Supplementary Materials for GoRela: Go Relative for Viewpoint-Invariant Motion Forecasting

Alexander Cui, Sergio Casas, Kelvin Wong, Simon Suo, Raquel Urtasun
Waabi, University of Toronto
{acui, sergio, kwong, ssuo, urtasun}@waabi.ai

This document presents additional experiments in Section I that support the claims in the main paper: improved robustness to rare SDV states, improved sample efficiency, improved runtime as well as providing additional insights via qualitative comparisons on both urban and highway scenarios. Additionally, implementation details are provided in Section II for improved reproducibility.

I. Additional experiments

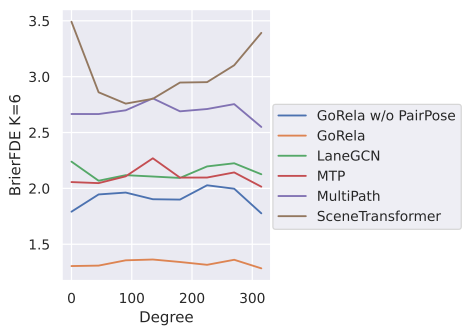
Viewpoint sensitivity analysis: Self-driving is a safety-critical application. As such, it is important to evaluate the robustness of autonomy components with respect to rarely seen vehicle states. In this experiment, we simulate the self-driving vehicle (SDV) in rare states by shifting its heading from its original pose recorded in the log (i.e., rotating the scene around the ego vehicle centroid). We divide the scenarios in the validation set of Argoverse 2 into 8 rotation buckets $\{[0°, 45°), [45°, 90°), \ldots, [315°, 360°)\}$ In particular, Fig. 1-Left shows the bucketed BrierMinFDE@K=6. We can observe that GoRela's performance is almost constant across buckets. We believe the small offsets are due to different scenarios being on every bucket. On the other hand, the baselines (MultiPath[1], LaneGCN[2], MTP[3], Multipath [1], SceneTransformer[4]) have a much higher variability over different rotation buckets. To make sure this effect comes from the rotation invariance of our method and not our high-level architecture, we include the ablation of GoRela without PairPose, which preserves the same architecture but doesn't use relative positional encoding on the edges of the graphs. We include a box plot in Fig. 1-Right, which directly displays the variance across buckets. Finally, we showcase this qualitatively in Fig. 2. It is obvious that the baselines' predictions when performing inference on the original vs. the rotated scenarios are very inconsistent. In contrast, GoRela's predictions are exactly the same thanks to its viewpoint invariance.

Sample efficiency curves: Fig. 3 showcases the sample efficiency of the different methods. In this experiment, we train GoRela and some interesting baselines for different training set sizes (we do not train all of them as these experiments are computationally costly). We can observe that GoRela converges much faster than the baselines, attaining better performance than the baselines even when using 90% less data (only 20,000 examples). In other words, the baselines are a lot more data hungry, and it seems that they would greatly benefit from using more than 200,000 examples. Sample efficiency in motion forecasting is very important since labeling these datasets for supervised learning is very costly. We also ablate the importance of PairPose by removing it from our model, which makes it very clear that our viewpoint-invariant approach with pair-wise relative positional encodings is really the key to this improvement in sample efficiency. Intuitively, this result aligns with our expectations, since the rotation and translation invariance of our method shrinks the input space quite a bit, making scenarios in the validation set look more similar to those in the training set.

Runtime analysis: In this experiment, we measure the benefit of a shared scene encoding for all agents as achieved by GoRela against processing the scene separately for each agent (batched in GPU). We measure how the encoder runtime scales with number of agents in Fig. 4 as well as map nodes in the graph in Fig. 5. The encoder here includes all up to the heterogeneous scene encoder (inclusive). We can see that our model has nearly constant runtime regardless of the number of actors due to our shared scene encoding. This is because the number of agents and their inbound and outbound connections in the graph is relatively low compared to the map nodes. On the other hand, the per-actor scene encoding increases the runtime linearly to 7x with 40x more actors. We can see that both models scale relatively slowly with the number of lane graph nodes, with per-actor inference starting with a much higher fixed cost. Overall, these results demonstrate the runtime benefits of using a shared scene encoding. We do not compare the decoder runtime between the two approaches as it would be the same given that GoRela's decoder acts on a graph composed of one connected component per agent.

Graph layer ablation: Table I shows the results of the ablation of heterogeneous message passing (HMP) as our graph convolutional layer. This layer is used in the heterogeneous scene encoder and per-actor goal decoder. We based our experiment on the public HEAT [5] and the GATv2 [6] implementations in [7]. We show that our model outperforms these state-of-the-art layers in both multimodal ($K = 6$) and unimodal error.

Qualitative comparisons in Argoverse 2: Fig. 6 shows



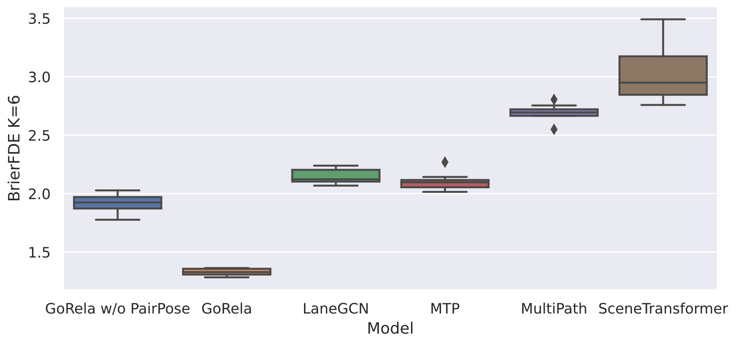


Fig. 1: **Quantitative analysis on viewpoint sensitivity** by randomly rotating scenes around the SDV. Left: error bucketed by rotation bins. Right: boxplot of error (variance measures the variability across different rotation bins.)

Graph layer [†]	BrierFDE K=6	FDE K=6	FDE K=1
GATv2 [6]	1.68	1.32	3.42
HEAT [5]	1.52	1.17	2.98
HMP (ours)	**1.45**	**1.08**	**2.77**

TABLE I: **Ablation study** of graph convolutional layers on Argoverse 2 (val). We ablate HMP with other state-of-the-art graph convolution layers. [†] Models trained on 25% of the training set.

a qualitative comparison between GoRela and the baselines for 8 different scenarios. We can clearly see that our model achieves a much better map understanding, showcased by predictions that follow the lanes well. In contrast, the baselines predict some unrealistic modes for most scenarios that are not compliant with the traffic rules and do not align well with typical driving behavior. As shown by the quantitative experiments in the main paper, our model's best mode attains lower error than the baselines. The qualitative results show that our model is able to achieve this while predicting lower diversity modes. Given the analysis carried out by [8], which highlights the importance of precision for downstream motion planning, we hypothesize this would be an important characteristic for safe and comfortable driving.

Qualitative comparisons in HighwaySim: Fig. 7 displays predictions in highway environments. In the first scenario, the highlighted agent is taking the fork lane on its left as shown by the ground truth trajectory in gray. All baselines predict the agent is going to continue straight on its original lane. In contrast, GoRela is able to predict a bi-modal distribution where we believe the agent might lane change or continue straight. The second scenario highlights an agent on an on-ramp (i.e., a lane merging into the highway), that is following another agent which is yielding to highway traffic. All baselines do not understand this interaction well enough, and they predict trajectories that collide with the ground truth future motion of the lead vehicle. Our model understands this interaction and is able to predict that the highlighted agent will brake and stay behind its lead vehicle.

II. Implementation details

In this section we provide implementation details for improved reproducibility.

Input features: For the pair-wise relative positional encoding (PairPose), we use 16 frequencies for each of sine and cosine. For each timestep t in agent history encoder inputs, we calculate the PairPose using the pose at t and the pose at the present timestep. We also include the finite difference of the PairPose between t and $t-1$, the velocity, bounding box size and boolean for whether the timestep is observed. For each map node input in the map encoder, we include the node length, curvature, degree for each edge type, and boolean of whether it's in an intersection and whether it's part of a crosswalk. We also include left and right lane boundary color, type, and distance from the centerline.

Training: For our multi-task objective, we use an equal weight of 1.0 for goal classification, goal regression and trajectory completion. We use $\gamma = 2.0$ for the goal classification focal loss, and only supervise the closest node to the ground-truth goal in terms of goal regression. For the goal classification and regression, our loss ignores those agents which ground-truth goal is not present in the data (labeled track is shorter than the prediction horizon). For trajectory completion we use the ground-truth goal during training when this is available (teacher forcing). When it is not available in the data, we find it beneficial to use the most likely predicted goal as a weak teacher. For our multi-task loss, we do not supervise agents which trajectory does not stay within 10 meters of the lane-graph boundaries at all time steps within the prediction horizon. Agents outside this region are still taken into account in our heterogeneous graphs, but their corresponding predictions do not contribute to the loss. All agents are weighted equally in the loss, including the focal, scored and unscored agents in Argoverse 2. Since our model is invariant to rotations and translations, we do not require such augmentations during training. However, we find useful to perform a scale augmentation by uniformly sampling the scaling factor between 0.8 and 1.2. Our model was trained for 17 epochs, using 16 GPUs for a total batch size of 64. We use Adam optimizer with a learning rate 1e-5, with a step-wise scheduler with 0.25 decay and 15 step-size.

Greedy goal sampler: We use hard threshold radius $\gamma = 2$, downweight radius $\nu = 4$, and downweight factor $\zeta = 10$.

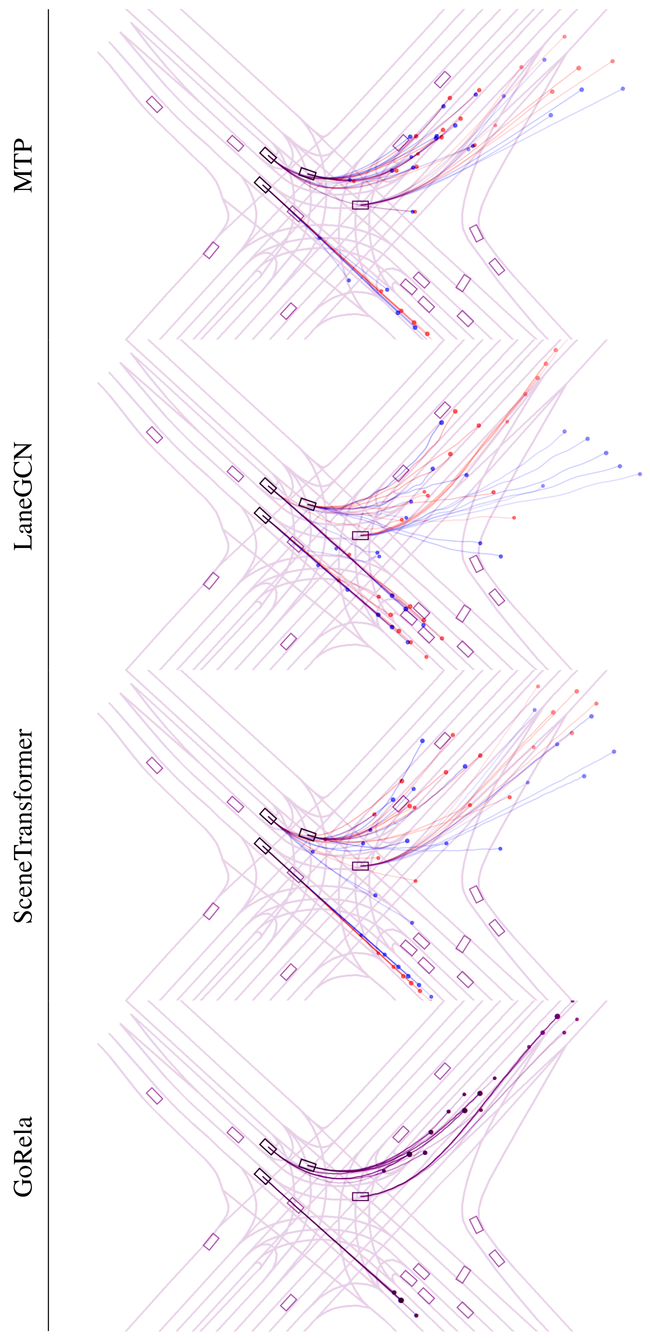


Fig. 2: **Qualitative results on viewpoint sensitivity**. Overlaid predictions when performing inference with the original scenario vs. rotated scenario by 90°.

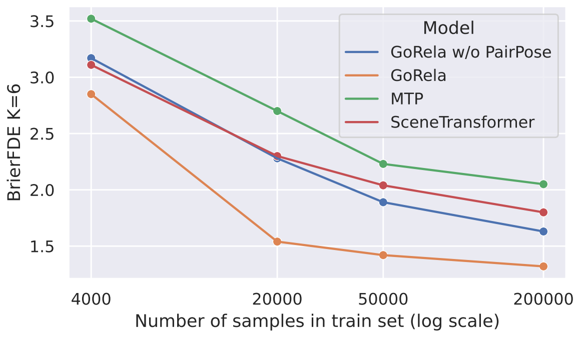


Fig. 3: **Sample efficiency analysis**. We evaluate the performance of GoRela and the baselines when training with different data scales.

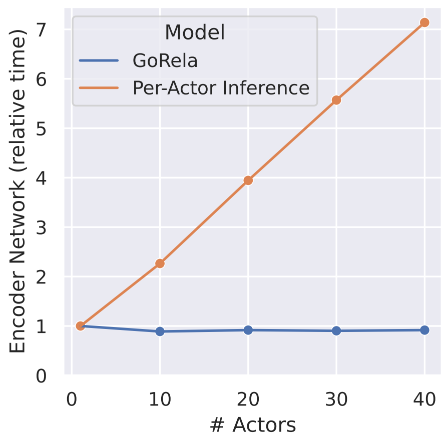


Fig. 4: **Runtime scaling with actors analysis**. We measure how the encoder runtime scales with actors where we share a scene encoding across all actors (GoRela) and where the scene-context is encoded per actor (batched across actors). The runtime is measured relative to the runtime with a single actor. The number of lane graph nodes (250) is held constant.

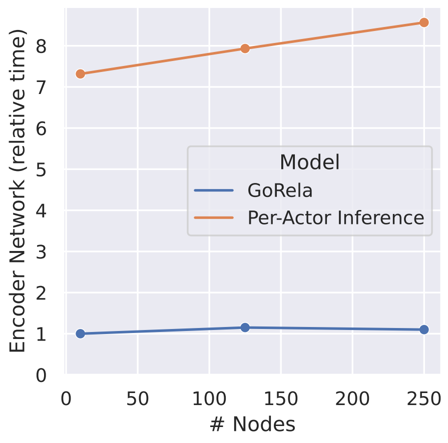


Fig. 5: **Runtime scaling with lane graph nodes analysis**. We measure how the encoder runtime scales with the number of lane graph nodes, where we share a scene encoding across all actors (GoRela) and where the scene-context is encoded per actor (batched across actors). The runtime is measured relative to the runtime with 10 nodes using shared scene encoding. The number of actors (40) is held constant.

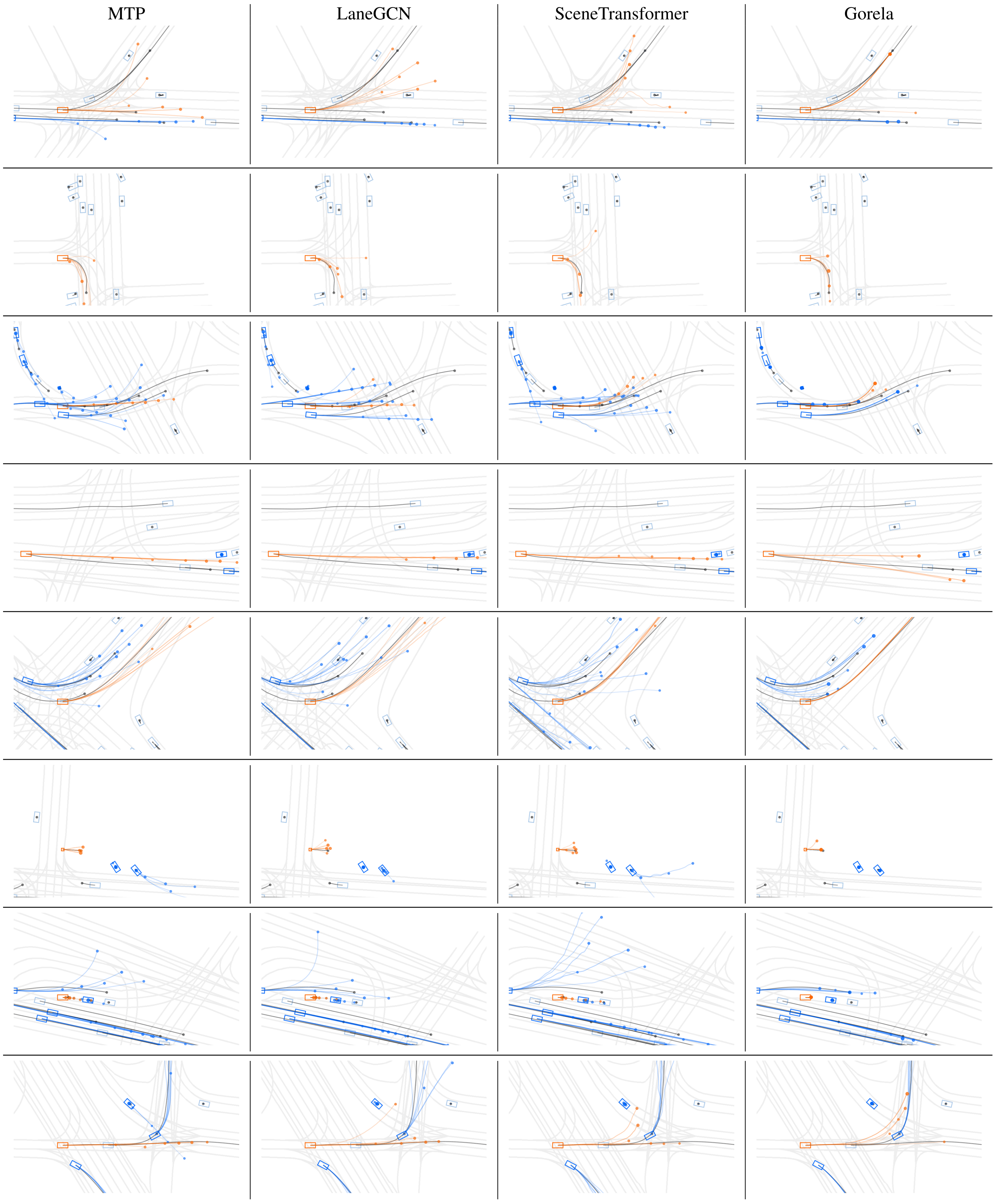


Fig. 6: **Qualitative comparison in Argoverse 2**. 6 seconds long multi-modal motion forecasts for focal and scored agents. Ground-truth plotted in gray. Every row showcases a different scenario.

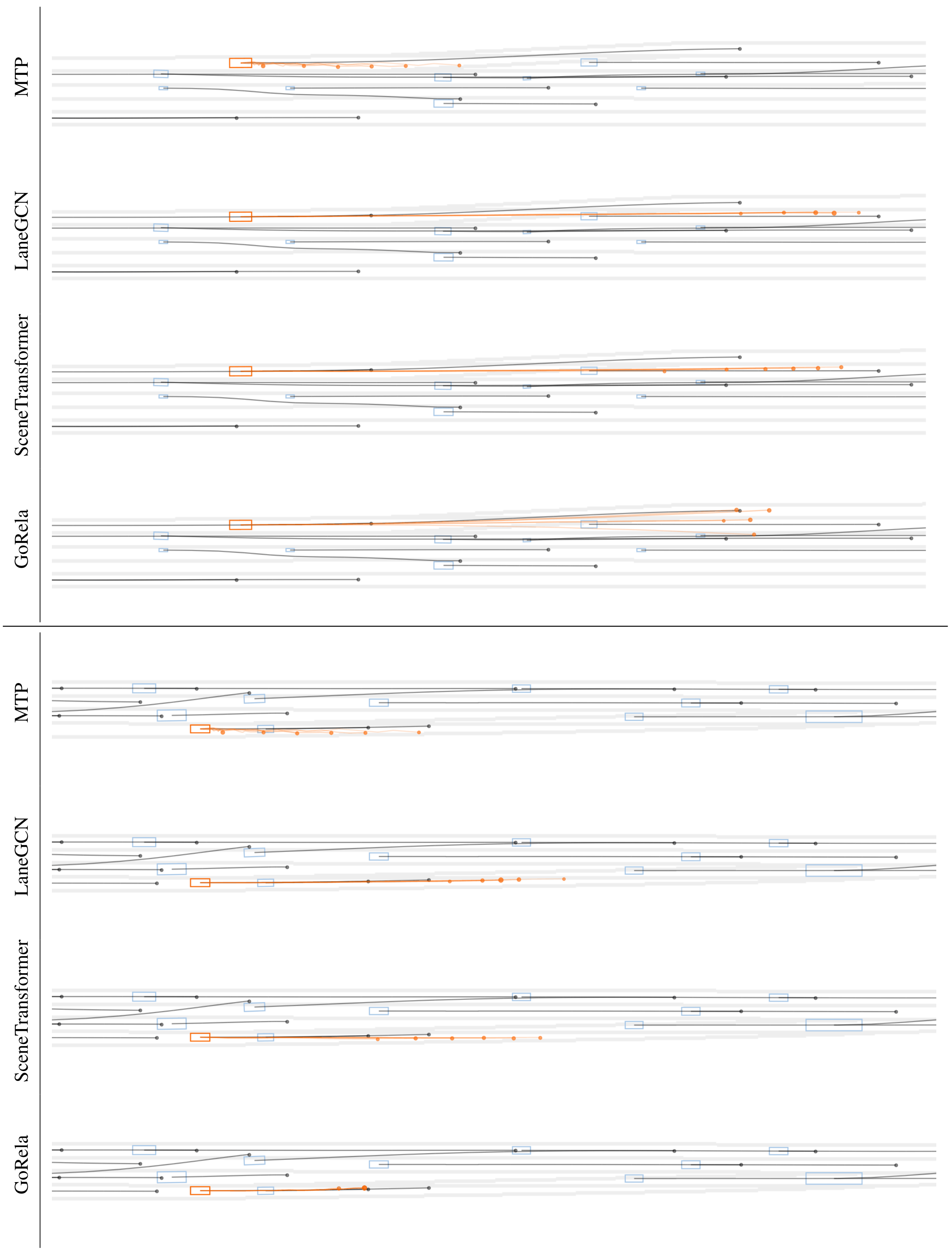


Fig. 7: **Qualitative comparison in HighwaySim**. 6 seconds long multi-modal motion forecasts. We only showcase predictions for 1 randomly sampled agent in each scenario since otherwise the visualization is very cluttered due to highway high speeds. Ground-truth plotted in gray. Rows 1-4 correspond to one scenario, and rows 5-8 to another.

References

[1] Y. Chai, B. Sapp, M. Bansal, and D. Anguelov, “Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction,” *arXiv preprint arXiv:1910.05449*, 2019. 1

[2] M. Liang, B. Yang, R. Hu, Y. Chen, R. Liao, S. Feng, and R. Urtasun, “Learning lane graph representations for motion forecasting,” in *ECCV*, 2020. 1

[3] H. Cui, V. Radosavljevic, F.-C. Chou, T.-H. Lin, T. Nguyen, T.-K. Huang, J. Schneider, and N. Djuric, “Multimodal trajectory predictions for autonomous driving using deep convolutional networks,” *arXiv preprint arXiv:1809.10732*, 2018. 1

[4] J. Ngiam, V. Vasudevan, B. Caine, Z. Zhang, H.-T. L. Chiang, J. Ling, R. Roelofs, A. Bewley, C. Liu, A. Venugopal, *et al.*, “Scene transformer: A unified architecture for predicting future trajectories of multiple agents,” in *International Conference on Learning Representations*, 2021. 1

[5] X. Mo, Z. Huang, Y. Xing, and C. Lv, “Multi-agent trajectory prediction with heterogeneous edge-enhanced graph attention network,” *IEEE Transactions on Intelligent Transportation Systems*, 2022. 1, 2

[6] S. Brody, U. Alon, and E. Yahav, “How attentive are graph attention networks?” *arXiv preprint arXiv:2105.14491*, 2021. 1, 2

[7] M. Fey and J. E. Lenssen, “Fast graph representation learning with PyTorch Geometric,” in *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019. 1

[8] S. Casas, C. Gulino, S. Suo, and R. Urtasun, “The importance of prior knowledge in precise multimodal prediction,” *arXiv preprint arXiv:2006.02636*, 2020. 2