Convolutions for Spatial Interaction Modeling
- Appendix -

In the appendix, we provide complete details on the model and experiment implementation to facilitate reproducing the proposed approaches and models: the network design of the first stage backbone (Section A), the ICNN network designs (Section B), the GNN network design (Section C), the training setup (Section D), the data set choices (E), and the metric design and metric variances (Section F).

We also provide additional quantitative and qualitative results: the experimental results focused on forecasted actor trajectories intersecting with non-vehicle traffic objects (Section G) and videos (Section H).

A. The extractor backbone network

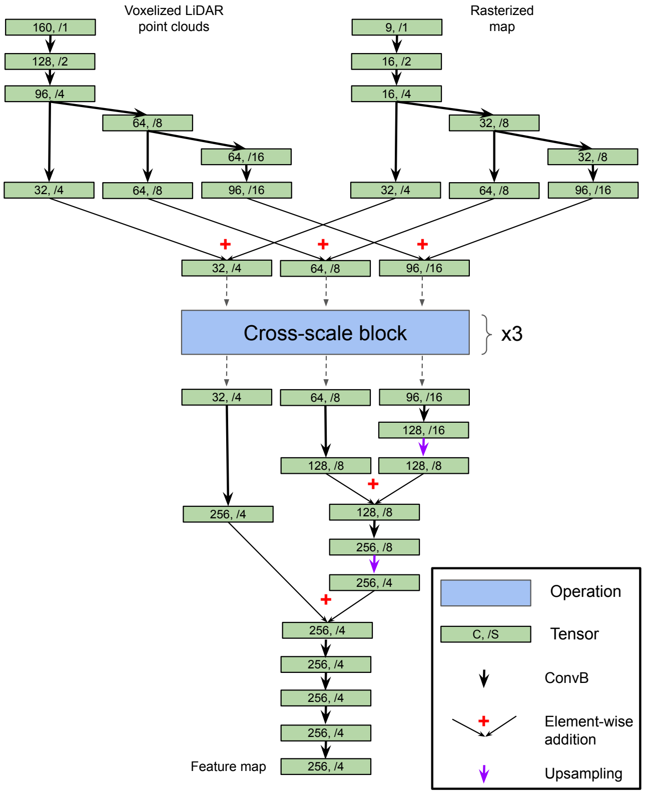

Figure S1. Multi-scale network design of the feature extractor. As illustrated in Fig. 1, the inputs are voxelized LiDAR point clouds and rasterized map, while the network output is a feature map of size $(256, W/4, L/4)$, where W and L are the grid width and length of the input BEV representation, respectively. The green boxes labeled as $C, /S$ represent tensors where C and S represent the number of channels and down-sampled scale relative to the input size, respectively. The operations connecting two tensors are **ConvB**s, except for the specified up-sampling, element-wise addition, and the cross-scale block. The cross-scale block, detailed in Fig. S2, is repeated 3 times.

In Fig. S1 we provide full and detailed design of the CNN feature extractor used in all of the models studied (see the high-level overview in Fig. 1). We note that the multi-scale design (as indicated by $/1$, $/2$, $/4$, $/8$, and $/16$, where the numbers represent the down-sampling scales relative to the input size) and cross-scale blocks (see Fig. S2) already encourage a large receptive field. Nevertheless, the experimental results presented in the main paper show that such single-stage CNN architecture still models the spatial interaction less effectively. By adding either the shallow ICNN or the GNN module in the second stage the interaction modeling performance is significantly improved.

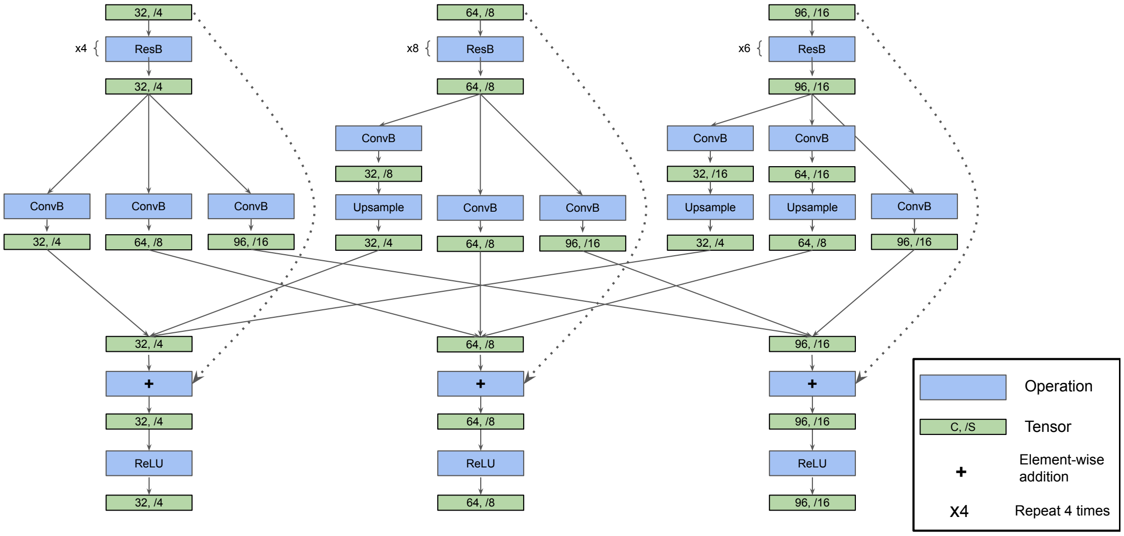

Figure S2. The cross-scale block

B. The ICNN network

For the IRs equal to 80m, 60m, 40m, 20m, and 5m, we set the grid sizes of the feature map crops to 64, 48, 32, 16 and 4, respectively. Zero-valued padding is utilized in the convolutional layers when necessary.

We did not extensively investigate network designs for the ICNN module. Several straightforward options (see Fig. S3) that stacked **ConvB** and **ResB** blocks in series were evaluated empirically. These options set the strides of the last few **ConvB** blocks to 2 so that the input feature map crop was down-sampled gradually to 1×1 after being processed by the ICNN. We observed that the model performance was not sensitive to the changes in these ICNN variants.

C. The GNN network

Two-layer MLPs are used in a few places in the GNN networks of this work. All output vectors through the MLPs have the identical dimension as fv, 256. In other words, the dimensions remain unchanged through the MLPs.

Note that both max-pooling and mean-pooling were studied in the GNNs (see Eq. 6) and no experimental difference was observed. We also explored adding other relative relations such as relative velocities to the graph edge attribute and observed insignificant changes to the model performances. Besides, in the main text all node and edge attributes were based on deterministic model outputs. We studied using probabilistic (Gaussian and Laplace) outputs for the attributes, and only measured performance difference within the metric variance level. Note that the GNNs experimented in this work built edges between all vehicle actors. It was clear that some of the edges were unnecessary (e.g, between two far-apart vehicles that were driving farther apart). However, given the high vehicle speed and long prediction horizon of 4s, such optimization was not straightforward and thus not experimented with in more depth.

D. The training setup

Each training sequential example comprises 10 past and current sweeps (-0.9s, -0.8s, $\dots$, 0s), and 41 current and future timestamps for ground-truth supervision (0s, 0.1s, $\dots$, 4.0s). The frame at current timestamp is referred to as the key frame. Each scene on the in-house data set is 25s long, producing at most 200 complete sequential examples. We trained all of the models with decimated key frames in the training split once (i.e., every sequential example whose key frame is at t, $t + 0.2$s, $t + 0.4$s, $\dots$, is used once during model training).

The models were implemented in PyTorch [5] and trained end-to-end with 16 GPUs (Nvidia RTX 2080), with a batch size of 2 per-GPU. Training without the GNN module is completed in about 12 hours. We use the Adam optimizer [4] with a learning rate of 2e-4, decayed to 2e-5 at 75% and 2e-6 at 95% of the training iterations.

For the models with the interaction loss, the weight of the interaction loss was set to 2.0, which slightly outperformed those with weights of 1.0 and 3.0 in the interaction metrics. No significant displacement error difference was observed.

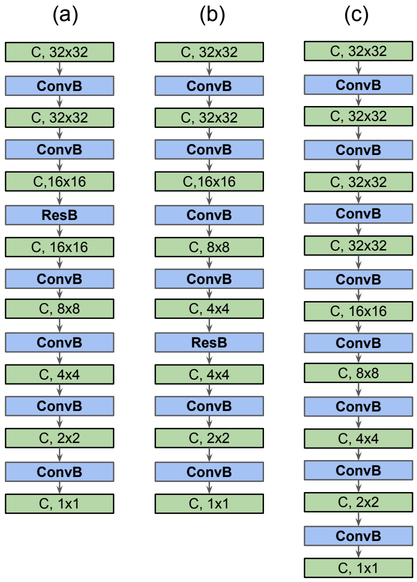

Figure S3. Various considered designs for ICNN, using 32×32 input as an example. The green boxes $(C, W \times W)$ represent tensors of grid size equal to $W \times W$ and C channels (C is equal to 256 in all designs). The presented main results were based on design (a).

E. The data

In the Methodology Section, we briefly explained that large data was required to conduct experiments with low metric variances and derive general conclusion. Here we provide more details of the in-house data set and its comparison to some open-sourced data sets.

As shown by the experimental results, the overlap rates are low, particularly for models equipped with high abilities in modeling interaction. This means large and diverse test data is required to achieve low metric variances. Take the popular nuScenes data set [1] for autonomous driving as an example, its training, validation, and test sets for prediction task combined have 1000 scenes (each is 20s long). On the in-house data, our work uses a test set of 5000 scenes (each is 25s long), which can help facilitate the comparison and analysis of fine differences (e.g., those between **+GNN** and **+GNN (no edges)** at large IRs, see metric variances in the next section).

In pure trajectory prediction tasks where the metric variance of displacement error is low even based on smaller data sets, we observed that the methodological comparison concluded on this in-house data set was generalized to other AV data sets. Specifically, in our earlier trajectory prediction works [3,7] we found the results on this in-house data set to correlate with the smaller open-sourced data sets. Comparing the trajectory prediction performance on the nuScenes data, it was also observed that the 3s displacement errors (vehicles) were 1.58, 1.45, and 1.04m for CAR-Net [2,6], SpaGNN [2], and our conv-only model (+ICM (IR=60m)), respectively. For fair comparison, all models were measured at a same detection recall of 60%.

Finally, we confirm that the observation that the convolutional approach performed comparably to or better than GNN remained valid with smaller data subsets. E.g., trained with 1/3 of the full training set (training time is also 1/3 long) and evaluated on the same large test set (5000 scenes), we measured displacement errors of 0.818, 0.641, and 0.714m in the baseline, +ICM (IR=80m), and +GNN (IR=0m) models, respectively. Their actor-actor overlap rates were 3.26, 1.04, and 1.70%; actor-static overlap rates were 1.22, 0.49, and 0.66%, respectively. These results suggest that the empirical studies presented in this work are generally valid.

We complete the discussion on data with two final notes. The first note is about the labeling standard for object detection. Actors that were far away from the traffic regions (e.g., roads and side-walks) were excluded during training and performance evaluation, which could lead to a higher detection performance than those based on data sets that considered all actors in the scene. The second note is about the distribution of the multimodal predictions, where approximately 80% of the moving vehicles were in the "straight"-driving mode (as opposed to left and right turns).

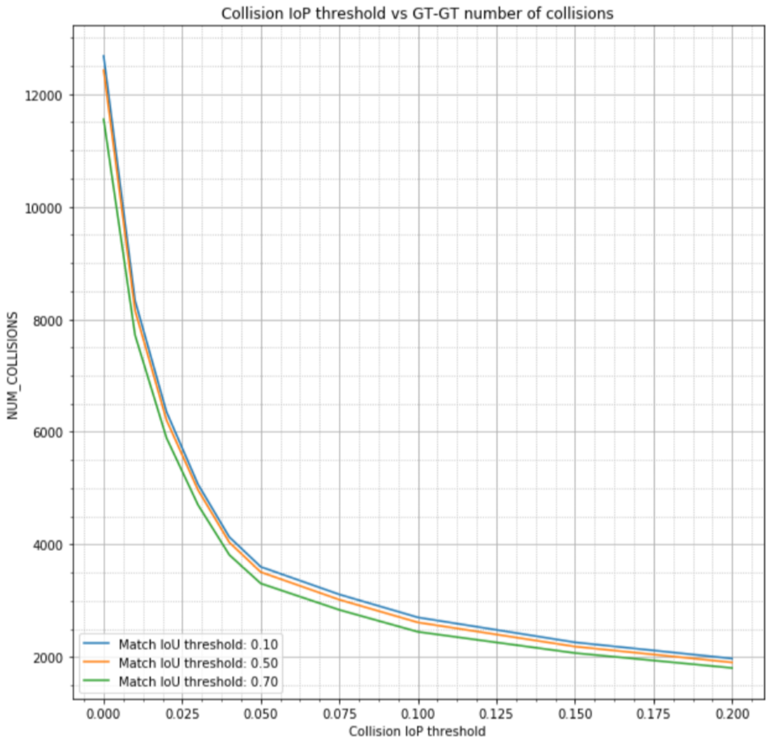

Figure S4. Number of ground-truth label overlaps of all timestamps vs. IoP through the test set. Here we only consider label trajectories whose boxes at the key-frame match the detections of the baseline model. We threshold the matching in terms of IoU at 0.1, 0.5, and 0.75 on the plot. The number of overlaps is non-zero even at high IoP, because large overlaps indeed occur in this data set, for instance, when the arm of a construction vehicle is over another vehicle, which would be a large overlap in the bird's eye view.

F. The metrics

The overlap interaction metrics are defined in terms of the ratio of intersection between two objects to the the area of the smaller object (IoP) rather than the more common IoU, because the latter is insensitive to an overlap between a large object and a small object. Although there are no collisions between actors in the data set, we have measured overlap rates between actor labels because some labeling boxes are slightly and imprecisely over-sized (thus the learnt object detections would be over-sized too). In Figs. S4 we plot the number of overlaps between label actors as a function of IoP threshold. The number drops rapidly after IoP at 0.05 approximately, meaning thresholding IoP at 0.05 would just eliminate the majority of the false positive overlaps. We adapted this threshold value in evaluating the overlaps of predicted trajectories. Such setting allowed the metrics to measure interaction modeling in the experiments robustly and indicatively.

The proposed convolutional approach can be applied to improve interaction modeling for other traffic participants such as pedestrians and cyclists. We observed significant effects in case studies. The studies were not discussed in this work because even labels of pedestrians and cyclists often had considerable and arguably correct overlaps. As a result, the overlap rates were no longer unambiguous metrics for interaction modeling. We thus limited our discussion to interaction between vehicles and vehicles, and between vehicles and static obstacles.

By including these careful designs for the data and metrics, the resulting metrics variances were 0.004m, 2e-4, and 3e-4 for DE, actor-actor, and actor-static collision rates, respectively, computed by training **+ICM** (80m) 4 times (other models were trained and evaluated once). This level of variances allowed us to resolve difference in interaction modeling between models, even on rare events such as overlaps.

G. Additional results focusing on overlaps with non-vehicle actors

The actor-static overlap rate in the main paper considers overlaps between forecasted trajectories with both vehicle and static non-vehicle traffic objects. In this section, we provide additional results focusing on overlap with static *non-vehicle* traffic objects. Here the overlap rate is defined as the percentage of forecasted trajectories of detected actors that overlap with

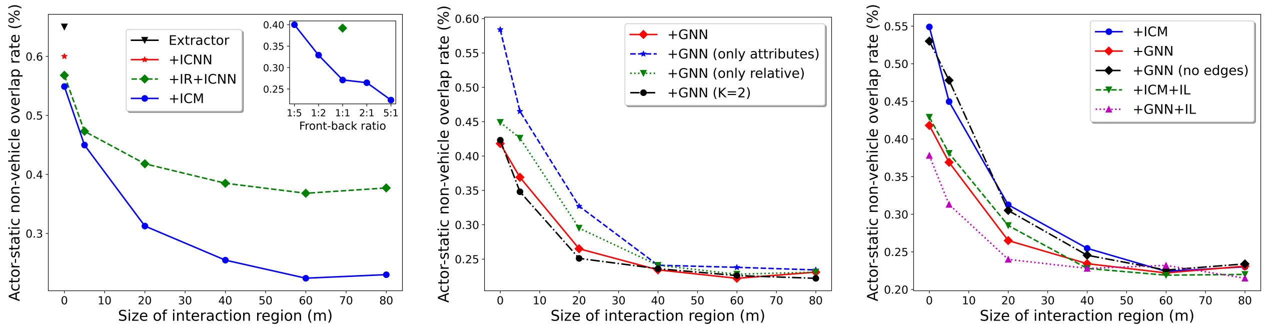

Figure S5. Overlap rate of forecasted actor trajectories overlapping with static *non-vehicle* traffic objects.

ground-truth static *non-vehicle* traffic objects. The three panels in Fig. S5 correspond to Figs. 3 - 5 in the main paper. As the feature map input cropped by the IR covers features of both vehicle and non-vehicle traffic objects in the ICM approach, it is not surprising that ICM effectively improves this interaction metric too. It is, however, interesting to note that even though GNN does not build nodes for the non-vehicle traffic objects in the graph, it also lowers this overlap rate by 24%, by comparing **+ICM** (0m) to **+GNN** (0m). The reduction is attributed to the fact that by avoiding overlaps with vehicles (after adding the GNN), the overlaps with some of the non-vehicle objects near those vehicles are also avoided. Another factor may be the proximity effect of CNNs, as the pixel features of the vehicle actors might comprise information about its nearby non-vehicle objects. The improvement of GNN on the overlap avoidance with non-vehicle objects (24%), however, is considerably lower than that with vehicle actors (42% as shown in the main paper by comparing **+ICM** (0m) to **+GNN** (0m) in Fig. 5 right), which is reasonable as the GNN does not model the interactions with non-vehicle objects directly.

H. Videos

In addition to Fig. 6, we provide qualitative results with three videos where the predictions of the baseline (**+ICM**, 0m) are on the left and the predictions of the ICM model (**+ICM**, 60m) are on the right. The overlapped obstacles are filled in red, the ground-truth trajectories are in grey, and the forecasted trajectories are in blue. Trajectory visualization is downsampled to 2Hz for clarity. Different from Fig. 6, we visualize the predictions of all actors in the common AV frame of reference. Note that the videos are 20s long, because each scene in the data set is 25s long, where the first second is used for the 10-sweep input and the last four seconds are used for the 4s forecasting time horizon.

References

[1] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11621–11631, 2020. 3

[2] Sergio Casas, Cole Gulino, Renjie Liao, and Raquel Urtasun. Spatially-aware graph neural networks for relational behavior forecasting from sensor data. *arXiv preprint arXiv:1910.08233*, 2019. 3

[3] Nemanja Djuric, Henggang Cui, Zhaoen Su, Shangxuan Wu, Huahua Wang, Fang-Chieh Chou, Luisa San Martin, Song Feng, Rui Hu, Yang Xu, et al. Multinet: Multiclass multistage multimodal motion prediction. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, 2020. 3

[4] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 2

[5] Adam Paszke, Sam Gross, et al. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 2

[6] Amir Sadeghian, Ferdinand Legros, Maxime Voisin, Ricky Vesel, Alexandre Alahi, and Silvio Savarese. Car-net: Clairvoyant attentive recurrent network. *CoRR*, abs/1711.10061, 2017. 3

[7] Zhaoen Su, Chao Wang, Henggang Cui, Nemanja Djuric, Carlos Vallespi-Gonzalez, and David Bradley. Temporally-continuous probabilistic prediction using polynomial trajectory parameterization. In *IEEE International Conference on Robotics and Automation (IROS)*, 2021. 3