A. Motion Forecasting Metrics

Distance error metrics are the most commonly used to compare methods, capturing how close a predicted trajectory (discrete time sequence of states) matches a future object track, under Euclidean distance. The most common is Average Displacement Error (ADE) [1, 24]. Because the future is inherently stochastic and multi-modal, most models output a (weighted) set of trajectory hypotheses, and then a minimal error over the set (of constrained size) is reported (i.e. minADE [9]). For methods that provide explicit or implicit future probability distributions, the likelihood of the ground truth future trajectory can be used as a metric [8, 31, 27, 28]. Framing the problem instead as one of detection of future locations, Argoverse [9] employs Miss Rate within 2 meters as their primary metric, which has the benefit to being tolerant to outliers. A number of metrics including minADE have been extended for use with jointly predicted agent trajectories[6].

B. Dataset Splits

The dataset provides 6 different splits of the original set of 20 second scenarios. The scenarios are first split into training, validation and test sets. This is done by hashing a string containing the date of the data capture and the unique ID of the vehicle used to capture the data. The hashed values are split into mutually exclusive 70% training, 15% validation, and 15% testing subsets of the 20 second scenarios. From these 3 subsets we generate examples by extracting 9.1 second windows from the longer 20 second scenarios. Each 9.1 second window contains 91 time steps at 10Hz - 10 history samples, 1 sample at the current time, and 80 future steps. We extract 5 different sets of windowed examples from the respective 20 second splits, training, validation, testing, validation interactive, and testing interactive. The training set contains 9.1 second windows starting at times $\{0, 2, 4, 5, 6, 8, 10\}$ seconds within the 20 second scenarios. The validation and testing sets contain 9.1 second windows starting at times $\{0, 5, 10\}$ seconds. The validation interactive and testing interactive sets contain 9 second windows starting at times $\{4, 5, 6\}$ seconds to focus on the interactive portion of the scenario. The 5 windowed sets are included in the published dataset along with the full 20 second training set. Each of the windowed sets contains a list of objects in the scene to be predicted. The training, validation, and testing sets contain up to 8 objects per scenario chosen to include at least 2 objects of each type if available. Selection is biased to include objects that do not follow a constant velocity model or straight paths. For the validation interactive and testing interactive sets, only the mined interactive agent pair objects are included in the list of objects to predict. In addition, each object to predict has a difficulty level based on how easily it is predicted by an LSTM extrapolation model.

C. Metrics Details

Overlap rate (OR) details. A binary indicator is assigned to each sample alerting of self-overlapping. The average over the dataset creates the overlap rate. We only consider the highest scoring joint prediction $\tilde{\mathbf{p}}$ here. Our metric counts an overlap with the following criteria: given the joint predicted trajectories of A agents, an overlap is counted if the rotated bounding box of any of the A agents overlaps with any other visible object at any time step within the prediction interval T. Note that agents not visible at prediction time (due to their later appearance) are not considered for potential overlaps. Consider $\mathcal{G}_t = \{\tilde{s}_{a,t} \forall a, g_{b,t} \forall b \in 1 \dots B\}$ where $\tilde{s}_{a,t}$ are waypoints from $\tilde{\mathbf{p}}$ at time t, and $g_{b,t}$ are groundtruth waypoints from B nearby environmental agents, the single overlap indicator is defined as:

$$\mu_{\text{OR}}(\mathbf{e}) = \sum_t \sum_a \sum_{s' \in \mathcal{G}_t \setminus \tilde{s}_{a,t}} \mathbb{1}[\text{IOU}(b(\tilde{s}_{a,t}), b(s'_t)) > 0] \tag{2}$$

where $b(.)$ is a function to derive a 5-dof (x, y, width, length and heading) bounding box from a waypoint. The groundtruth bounding box is used for an environmental agent. For a predicted waypoint $s_{a,t}$, we derive the heading from the derivative to the previous waypoint and use the groundtruth bounding box sizes. $IOU(\cdot)$ computes the intersection-over-union between two 5-dof boxes.

D. Overlap Metric

We use a marginal overlap-based metric with the simple baseline models to quantify the difficulty and interactivity in our dataset. We consider a trajectory for an agent to contain an overlap if at any time point, the agent bounding box overlaps with a ground-truth box at that time. The overlap rate is the number of agents whose trajectories have overlaps divided by the total number of predicted agents.

We compute the overlap rate for the constant velocity model and compare the performance between the regular split and interactive split of the dataset. For the constant velocity model, we found that 38.4% of predicted vehicles in the regular split, and 44.2% of predicted vehicles in the interactive split have trajectories that overlap with a groundtruth (Table 5). This shows that the interactive split is more challenging, and suggests that more interactions between agents in that split.

E. Conditional Model Details

The model we use for conditional behavior prediction is based on the baseline model we describe in 5.1. Figure 7 provides an overview diagram of the proposed model. We

Val. set	Model	Overlap Rate Vehicle	Pedestrian	Cyclist
Regular	Const. Vel.	38.4%	29.8%	22.3%
	LSTM	27.9%	22.9%	22.1%
Interactive	Const. Vel.	44.2%	30.6%	27.0%
	LSTM	36.3%	32.3%	25.6%

Table 5: **The interactive split of the data has more overlaps per scene.** Despite the interactive set only requiring predictions for two agents instead of up to eight agents for the regular dataset, the split contains more scenes where a constant velocity model or an LSTM model – neither of which models other agents – produces at least one overlap. Statistics are reported on the validation set for both dataset splits. The marginal-based overlap metric is used for both splits so that the rates can be compared across the splits. Constant velocity model only predicts a single trajectory per agent. For the LSTM model, the highest scoring trajectory for each agent is used.

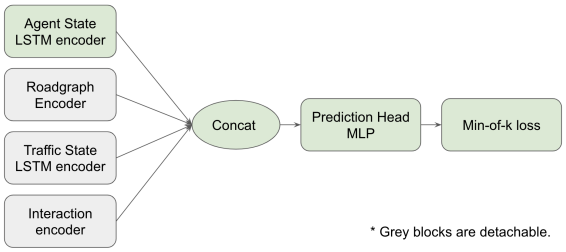

Figure 7: **Diagram of baseline architecture.** An illustration of the baseline architecture employed for the family of learned models with a base LSTM encoder for agent states. The three detachable components are a roadgraph polyline encoder [14], a traffic state LSTM encoder, and a high-order interactions encoder following [14]. The trajectories are predicted through a MLP with min-of-k loss.

use the LSTM encoder and all three enhancements (roadgraph encoding with polylines , traffic signal states encoded in an LSTM, modeling high-order interactions with a global interaction graph). To make this model suitable for conditional predictions, we add an early fusion conditional encoder similar to [36]. Just like [36], we train the model to do both conditional and unconditional prediction by passing in a randomly selected query agent's ground truth future trajectory as conditional query input in 95% of training samples while providing no conditional query in the other 5%. We generate 6 predictions per agent and evaluate the KL divergence over the full 8 second future trajectory.

F. Videos

The included videos show visualization of some samples of scenarios from the dataset including those in Figure 1a and Figure 1b.

Acknowledgements

We thank Paul Hempstead, David Margines, Dietmar Ebner, Peter Pawlowski, Balakrishnan Varadarajan, Avikalp Srivastava, Zhifeng Chen, and Rebecca Roelofs for their comments and suggestions. Additionally, we thank the larger Google Brain team and Waymo Research teams for their support.

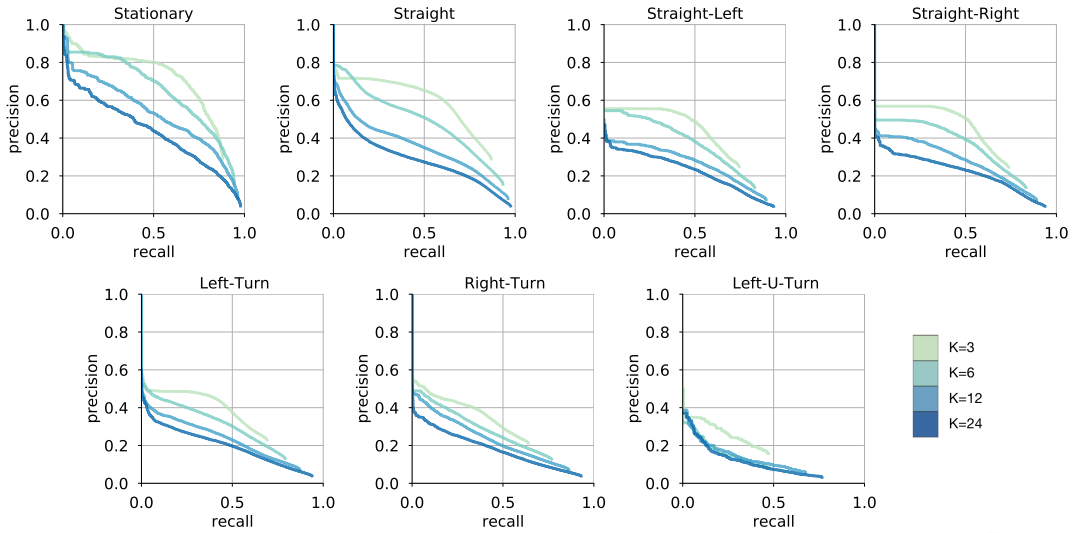

Figure 8: Precision versus recall curves for increasing number of predictions (K) for the polyline model at **3 seconds** for vehicles across trajectory shape buckets for the standard validation dataset. Recall increases with K but AUC decreases.

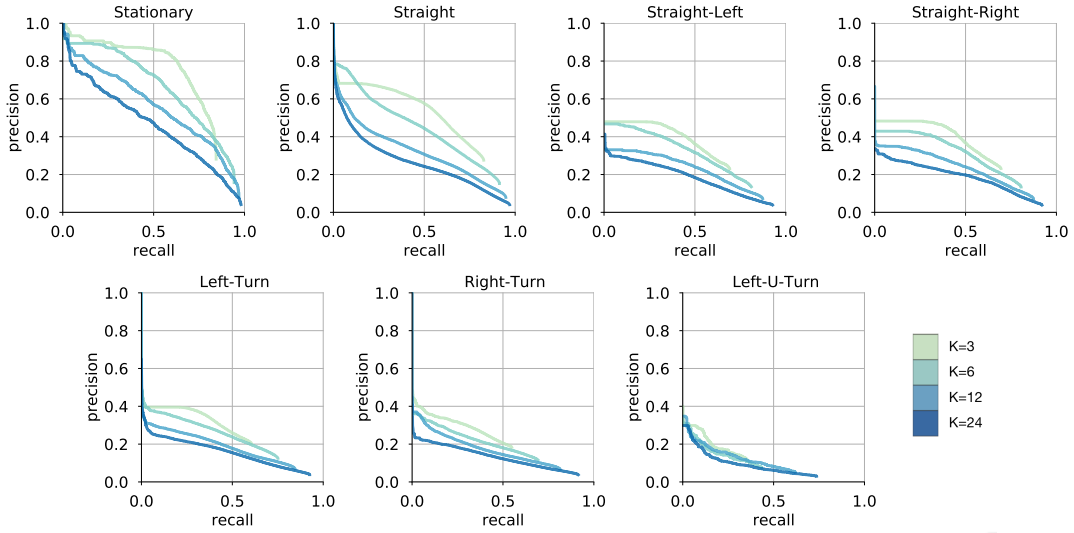

Figure 9: Precision versus recall curves for increasing number of predictions (K) for the polyline model at **5 seconds** for vehicles across trajectory shape buckets for the standard validation dataset.

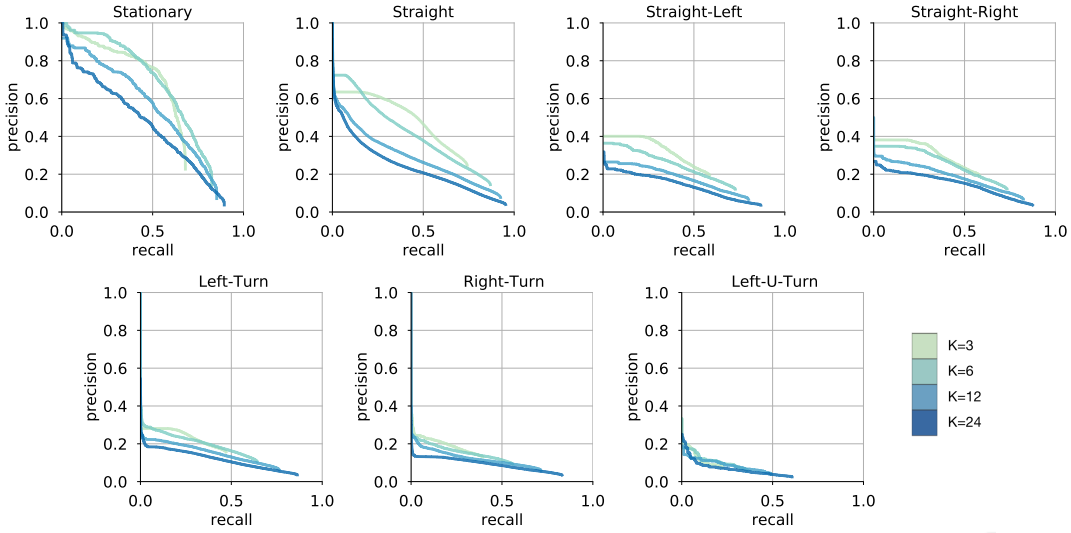

Figure 10: Precision versus recall curves for increasing number of predictions (K) for the polyline model at **8 seconds** for vehicles across trajectory shape buckets for the standard validation dataset.