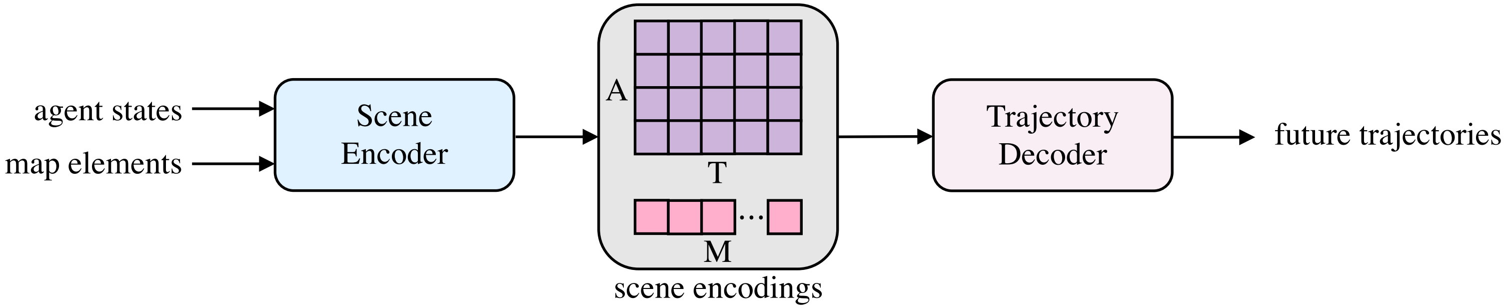
Query-Centric Trajectory Prediction

Zikang Zhou [1, 2]   Jianping Wang [1, 2]   Yung-Hui Li [3]   Yu-Kai Huang [4]

[1] City University of Hong Kong   [2] City University of Hong Kong Shenzhen Research Institute
[3] Hon Hai Research Institute   [4] Carnegie Mellon University

THU-AM-132

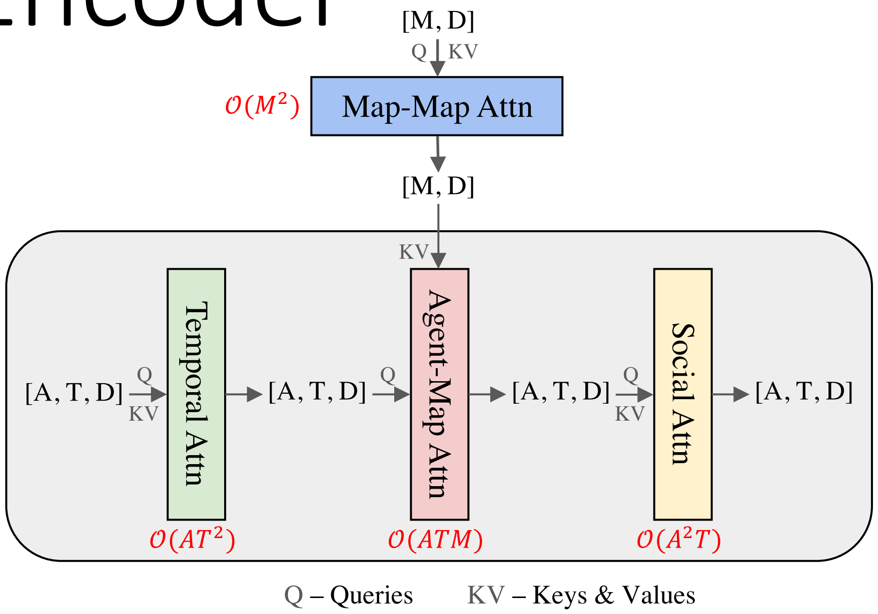
A typical Framework for Trajectory Prediction



Scene encodings

- Agent encodings of shape [A, T, D] – A agents x T historical time steps x D hidden dimensions
- Map encodings of shape [M, D] – M map elements x D hidden dimensions

Existing Factorized Attention-Based Scene Encoder



- Map encodings of shape [M, D]
  - Self-attention among map polygons $\mathcal{O}(M^2)$

    Query: map polygons; Key & Value: map polygons
- Agent encodings of shape [A, T, D]
  - Temporal self-attention for each agent $\mathcal{O}(AT^2)$

    Query: agent states; Key & Value: agent states of the same agent
  - Agent-map cross-attention for each agent at each past time step $\mathcal{O}(ATM)$

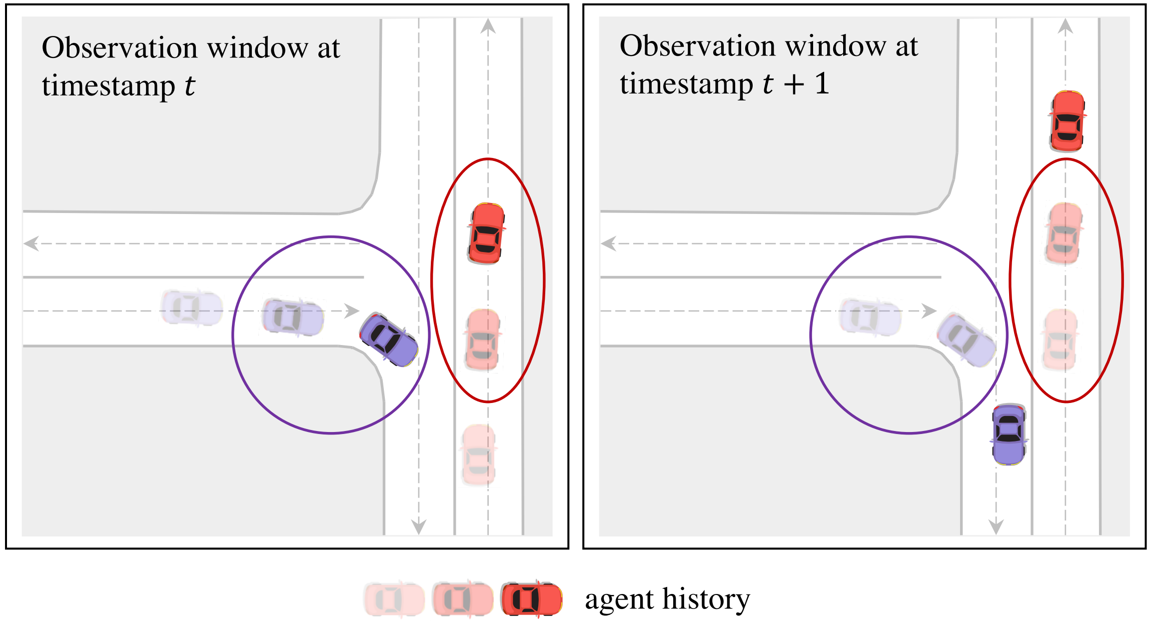
    Query: agent states; Key & Value: map polygons
  - Social self-attention among agents at each past time step $\mathcal{O}(A^2T)$

    Query: agent states; Key & Value: agent states at the same time step

Limited scalability: each layer in the factorized attention has **cubic** complexity

Is it possible to reduce the inference latency while enjoying the representational power of factorized attention?

Key Observation: Trajectory Prediction is a Streaming Processing Task

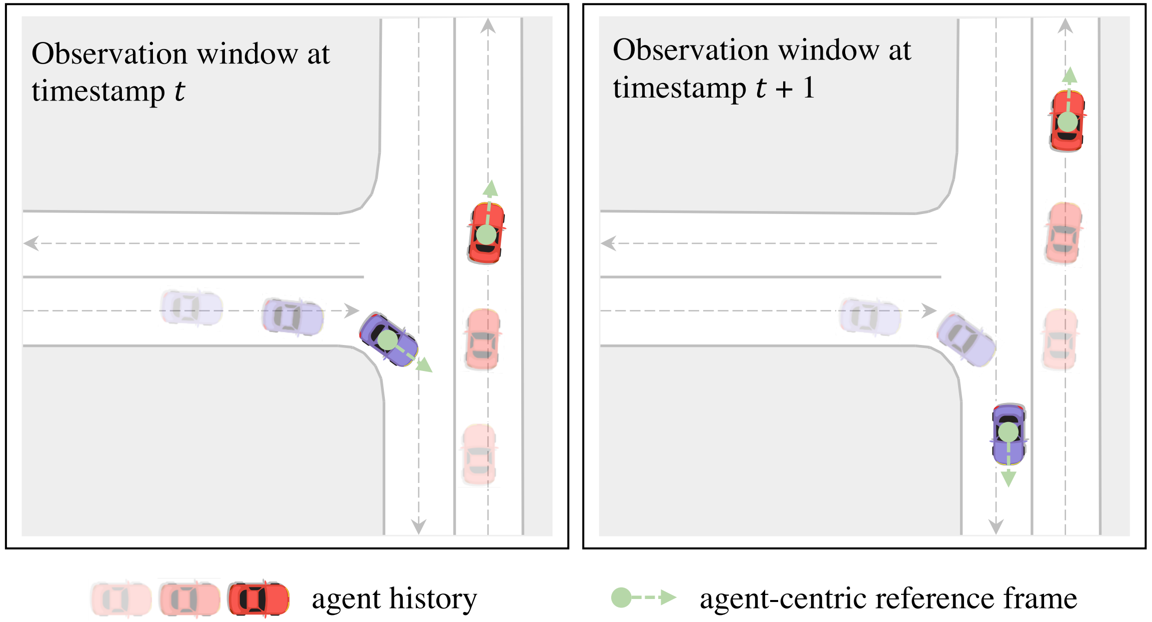


The latest observation window has T - 1 time steps overlapping with its predecessor

Can we reuse the overlapped time steps' encodings computed in previous observation windows after the observation window slides forward?

Example: observation windows with 2 agents and 3 historical time steps. From timestamp t to timestamp t+1, the prediction module updates the data buffer by: (1) dropping the oldest agent states of the two agents; and (2) adding the latest agent states that arrive at timestamp t. As a result, the two consecutive observation windows have two overlapped time steps. On the other hand, the map polygons in the two consecutive observation windows also largely overlap.

Obstacle: Input Normalization Required by Existing Approaches

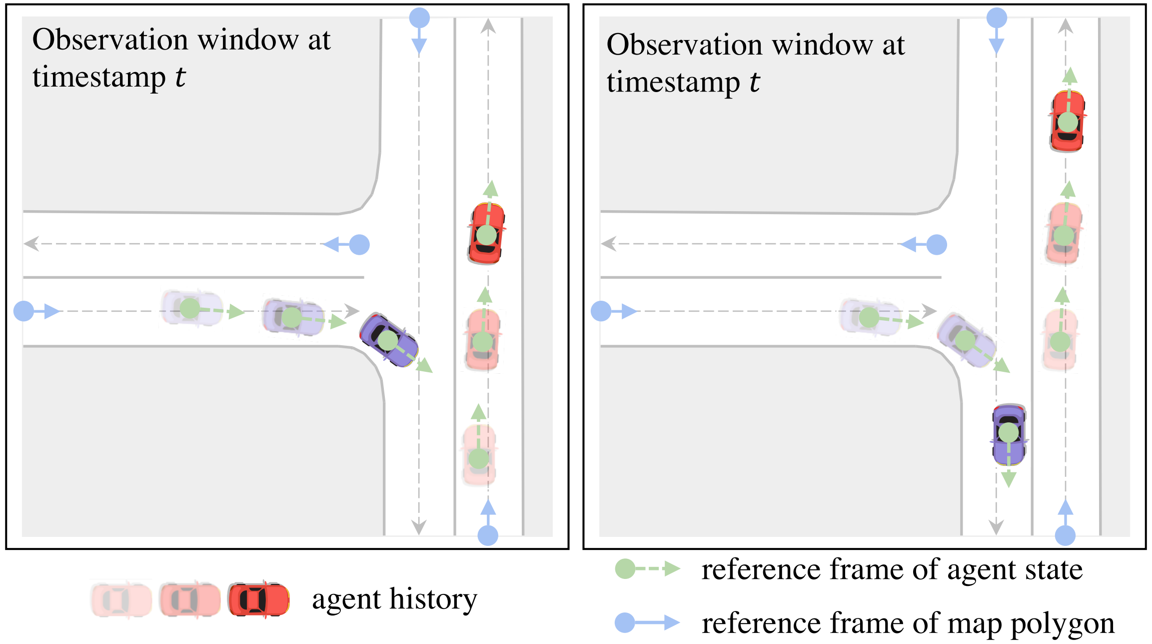


	minADE (↓)	minFDE (↓)	MissRate (↓)
w/o normalization	1.09	1.89	0.26
w/ normalization	**0.69**	**1.04**	**0.10**

Zhou, Zikang, et al. HiVT: Hierarchical Vector Transformer for Multi-Agent Motion Prediction. *CVPR* 2022.

- Existing methods use agent-centric reference frames to achieve **viewpoint invariance**
  - Each agent's historical states are normalized in the agent's local reference frame, which is determined by the agent's position and heading at the ***current*** time step
  - Each map element is copied A times, and each copy is normalized in one agent's local reference frame
  - Each time the observation window slides forward, the "current time step" also shifts accordingly, and all inputs need to be re-normalized based on the up-to-date reference frames
  - Due to the variation in input, we're forced to re-compute all time steps' encodings even though the observation windows largely overlap

Solution: Query-Centric Reference Frame for Invariant Scene Encodings



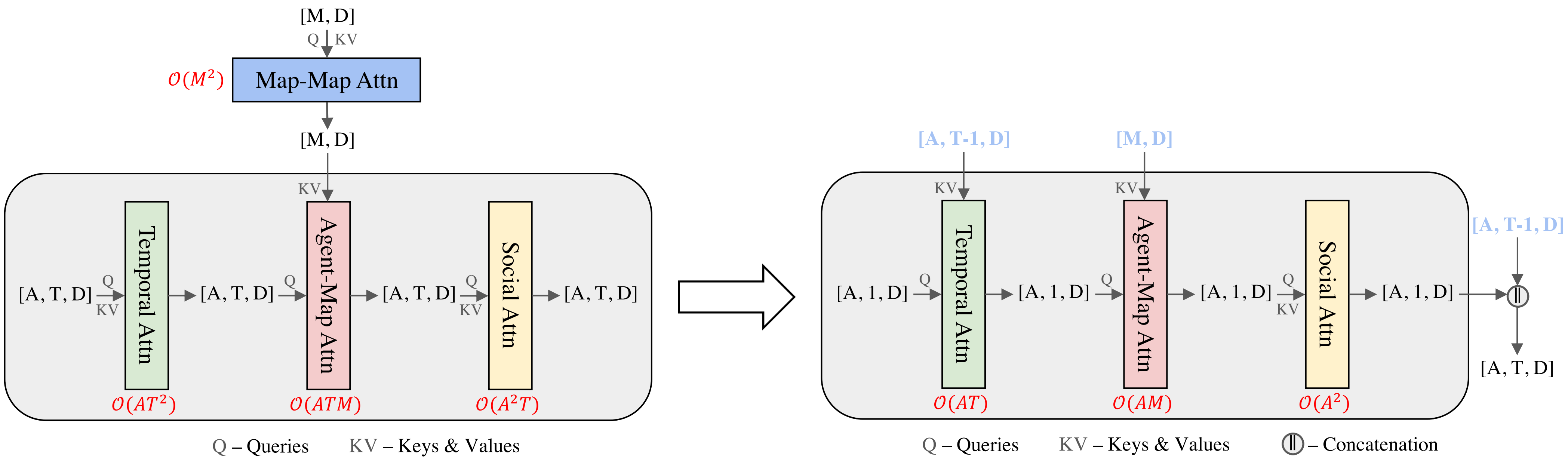
1. Set up a local spacetime coordinate system for each agent state and each map element that a query vector will derive from

2. Encode query elements' inputs in local reference frames
   - Inputs are independent of the global reference frame -> avoid input re-normalization -> encodings are always invariant -> enable reusing the encodings computed in previous observation windows

3. Compute relative spatial-temporal positional embeddings
   - Relative distance, relative direction, relative orientation, time difference

4. Inject the relative positional embeddings into the key and value elements in the attention layers
   - To help the attention layers be aware of the difference between local reference frames

A More Efficient Factorized Attention-Based Scene Encoder with Streaming Processing



$$\mathcal{O}(M^2 + AT^2 + ATM + A^2T) \Longrightarrow \mathcal{O}(AT + AM + A^2)$$

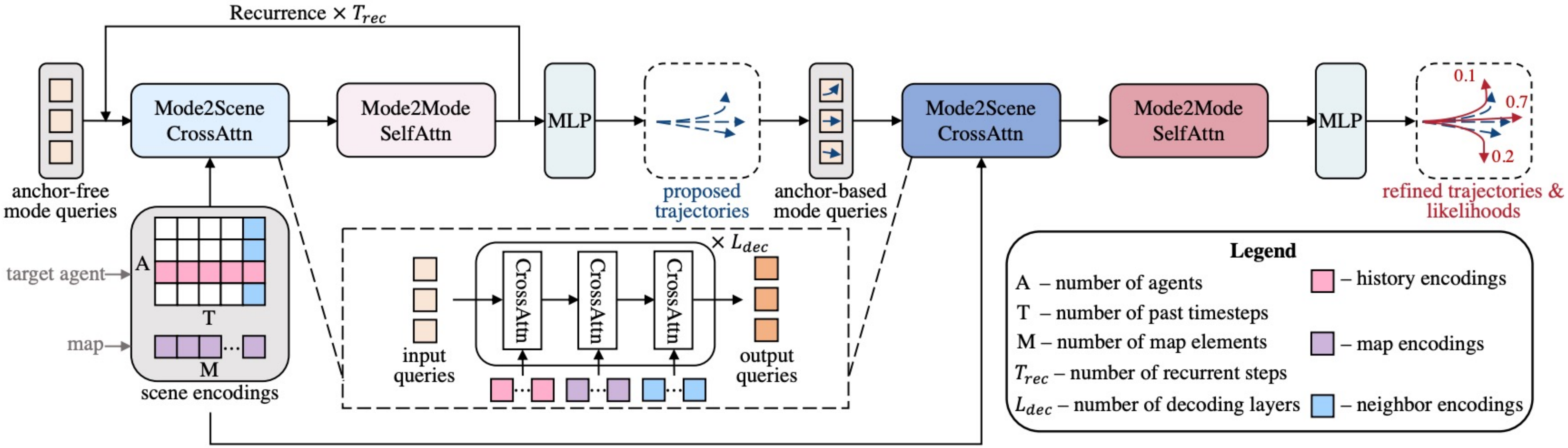
- Cache and reuse the static map encodings (tensor of shape [M, D]) pre-computed offline
- Cache and reuse the agent encodings (tensors of shape [A, T-1, D]) computed in previous observation windows
  - When a new data frame arrives, the model only performs factorized attention for the A incoming agent states

Experimental Results

Model	Online Inference (ms)		$\text{minADE}_6\downarrow$	$\text{minFDE}_6\downarrow$	$\text{MR}_6\downarrow$
	w/o reuse	w/ reuse			
QCNet ($L_{\text{enc}} = 0$)	8±1	1±0	0.76	1.33	0.18
QCNet ($L_{\text{enc}} = 1$)	64±1	10±1	0.74	1.30	0.17
QCNet ($L_{\text{enc}} = 2$)	82±1	13±1	**0.73**	**1.27**	**0.16**

- More encoding layers -> better performance & slower inference
- Caching and reusing the previously computed encodings reduces the inference latency drastically without affecting the prediction performance

Decoding Pipeline



- DETR-like decoder: trajectory proposal + trajectory refinement
- A recurrent, anchor-free proposal module for generating adaptive trajectory anchors
- An anchor-based module for refining the proposed trajectory anchors

Ablation Study

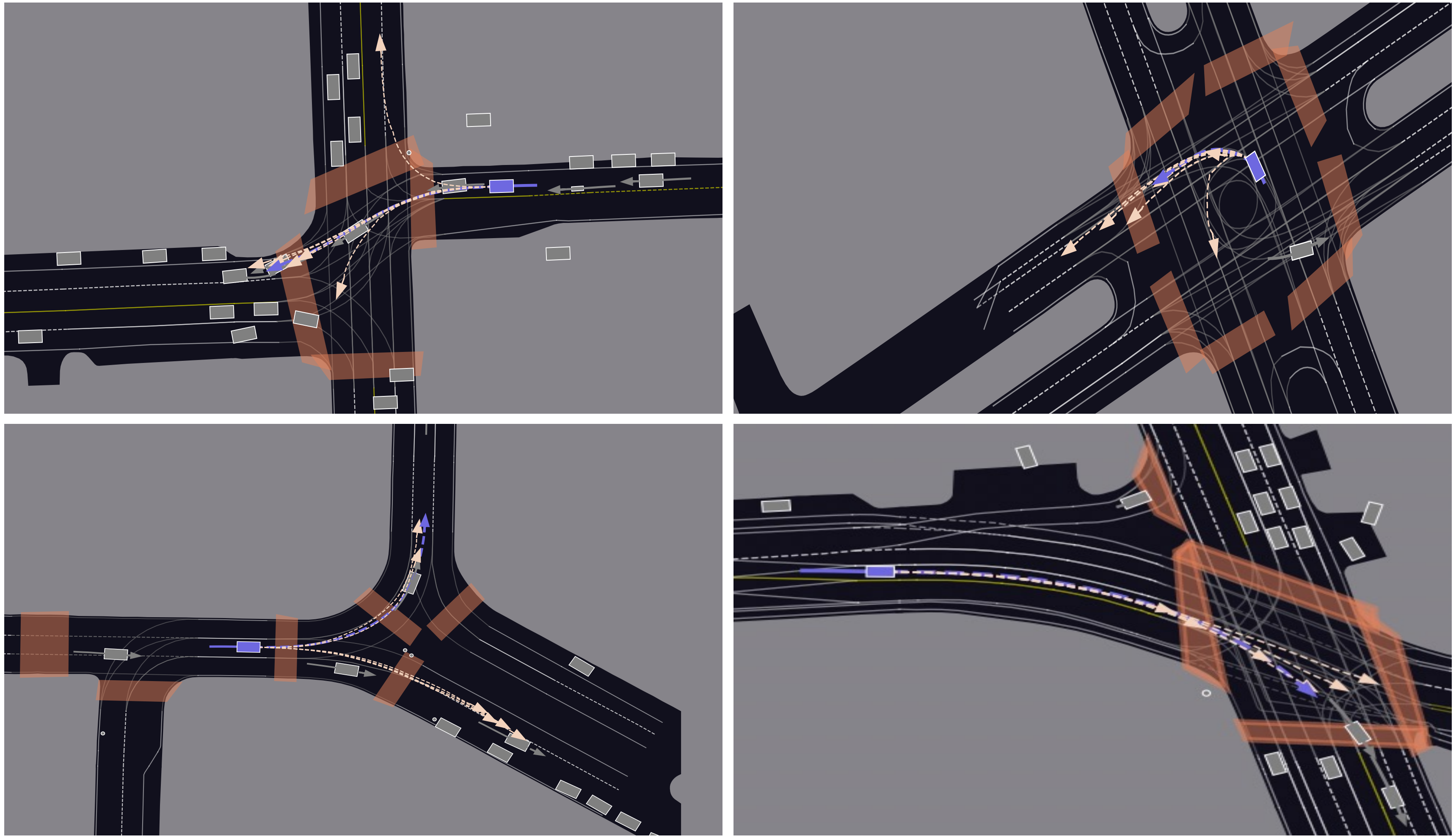
Dataset	#Recurrent Step	Refinement	b-minFDE$_6$↓	minFDE$_6$↓	MR$_6$↓
Argoverse 1 (3-sec pred.)	1 (3 sec/step)	×	1.58	0.92	0.09
	2 (1.5 sec/step)	×	1.57	0.90	0.08
	3 (1 sec/step)	×	1.56	0.90	0.08
	3 (1 sec/step)	✓	**1.55**	**0.89**	0.08
Argoverse 2 (6-sec pred.)	1 (6 sec/step)	×	2.10	1.47	0.20
	2 (3 sec/step)	×	2.04	1.42	0.19
	3 (2 sec/step)	×	2.02	1.40	0.19
	3 (2 sec/step)	✓	**1.90**	**1.27**	**0.16**
	6 (1 sec/step)	✓	**1.90**	**1.27**	**0.16**

SOTA Performance on Argoverse 1 & 2

Method	**b-minFDE$_6$**↓	minADE$_6$↓	minFDE$_6$↓	MR$_6$↓
LaneGCN [31]	2.06	0.87	1.36	0.16
mmTransformer [32]	2.03	0.84	1.34	0.15
DenseTNT [19]	1.98	0.88	1.28	0.13
TPCN [50]	1.93	0.82	1.24	0.13
SceneTransformer [38]	1.89	0.80	1.23	0.13
HOME+GOHOME [15, 16]	1.86	0.89	1.29	**0.08**
HiVT [56]	1.84	0.77	1.17	0.13
MultiPath++ [46]	1.79	0.79	1.21	0.13
GANet [48]	1.79	0.81	1.16	0.12
PAGA [11]	1.76	0.80	1.21	0.11
DCMS [51]	1.76	0.77	1.14	0.11
Wayformer [37]	1.74	0.77	1.16	0.12
Ours	**1.69**	**0.73**	**1.07**	0.11

Method	**b-minFDE$_6$** ↓	minADE$_6$ ↓	minFDE$_6$ ↓	MR$_6$ ↓	minADE$_1$ ↓	minFDE$_1$ ↓	MR$_1$ ↓
THOMAS [17]	2.16	0.88	1.51	0.20	1.95	4.71	0.64
GoRela [9]	2.01	0.76	1.48	0.22	1.82	4.62	0.66
MTR [42]	1.98	0.73	1.44	<u>0.15</u>	1.74	4.39	<u>0.58</u>
GANet [48]	1.96	0.72	1.34	0.17	1.77	4.48	0.59
QML* [43]	1.95	0.69	1.39	0.19	1.84	4.98	0.62
BANet* [54]	1.92	0.71	1.36	0.19	1.79	4.61	0.60
QCNet (w/o ensemble)	<u>1.91</u>	<u>0.65</u>	<u>1.29</u>	0.16	<u>1.69</u>	<u>4.30</u>	0.59
QCNet (w/ ensemble)	**1.78**	**0.62**	**1.19**	**0.14**	**1.56**	**3.96**	**0.55**

Qualitative Results



Summary

- A query-centric encoding paradigm that enables streaming scene encoding and parallel multi-agent trajectory decoding
- A query-based decoding pipeline for multimodal and long-term prediction, which consists of a recurrent, anchor-free trajectory proposal module and an anchor-based refinement module