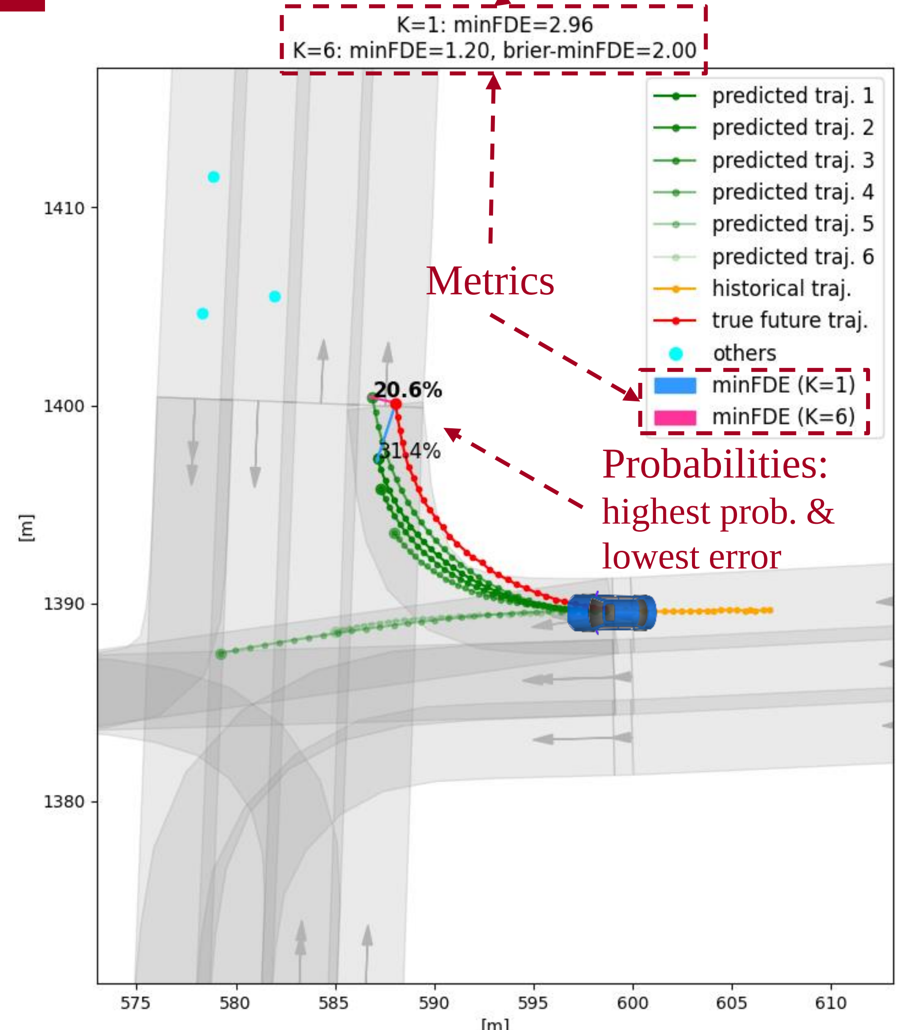


Fig. 25: Example of prediction result

WASEDA UNIVERSITY

3.1 Simulation Setup

Metrics (Lower is better)

- minADE: Minimum Average Displacement Error

$$\mathrm{minADE} = \min\left(\frac{1}{T}\sum_{t=1}^{T}\|\hat{\boldsymbol{y}}_t - \boldsymbol{y}_t\|_2\right)$$

- minFDE: Minimum Final Displacement Error

$$\mathrm{minFDE} = \min(\|\hat{\boldsymbol{y}}_T - \boldsymbol{y}_T\|_2)$$

- brier-minFDE [7]: minFDE + probability term
  → Evaluate both regression and classification performance

$$\mathrm{brier{-}minFDE} = \min(\|\hat{\boldsymbol{y}}_T - \boldsymbol{y}_T\|_2 + (1-\boldsymbol{p})^2)$$

- MR: Miss Rate

$$\mathrm{MR} = \frac{\#\mathrm{scenarios\_with\_minFDE{>}2[m]}}{\#\mathrm{scenarios}}$$

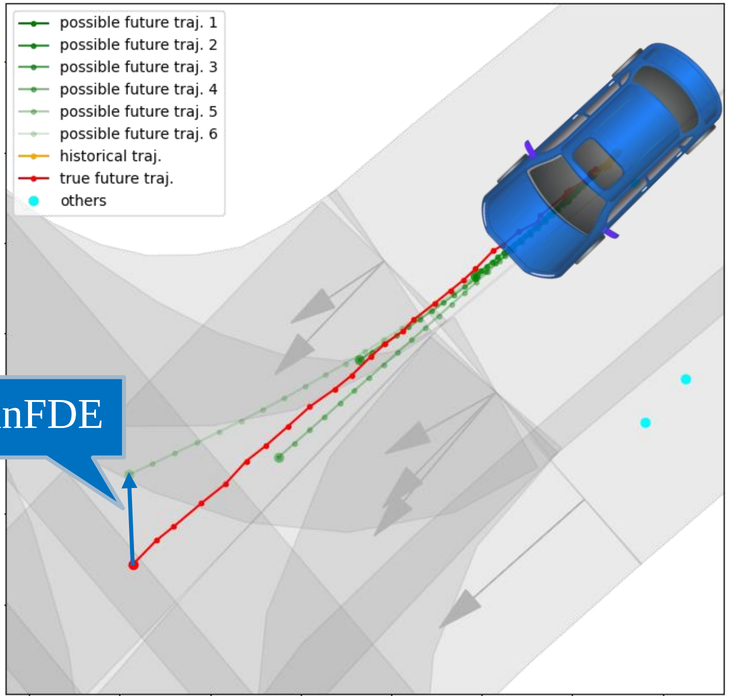


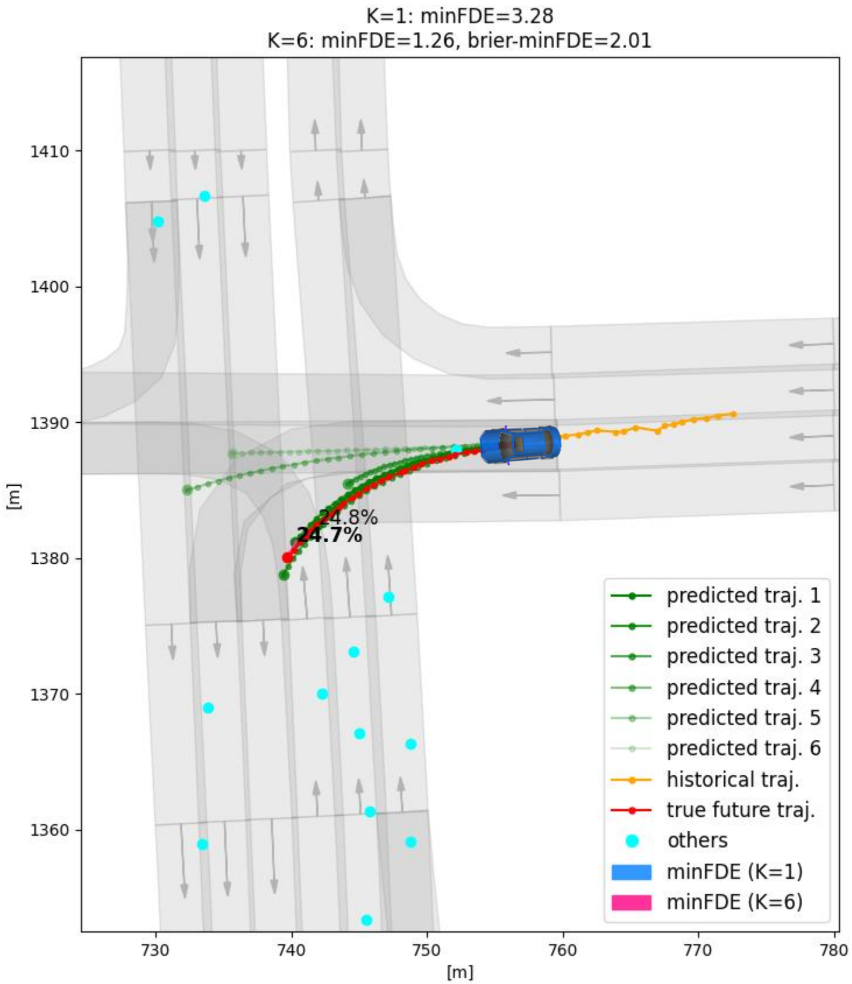
Fig. 29: Calculation of minFDE

$\hat{\boldsymbol{y}}_t$: Predicted future positions at time step t
$\boldsymbol{y}_t$: True position at time step t
$\boldsymbol{p}$: Predicted probabilities of future positions

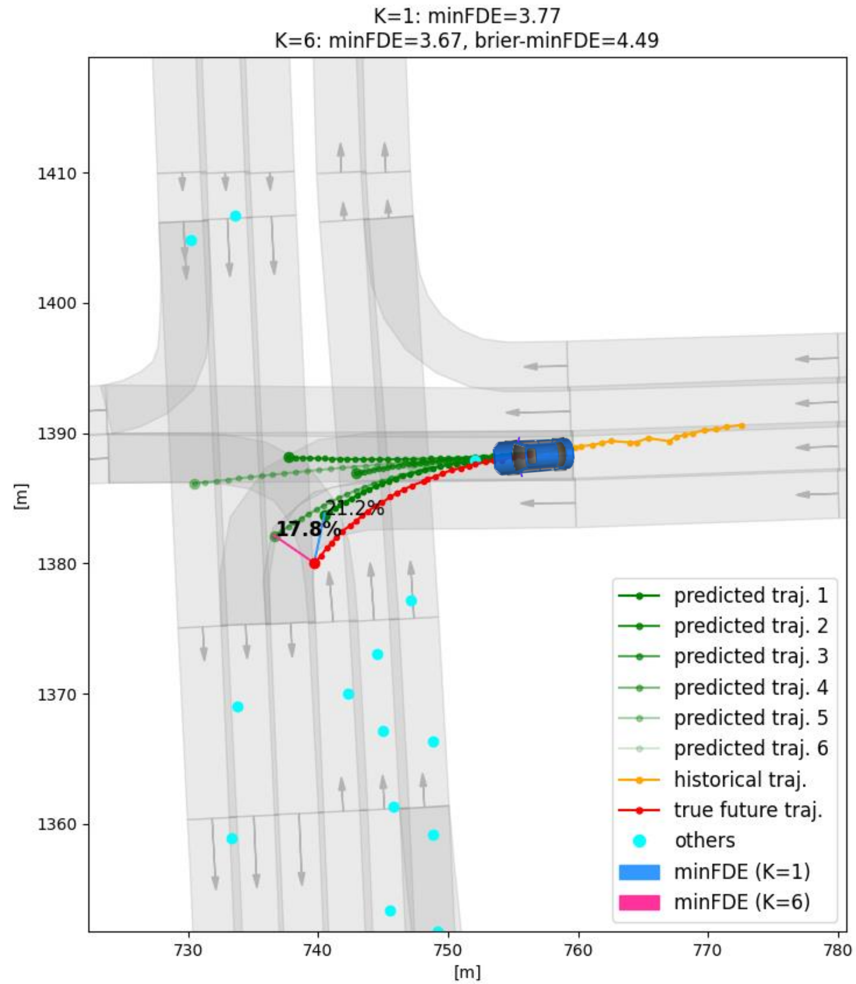
[7] Brier, Glenn W. "Verification of Forecasts Expressed in Terms of Probability." *Monthly Weather Review* 78.1 (1950): 1-3.

3.2 Visualization Results

Comparison with the original HiVT-64 model



(a) Proposed

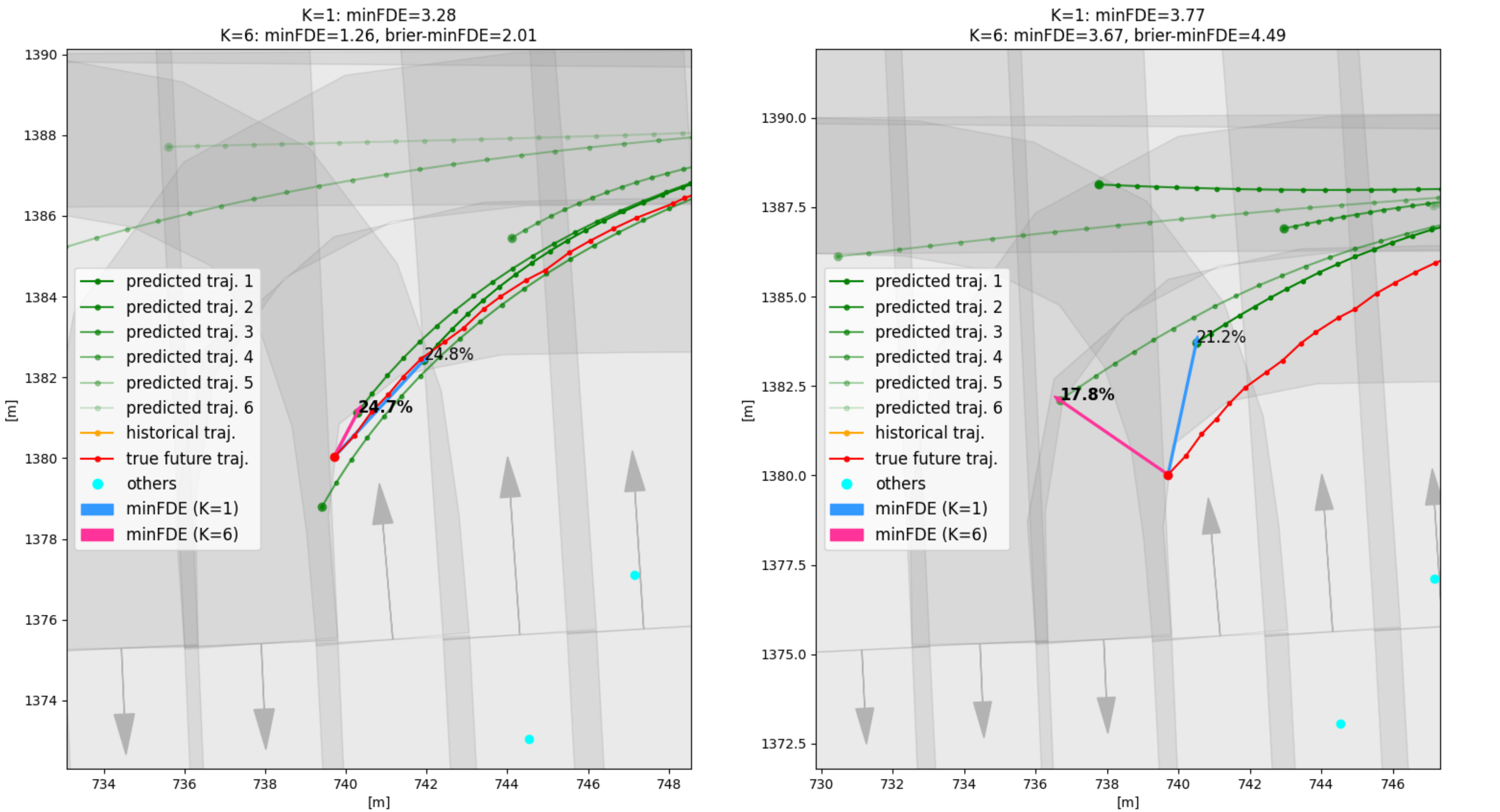


(b) Original

Fig. 26: Turning left

3.2 Visualization Results

Comparison with the original HiVT-64 model (zoom in)

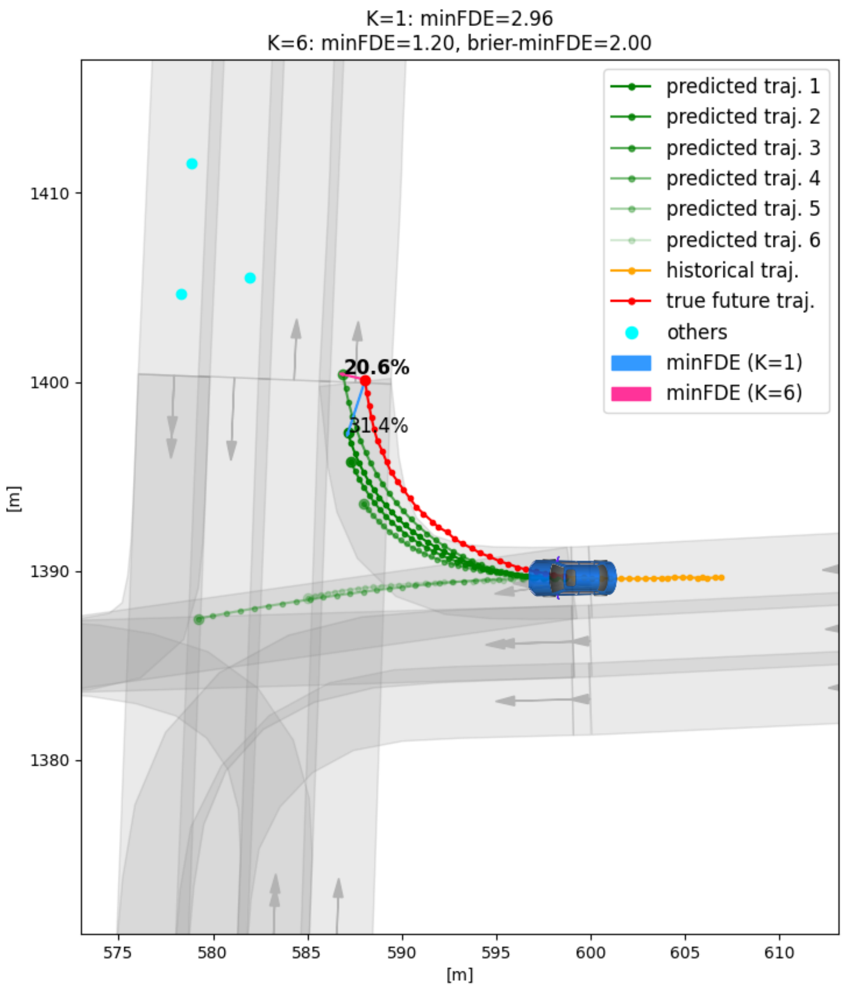


(a) Proposed (b) Original

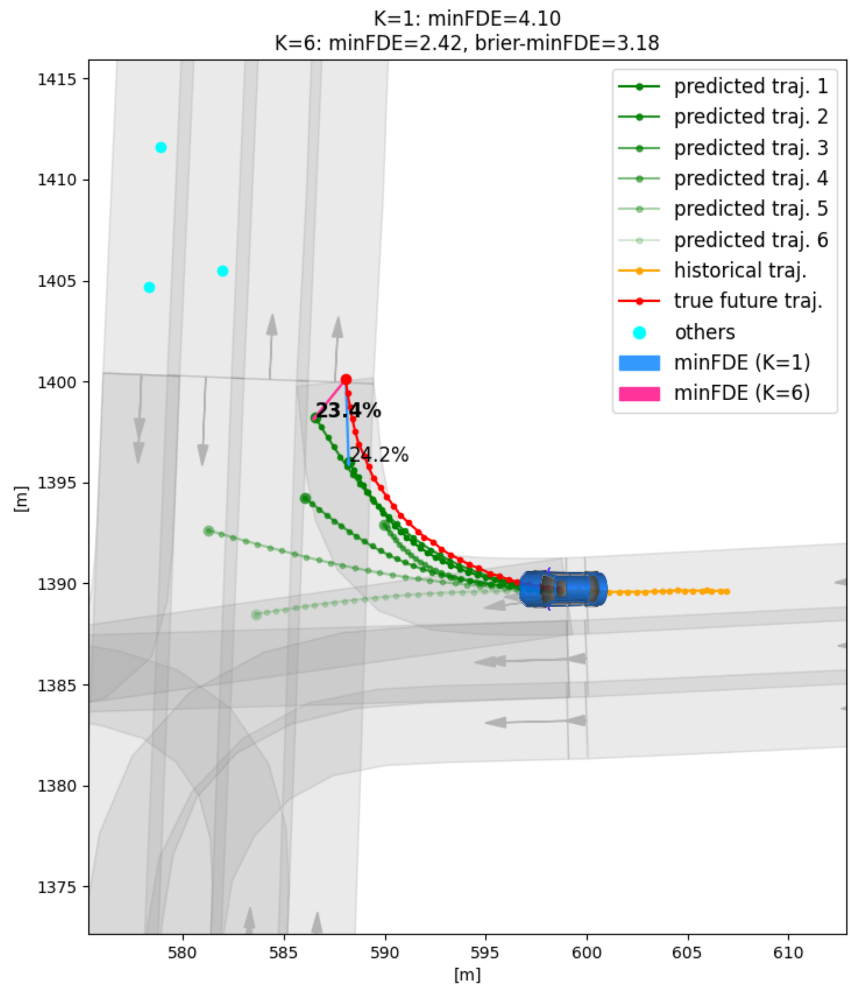
Fig. 26: Turning left

3.2 Visualization Results

Comparison with the original HiVT-64 model



(a) Proposed

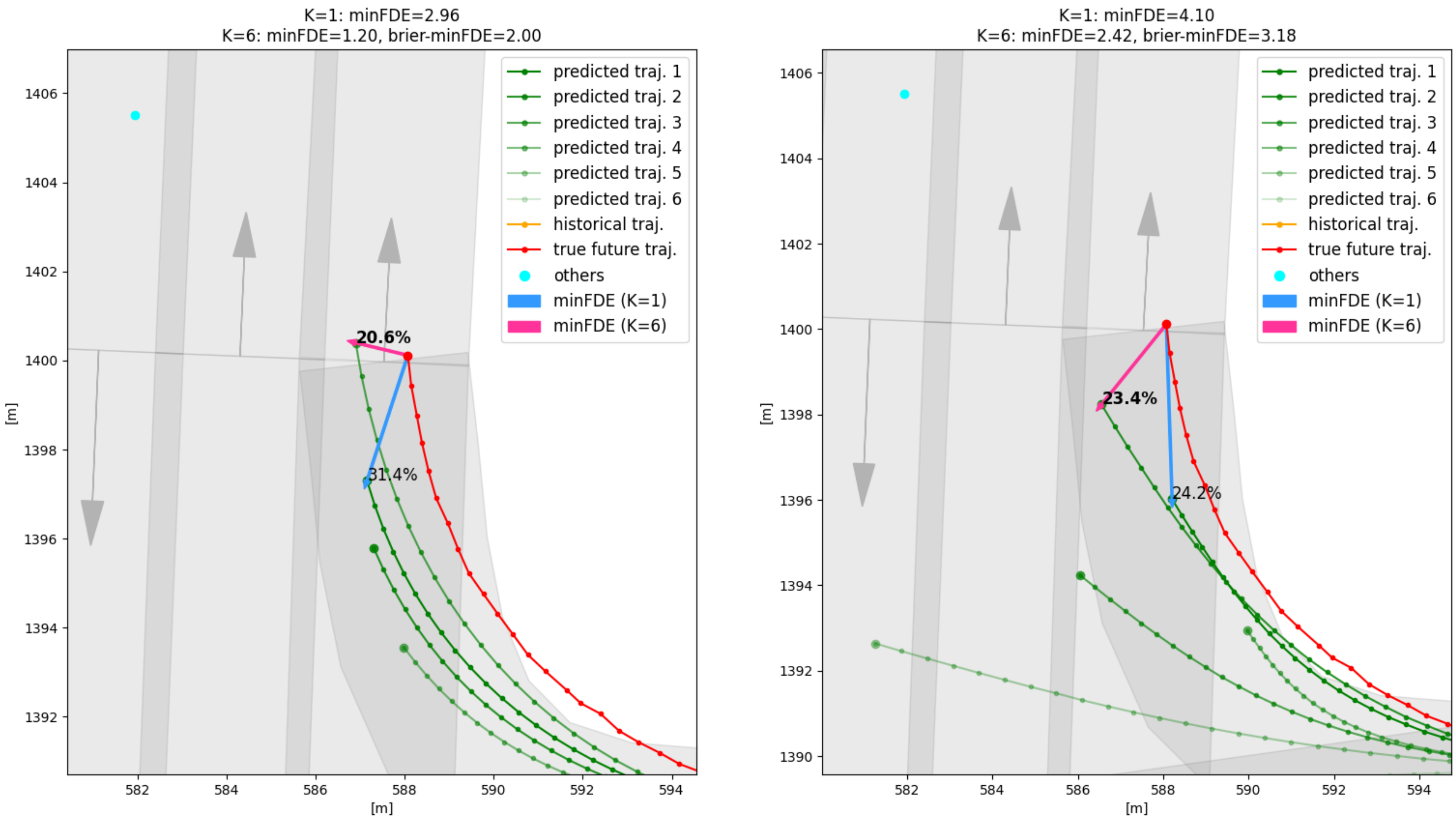


(b) Original

Fig. 27: Turning right (long historical trajectory)

3.2 Visualization Results

Comparison with the original HiVT-64 model (zoom in)

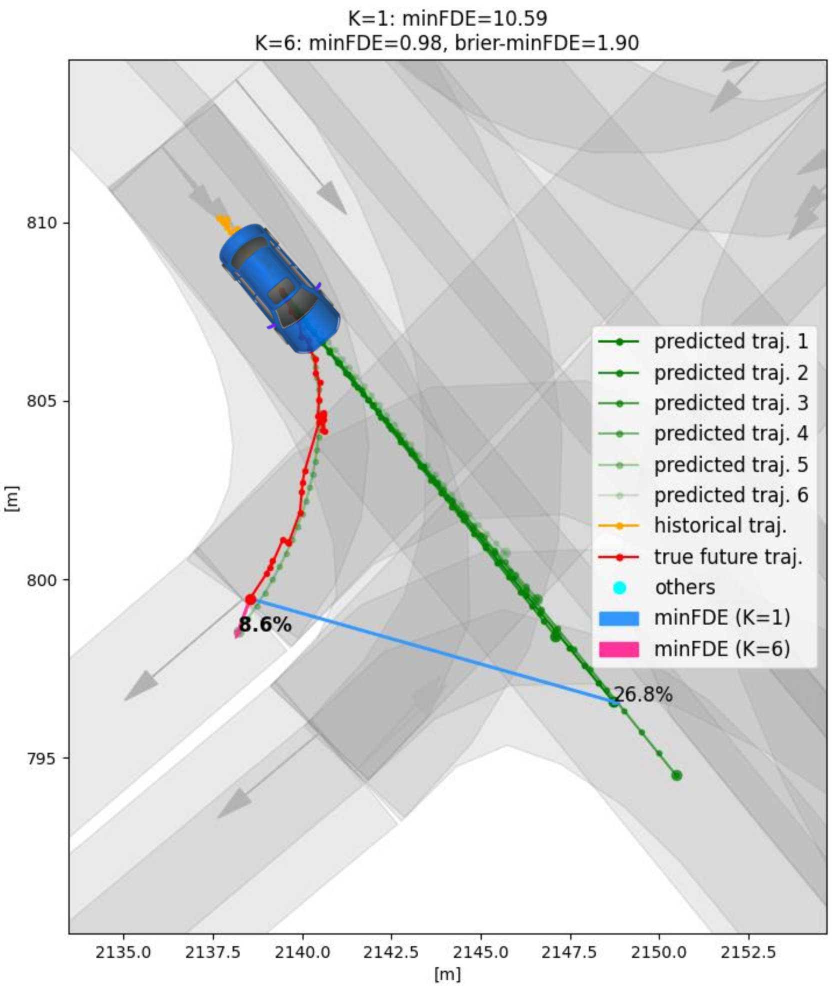


(a) Proposed (b) Original

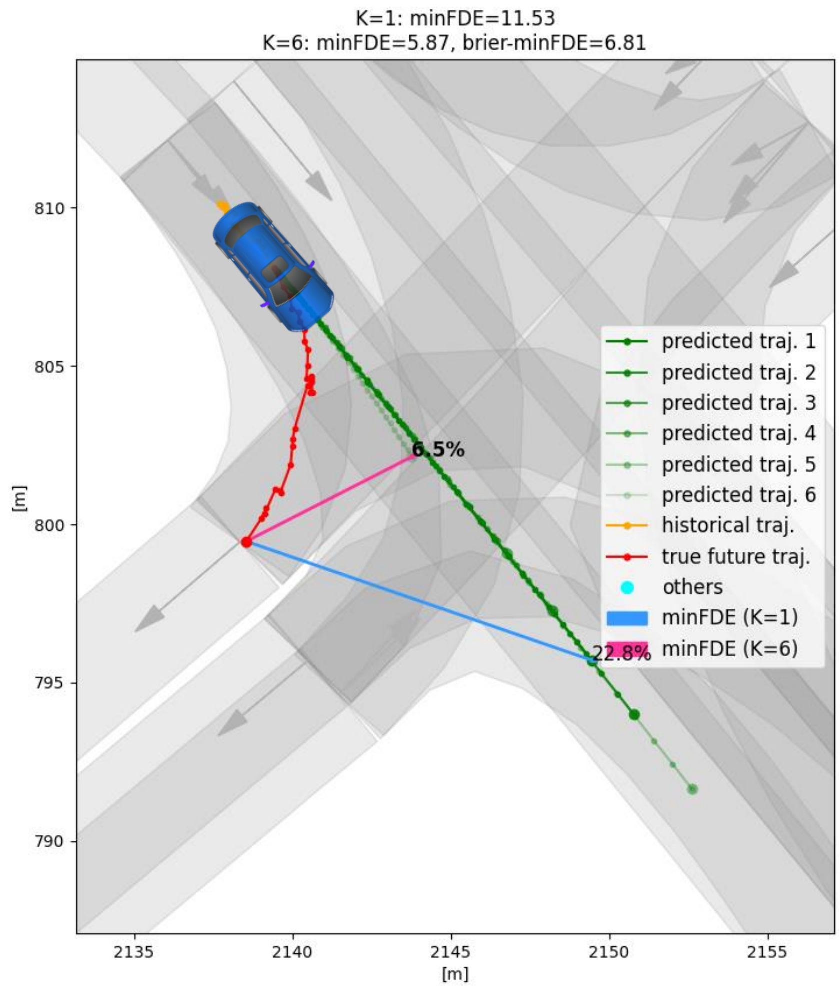
Fig. 27: Turning right (long historical trajectory)

3.2 Visualization Results

Comparison with the original HiVT-64 model



(a) Proposed

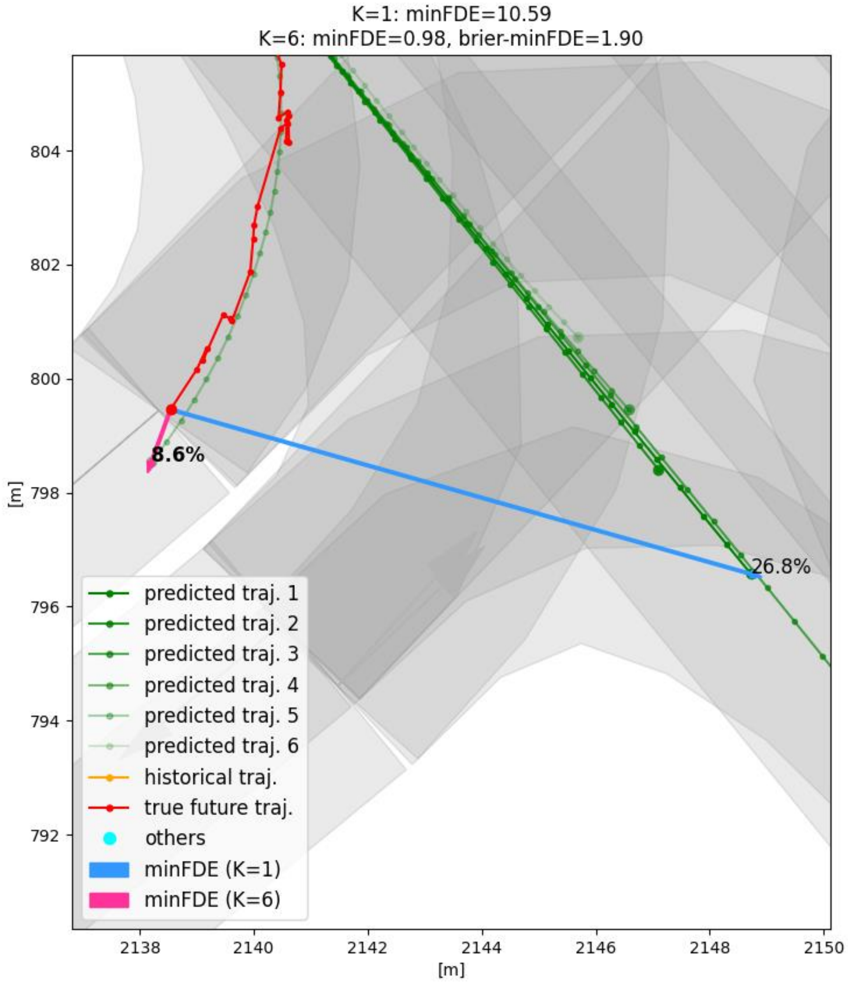


(b) Original

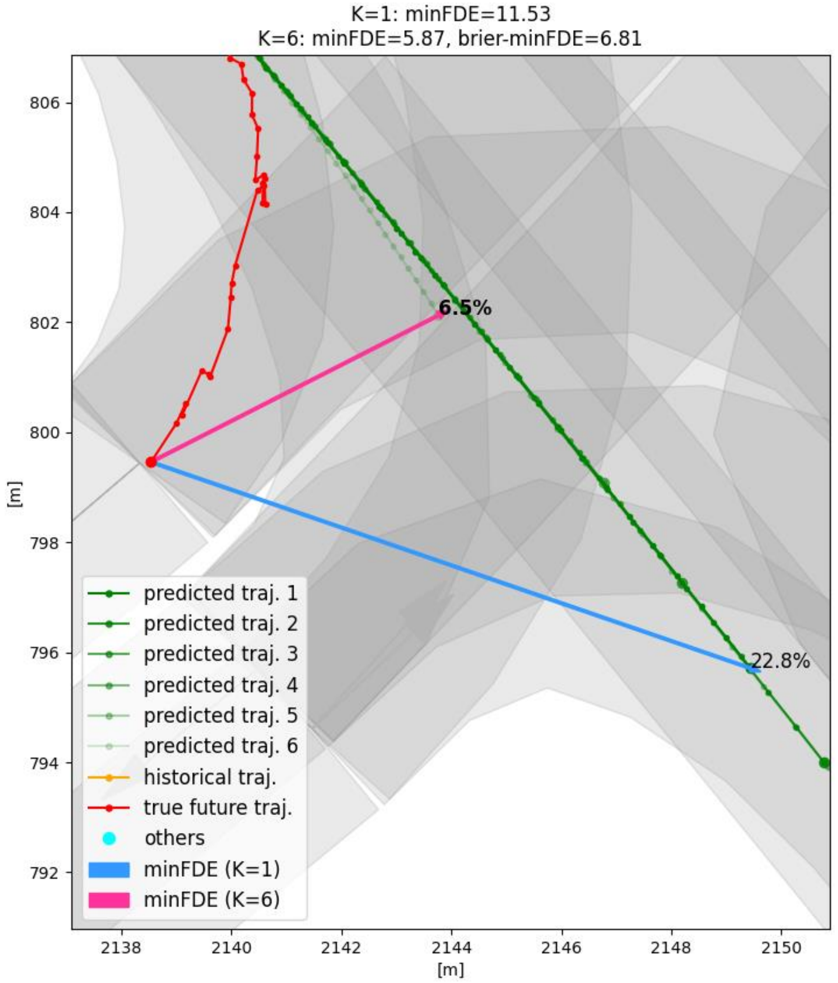
Fig. 28: Turning right (short historical trajectory)

3.2 Visualization Results

Comparison with the original HiVT-64 model (zoom in)



(a) Proposed



(b) Original

Fig. 28: Turning right (short historical trajectory)

3.3 Quantitative Comparisons

Metrics Comparison

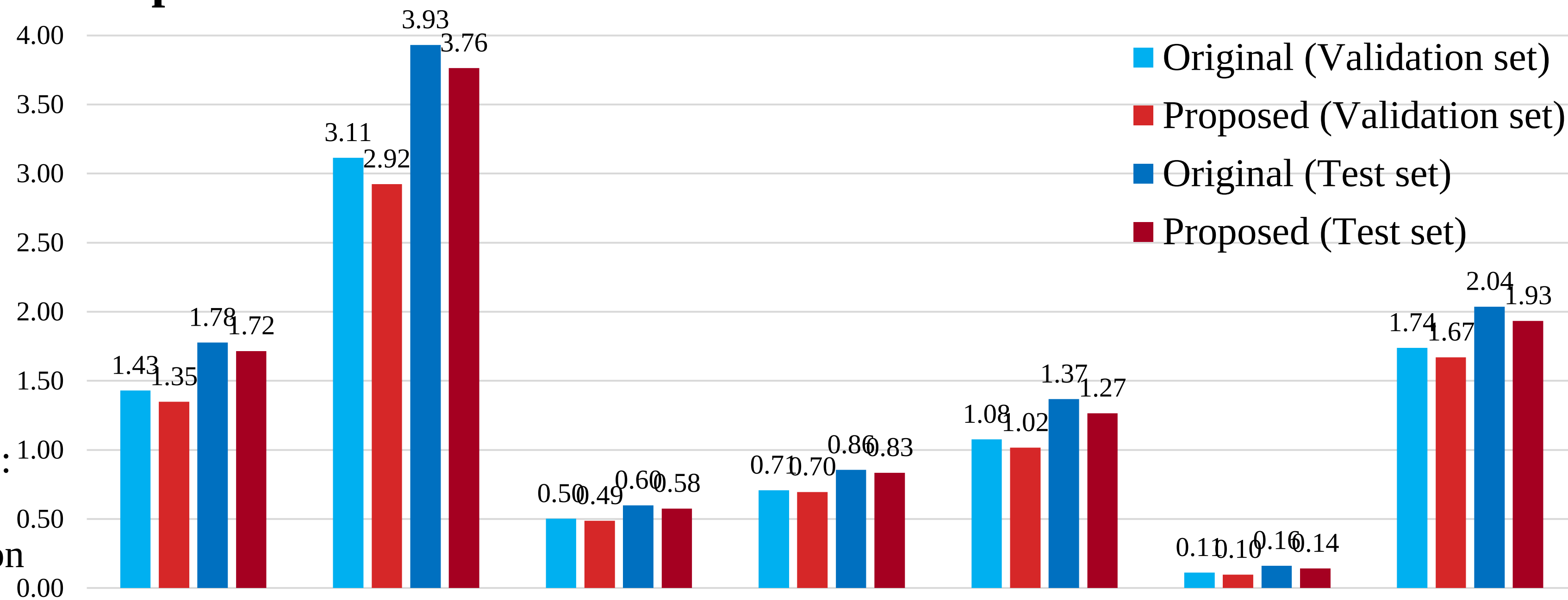


TABLE II: Metrics comparison

	Model	minADE	minFDE	MR	minADE	minFDE	MR	brier-minFDE
Val. Set	Original	1.43	3.11	50.2%	0.71	1.08	11.2%	1.74
	Proposed	**1.35**	**2.92**	**48.8%**	**0.70**	**1.02**	**9.8%**	**1.67**
	Improve	*5.7%*	*6.1%*	*2.9%*	*1.8%*	*5.6%*	*11.9%*	*3.9%*
Test set	Original	1.78	3.93	59.9%	0.86	1.37	16.3%	2.04
	Proposed	**1.72**	**3.76**	**57.6%**	**0.83**	**1.27**	**14.2%**	**1.93**
	Improve	*3.4%*	*4.3%*	*3.8%*	*2.4%*	*7.4%*	*12.5%*	*5.1%*
#Trajs.		K=1			K=6			

3.3 Quantitative Comparisons

Specification Comparison

- Inference speed (frames per second)
  - Test 3 times with the same 5,000 samples and take the average
- GPU memory cost
  - Feed the same data to the network
  - Measure the memory cost in each module

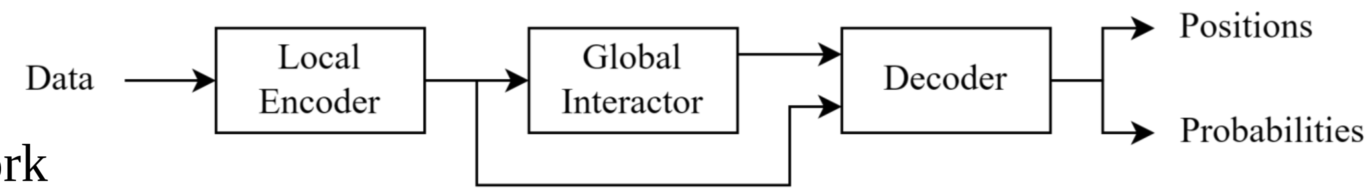


Fig. 30: Rough structure of the model

TABLE III: Specification comparison

Model	#Param.	Inference Speed (fps)	GPU Memory Cost (MB)			
			Local Encoder	Global Interactor	Decoder	Others†
Original	653,369	19.11	430	22	4	10
Proposed	**395,809**	**70.70**	**242**	**2**	**4**	**6**
Improve	*39.4%*	*270.0%*	*43.7%*	*90.9%*	*0.0%*	*40.0%*
Lightest*	389,689	74.05	/	/	/	/

* The model after first-stage modifications.
† Includes model and pre-trained weights.

3.4 Effectiveness Validation

Ablation study

TABLE IV: Ablation study of the second-stage modifications (Validation set, K=6)

Decoder Mods.	Encoder Mods.	Training Strategy	minADE	minFDE	MR	brier-minFDE	#Param.
			0.71	1.06	10.9%	1.72	389,689
✓			0.70	1.04	10.3%	1.69	391,117
✓	✓		0.70	1.03	10.1%	1.69	395,809
✓	✓	✓	**0.70**	**1.02**	**9.8%**	**1.67**	395,809

Validation of time weights

```
self.time_wgt.squeeze().detach().cpu().numpy()
array([2.062761  , 0.51705956, 0.9902031 , 1.0632056 , 1.1559378 ,
       1.1959459 , 1.2953184 , 1.368692  , 1.4293575 , 1.5335826 ,
       1.616901  , 1.7310858 , 1.8082753 , 1.9473771 , 2.0934815 ,
       2.3145213 , 2.5368783 , 2.8813136 , 3.2132146 , 3.3525586 ],
      dtype=float32)
```

Fig. 31: Time weights

WASEDA UNIVERSITY

3.4 Effectiveness Validation

Validation of edge attributes fusion

TABLE V: Comparison with the original fusion method (Validation set)

Model	K=1			K=6			
	minADE	**minFDE**	**MR**	**minADE**	**minFDE**	**MR**	**brier-minFDE**
Original	1.37	2.99	49.1%	0.70	1.03	10.1%	1.69
Proposed	**1.35**	**2.92**	**48.8%**	**0.70**	**1.02**	**9.8%**	**1.67**

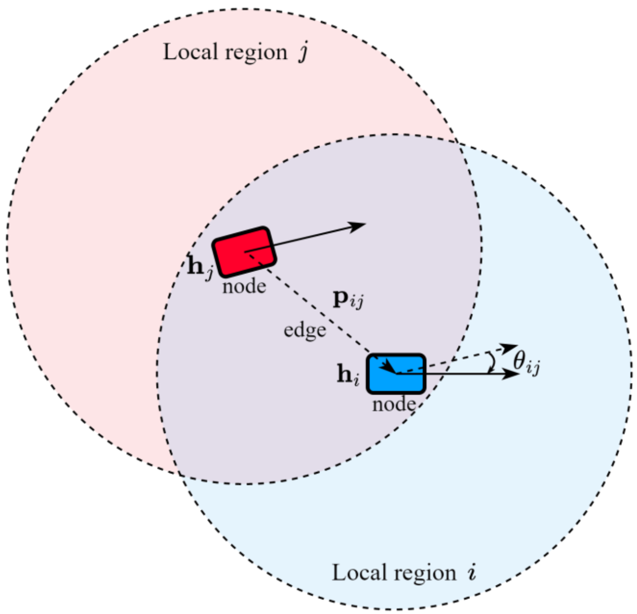


Fig. 17: Illustration of edge attributes

- Original
  - $\boldsymbol{e}_{ij} = \phi_{\text{MLP}}([R_{ij}\boldsymbol{p}_{ij}, \sin(\theta_{ij}), \cos(\theta_{ij})])$
    $\boldsymbol{h}_j = \boldsymbol{h}_j$
- Proposed
  - $\boldsymbol{e}_{ij} = [R_{ij}\boldsymbol{p}_{ij}, \sin(\theta_{ij}), \cos(\theta_{ij})]$
    $\boldsymbol{h}_j = \phi_{\text{MLP}}([\boldsymbol{h}_j, \boldsymbol{e}_{ij}])$

$$Q = W_{Q_{\text{node}}}\boldsymbol{h}_i$$
$$K = W_{K_{\text{node}}}\boldsymbol{h}_j \cancel{+ W_{K_{\text{edge}}}\boldsymbol{e}_{ij}}$$
$$V = W_{V_{\text{node}}}\boldsymbol{h}_j \cancel{+ W_{V_{\text{edge}}}\boldsymbol{e}_{ij}}$$

Contents

4. Conclusions

Conclusions

- Simulation results: The proposed model outperforms the original in both computational efficiency and accuracy
  - Computational efficiency improvement:
    - Inference speed: 2.8x ↑ Number of parameters: 40.4% ↓
    - No considerable accuracy degradation
  - Prediction accuracy improvement:
    - Inference speed: 2.7x ↑ Number of parameters: 39.4% ↓
    - GPU memory cost: 45.5% ↓ Accuracy*: ~ 5.5% ↑
  - → The two-stage modifications are effective
- Ablation study: Each modification contributes to the accuracy improvement
- Modifications in this research are more effective on small models (Refer to Appendix 6)
  - As the number of parameters increases, the advantage becomes smaller:
    - Hidden size = 64, Accuracy: ~ 5.5% ↑
    - Hidden size = 128, Accuracy: ~ 4.0% ↑

* Average of all improved percentages

WASEDA UNIVERSITY

Thanks for Listening

Appx. 4 Detailed Comparison of Parameters

Original model

Number of params: 653,369

name	#params
model	0.7M
local_encoder	0.4M
local_encoder.aa_encoder	86.0K
local_encoder.temporal_encoder	0.2M
local_encoder.al_encoder	76.3K
global_interactor	0.3M
global_interactor.rel_embed	13.4K
global_interactor.global_interactor_layers	0.2M
global_interactor.norm	0.1K
global_interactor.multihead_proj	25.0K
decoder	37.5K
decoder.aggr_embed	8.4K
decoder.loc	8.2K
decoder.scale	8.2K
decoder.pi	12.7K

Proposed model

Number of params: 395,809

name	#params
model	0.4M
local_encoder	0.3M
local_encoder.agt_embed	8.8K
local_encoder.bos_token	1.3K
local_encoder.temporal_encoder	0.1M
local_encoder.al_encoder	63.8K
local_encoder.aa_encoder	62.7K
global_interactor	92.2K
global_interactor.global_interactor_layer	67.1K
global_interactor.multihead_proj	25.1K
decoder	30.4K
decoder.aggr_embed	8.4K
decoder.loc	8.8K
decoder.scale	4.6K
decoder.pi	8.5K

WASEDA UNIVERSITY

Appx. 5 Metrics Comparison

Metrics Comparison

TABLE A5-1: Metrics comparison (Validation set)

Model	minADE	minFDE	MR	minADE	minFDE	MR	brier-minFDE
Original	1.4293	3.1138	0.5022	0.7082	1.0757	0.1117	1.7384
Proposed	**1.3480**	**2.9234**	**0.4875**	**0.6955**	**1.0157**	**0.0984**	**1.6707**
Improve	*5.69%*	*6.11%*	*2.92%*	*1.80%*	*5.57%*	*11.91%*	*3.89%*
#Trajs.	K=1			K=6			

TABLE A5-2: Metrics comparison (Test set)

Model	minADE	minFDE	MR	minADE	minFDE	MR	brier-minFDE
Original	1.7771	3.9306	0.5991	0.8552	1.3672	0.1625	2.0372
Proposed	**1.7161**	**3.7626**	**0.5762**	**0.8349**	**1.2654**	**0.1422**	**1.9338**
Improve	*3.43%*	*4.28%*	*3.82%*	*2.38%*	*7.44%*	*12.49%*	*5.07%*
#Trajs.	K=1			K=6			

WASEDA UNIVERSITY

Appx. 6 Comparison with Other Methods

Comparison with other methods on the leaderboard [8]

TABLE A6: Comparison with other methods on the leaderboard (Test set, K=6)

Model	**minADE**	**minFDE**	**MR**	**brier-minFDE**	**#Params.**
LaneGCN	0.8679	1.3640	0.1634	2.0585	3,701K
DenseTNT	0.8817	1.2815	0.1258	1.9759	1,103K
GOHOME	0.9425	1.4503	0.1048	1.9834	400K
HOME + GOHOME	0.8904	1.2919	0.0846	1.8601	5,100K
Scene Transformer	0.8026	1.2321	0.1255	1.8868	15,296K
HiVT-64	0.8552	1.3672	0.1625	2.0372	653K
HiVT-128	0.8209	1.2800	0.1482	1.9493	2,560K
Proposed	0.8349	1.2654	0.1422	1.9338	396K
Proposed-128	0.8140	1.2213	0.1326	1.8909	1,553K

Hidden size = 64

Hidden size = 128

[8] https://eval.ai/web/challenges/challenge-page/454/leaderboard/1279

Appx. 7 Detailed Comparisons

Ablation study

TABLE A7-1: Ablation study of the second-stage modifications (Validation set, K=6)

Decoder Mods.	Encoder Mods.	Training Strategy	minADE	minFDE	MR	brier-minFDE	#Param.
			0.7110	1.0608	0.1091	1.7236	389,689
✓			0.7042	1.0346	0.1027	1.6935	391,117
✓	✓		0.7041	1.0295	0.1005	1.6852	395,809
✓	✓	✓	**0.6955**	**1.0157**	**0.0984**	**1.6707**	395,809

Validation of edge attributes fusion

TABLE A7-2: Comparison with the original fusion method (Validation set)

Model	K=1			K=6			
	minADE	minFDE	MR	minADE	minFDE	MR	brier-minFDE
Original	1.3732	2.9864	0.4913	0.7018	1.0280	0.1010	1.6868
Proposed	**1.3480**	**2.9234**	**0.4875**	**0.6955**	**1.0157**	**0.0984**	**1.6707**