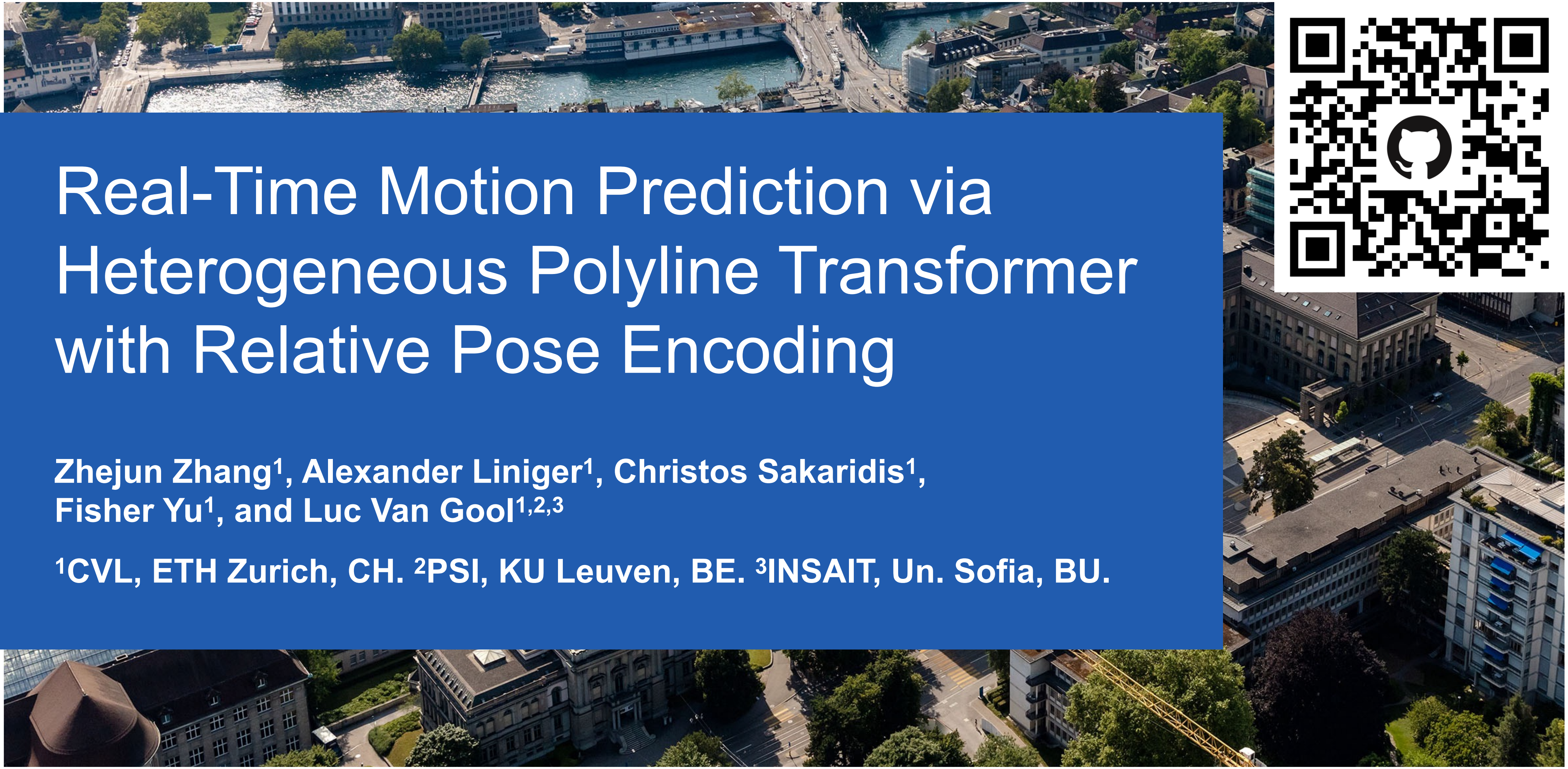
NEURAL INFORMATION PROCESSING SYSTEMS

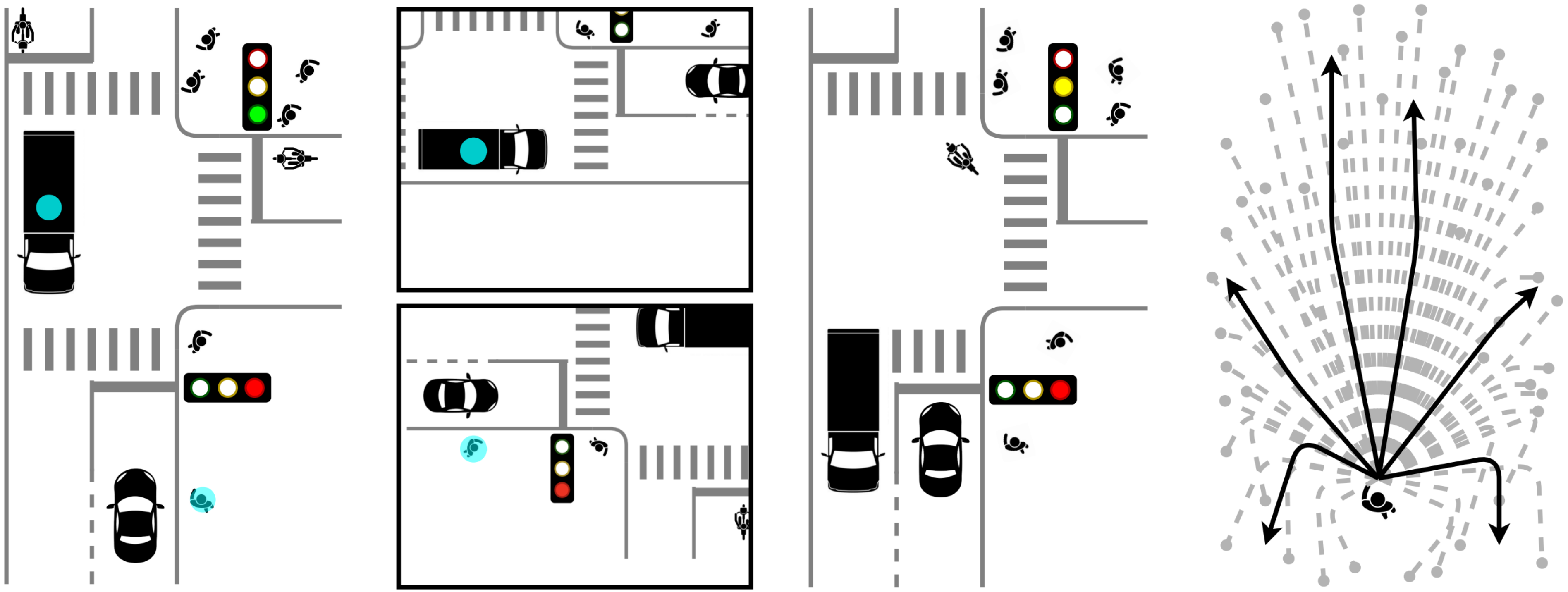


Real-Time Motion Prediction via Heterogeneous Polyline Transformer with Relative Pose Encoding

**Zhejun Zhang[1], Alexander Liniger[1], Christos Sakaridis[1],
Fisher Yu[1], and Luc Van Gool[1,2,3]**

[1]CVL, ETH Zurich, CH. [2]PSI, KU Leuven, BE. [3]INSAIT, Un. Sofia, BU.

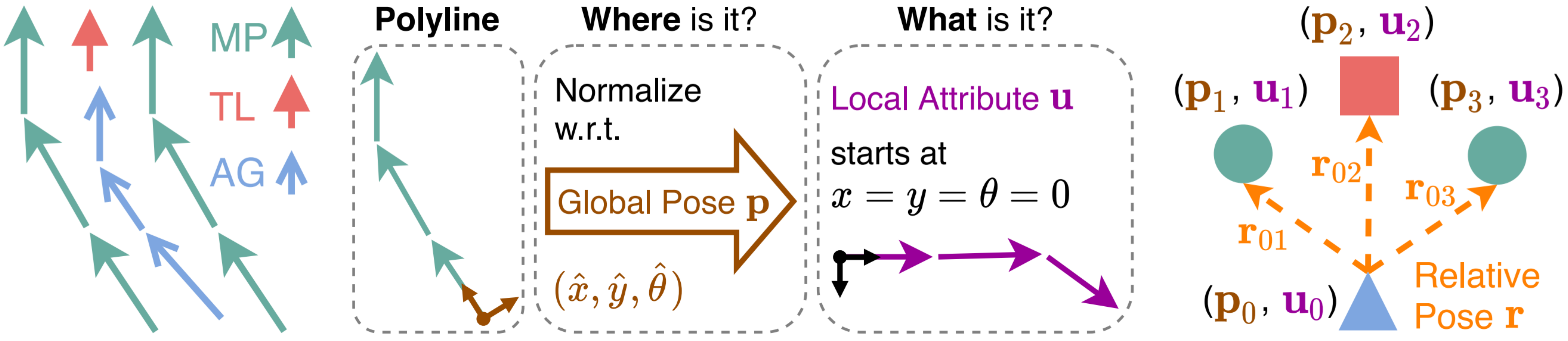
Motivation



(a) Dense traffic scenario (b) Agent-centric ROIs (c) Online inference (d) Trajectory aggregation

a) **Real-Time and On-Board** motion prediction in urban driving scenario with dense traffic.
b) **Agent-Centric SoTA**: Good performance. Bad scalability.
c) **Online Inference** with streaming inputs.
d) **Expensive Post-Processing** and ensembling.

Pairwise-Relative Polyline Representation



MP
TL
AG
Polyline
Where is it?
Normalize w.r.t.
Global Pose $\mathbf{p}$
$(\hat{x}, \hat{y}, \hat{\theta})$
What is it?
Local Attribute $\mathbf{u}$
starts at
$x = y = \theta = 0$
$(\mathbf{p}_2, \mathbf{u}_2)$
$(\mathbf{p}_1, \mathbf{u}_1)$
$(\mathbf{p}_3, \mathbf{u}_3)$
$\mathbf{r}_{02}$
$\mathbf{r}_{03}$
$\mathbf{r}_{01}$
$(\mathbf{p}_0, \mathbf{u}_0)$
Relative Pose $\mathbf{r}$

(a) Polylines. (b) Local attribute represented in the global pose's frame. (c) Obtain relative poses.

Global Pose p: Where is the polyline?

Local Attribute u: What kind of polyline is it?

Input to the Network:

a) **High-dimensional** local attribute **u**. Shared.

b) **3-dimensional** relative pose **r**. Computed from p.

Good Performance: Rotation and translation invariance.

Good Scalability: sharing high-dimensional **u.**

However, so far it is only exploited by GNNs.

KNARPE: **K**-nearest **N**eighbor **A**ttention with **R**elative **P**ose **E**ncoding

Relative Pose Encoding: $\mathrm{RPE}(\mathbf{r}_{ij}) = \mathrm{concat}(\mathrm{PE}(x_{ij}), \mathrm{PE}(y_{ij}), \mathrm{AE}(\theta_{ij})),$

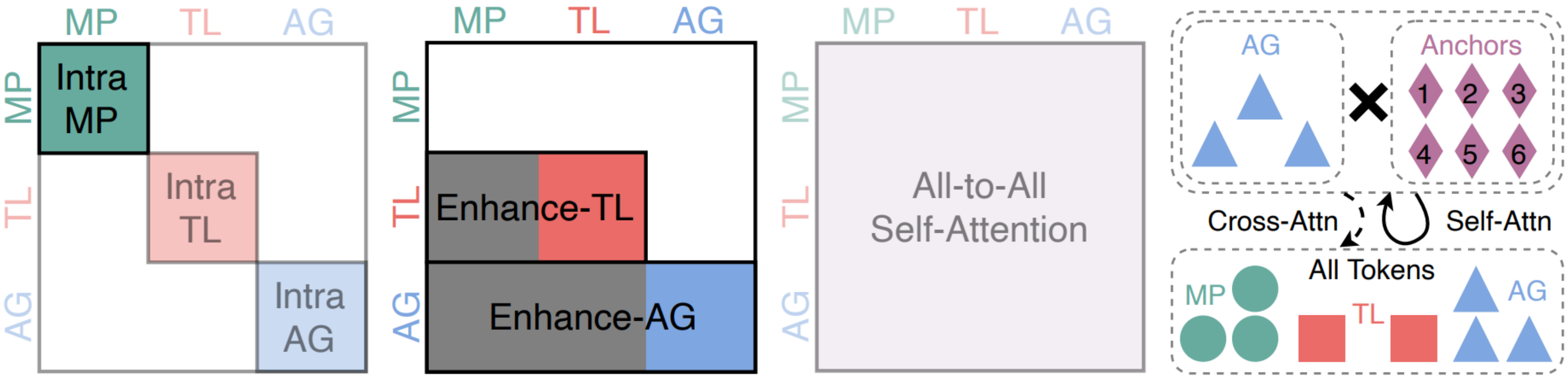
K-Nearest Neighbor: $j \in \kappa_i^K$

$$\mathbf{z}_i = \boxed{\textsc{Knarpe}}\left(\mathbf{u}_i, \mathbf{u}_j, \mathbf{r}_{ij} \mid j \in \kappa_i^K\right) = \sum_{\boxed{j \in \kappa_i^K}} \alpha_{ij} \left(\mathbf{u}_j \mathbf{W}^v + \mathbf{b}^v + \boxed{\mathrm{RPE}(\mathbf{r}_{ij}) \hat{\mathbf{W}}^v + \hat{\mathbf{b}}^v}\right),$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k \in \kappa_i^K} \exp(e_{ik})}, \quad e_{ij} = \frac{(\mathbf{u}_i \mathbf{W}^q + \mathbf{b}^q)(\mathbf{u}_j \mathbf{W}^k + \mathbf{b}^k + \boxed{\mathrm{RPE}(\mathbf{r}_{ij}) \hat{\mathbf{W}}^k + \hat{\mathbf{b}}^k})}{\sqrt{D}},$$

- Based on multi-head dot-product attention.
- Implemented with **basic matrix operations**: indexing, summation and element-wise multiplication.
- Self-Attention: Local context aggregation like CNN.
- Cross-Attention: Rotated ROI alignment with CNN.

HPTR: **H**eterogeneous **P**olyline **T**ransformer with **R**elative Pose Encoding



(a) Intra-class TF encoder (b) Inter-class TF decoder (c) All-to-all TF encoder (d) AC-to-all TF decoder

- Based on KNARPE.
- Transformers are organized in a hierarchical way.
- Remove redundant attentions.
- Intermediate results can be cached and reused.
- Asynchronous token update during online inference.
- Maps: Day. Traffic Lights: Second. Agents: Millisecond.

Results on Public Leaderboards

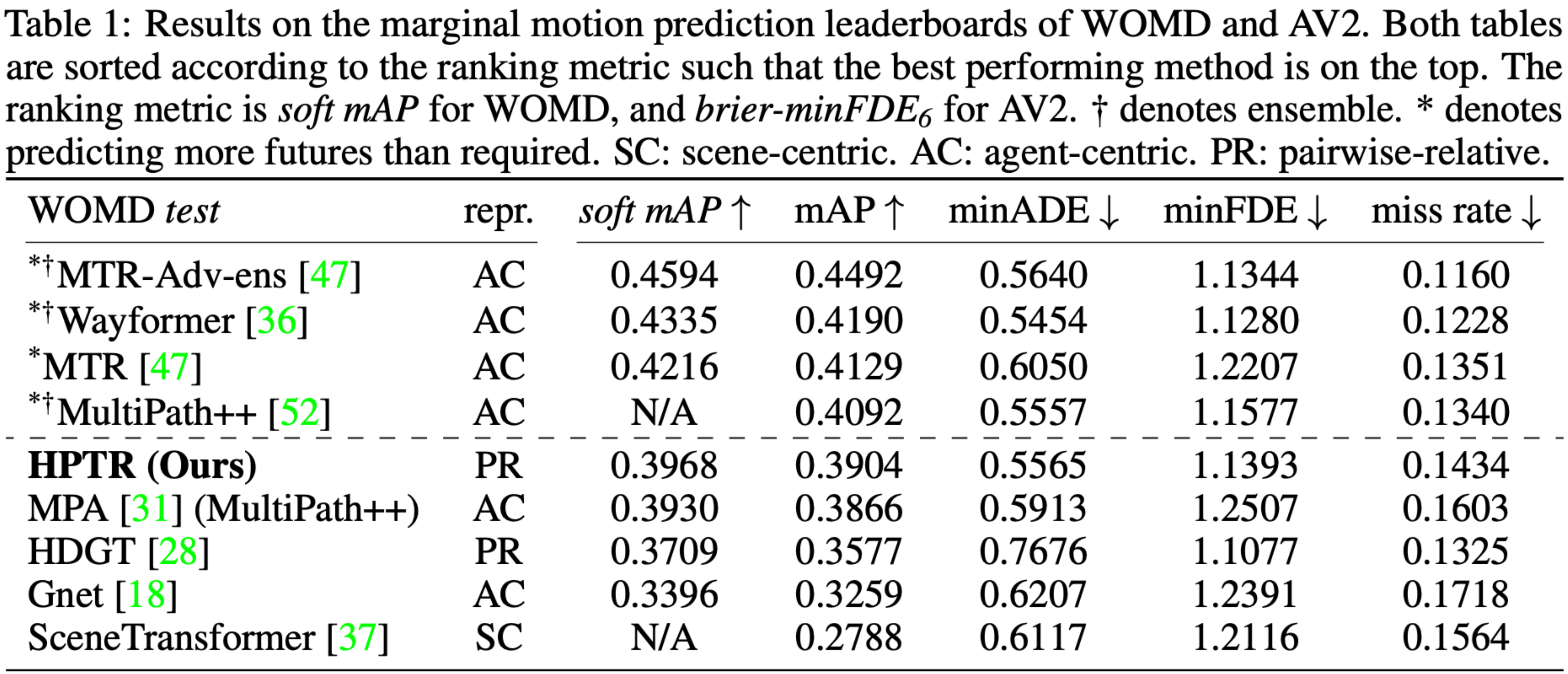


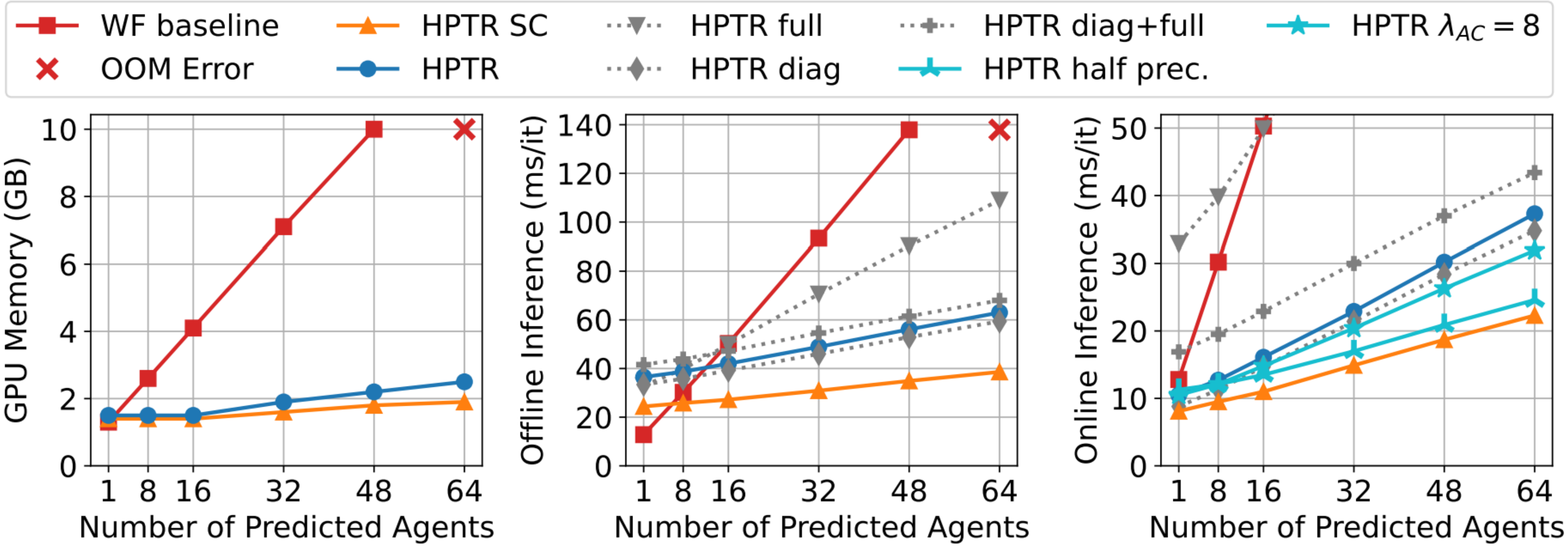
Table 1: Results on the marginal motion prediction leaderboards of WOMD and AV2. Both tables are sorted according to the ranking metric such that the best performing method is on the top. The ranking metric is *soft mAP* for WOMD, and *brier-minFDE$_6$* for AV2. † denotes ensemble. * denotes predicting more futures than required. SC: scene-centric. AC: agent-centric. PR: pairwise-relative.

WOMD *test*	repr.	*soft mAP* ↑	mAP ↑	minADE ↓	minFDE ↓	miss rate ↓
*†MTR-Adv-ens [47]	AC	0.4594	0.4492	0.5640	1.1344	0.1160
*†Wayformer [36]	AC	0.4335	0.4190	0.5454	1.1280	0.1228
*MTR [47]	AC	0.4216	0.4129	0.6050	1.2207	0.1351
*†MultiPath++ [52]	AC	N/A	0.4092	0.5557	1.1577	0.1340
HPTR (Ours)	PR	0.3968	0.3904	0.5565	1.1393	0.1434
MPA [31] (MultiPath++)	AC	0.3930	0.3866	0.5913	1.2507	0.1603
HDGT [28]	PR	0.3709	0.3577	0.7676	1.1077	0.1325
Gnet [18]	AC	0.3396	0.3259	0.6207	1.2391	0.1718
SceneTransformer [37]	SC	N/A	0.2788	0.6117	1.2116	0.1564

SoTA performance on Waymo Open Motion Dataset among the end-to-end methods (no ensemble, no redundant prediction).

Also, on Argoverse 2 (c.f. our paper) our method compares favorably against other end-to-end methods.

Efficiency and Scalability



- HPTR SC: The scene-centric baseline. The efficiency upper-bound.
- WF (Wayformer) baseline: The agent-centric baseline. SoTA performance.
- Efficiency gain **without sacrificing** the performance.
  - **80%** less GPU memory consumption and online inference latency.
  - **60%** less offline inference latency.
  - **40 FPS** during online inference with 64 agents by using half precision and caching the map.

Summary

- **KNARPE** allows the pairwise-relative representation to be used by Transformers.
- **HPTR** uses hierarchical architecture to enable asynchronous token update.
- **SoTA** performance among E2E methods: WOMD and AV2 dataset.
- **Good Performance and Good Scalability.**
    - As accurate as agent-centric methods.
    - As efficient as scene-centric methods.
- **Real-Time and On-Board** Motion Prediction.
    - 40 FPS during online inference with 64 agents.
    - 80% reduction on online inference latency and GPU memory.
- **Code**: https://github.com/zhejz/HPTR