Category	Set	Instances	Description
Intention Labels (Vehicle)	Stopped	130743	The vehicle is stopped. This can happen in many scenarios such as stopping for a traffic light, waiting to make a turn at an intersection, yielding for a pedestrian, etc.
	Parked	127150	The vehicle is parked along the street or parking lot
	Lane change to the left	2120	The vehicle is merging into the next lane.
	Lane change to the right	2087	
	Cut in to the left	347	The vehicle is cutting into another lane.
	Cut in to the right	736	
	Turn left	15190	The vehicle is turning (ex: at an intersection or towards a highway ramp).
	Turn right	13171	
	Moving / Other	206243	The vehicle is driving forward or some other movement that is not captured in the other labels.
Intention Labels (Pedestrian)	Stopped	32538	The pedestrian is stopped along the street
	Moving	241889	The pedestrian is walking (ex: along the street)
	Waiting to cross	49576	The pedestrian is waiting to cross the intersection.
	Crossing the road	64870	The pedestrian is crossing the road.
	Potential Destination	67862	The potential location where the pedestrian may walk to.
Environmental Labels	Lane information	440338	The possible actions a vehicle can take based on the current lane it is in. ($e.g.$ right turn, left Turn, go forward, u-turn, lane change not possible). Note that multiple choices can be selected depending on the situation. For example, a vehicle can be in a lane that goes forward or turns left. In our dataset, if a lane type is possible we select 1 and if it is not possible we select 0. Sometimes, if the vehicle is out of frame and lane information cannot be deduced, we label it as -1.
	Traffic light	42476	The current state of the traffic light ($e.g.$ Red straight, Green round, Yellow round, etc.)
	Traffic sign	39066	The type of the traffic sign ($e.g.$ Stop, Left turn only, Do not enter for all)
	Road Exit and Entrance	126889	The positions of the road entrances/exits for a given scene. There can be a variable number of road entrances/exits depending on map topology. Refer to figure 9 for more details.
Contextual Labels	Age	166874	The estimated age category (child, adult, senior) of the pedestrian.
	Gender	166874	The gender of the pedestrian (male/female)
	Weather	644	The weather condition of the scenario (Sunny/Dusk/Cloudy/Night).
	Road condition	644	The road surface condition (dry / wet).

Table 4: Details of the LOKI dataset. We report the various types of labels, number of instances of each label, and descriptions for all label types.

7. Details of the LOKI Dataset

The LOKI dataset is collected from central Tokyo, Japan using an instrumented Honda SHUTTLE DBA-GK9 vehicle. Driving scenarios are collected from both suburban and urban areas at different times of the day. The camera, LiDAR, GPS and vehicle CAN BUS information were recorded. The RGB camera and four LiDAR sensors are placed on top of the vehicle to obtain better environment coverage. In addition, the timestamps were recorded for post multi-sensor synchronization processing. The cameras and LiDARs were placed on top of the vehicle to obtain better environment coverage. The sensors used for recording this dataset are listed below:

- A color SEKONIX SF332X-10X video camera (30HZ frame rate, 1928×1280 resolution and $60°$ field-of-view (FOV)).
- Four Velodyne VLP-32C 3D LiDARs (10 HZ spin-rate, 32 laser beams, range: 200m, vertical FOV $40°$).
- A MTi-G-710-GNSS/INS-2A8G4 with output gyros, accelerometers and GPS.

We used the CAN BUS to compensate for the ego motion while merging the LiDAR data and then transformed it to the virtual position (the center of the vehicle). The calibration is obtained through the extrinsic (the transformation between virtual LiDAR point and camera) and intrinsic camera parameters.

The recorded agents fall into 8 main classes. The vehicle classes are truck, van, car, bicyclist, motorcyclist, and bus. The pedestrian classes are pedestrian and wheelchair. As described in the main manuscript, we have three types of labels in the LOKI dataset: Intention labels, Environmental labels and Contextual labels. Intention labels for the vehicle classes include diverse motion state such as stop, lane change, cut in, etc., which can be observed in suburban and urban driving scenarios. Similarly, we annotated intention labels for the pedestrian classes such as moving, waiting to cross, etc. We additionally annotate a potential destination of stopped / waiting agents under the pedestrian classes, which is a direct indicative of their intention. Note that the potential destination cannot be obtained from the future location of agents as they mostly stay still until the end of the video clip. Details of the labels and their description are shown in Table 4. To further explore how environments and scene context can affect the future behavior of agents, we annotate environmental labels (lane information, traf-

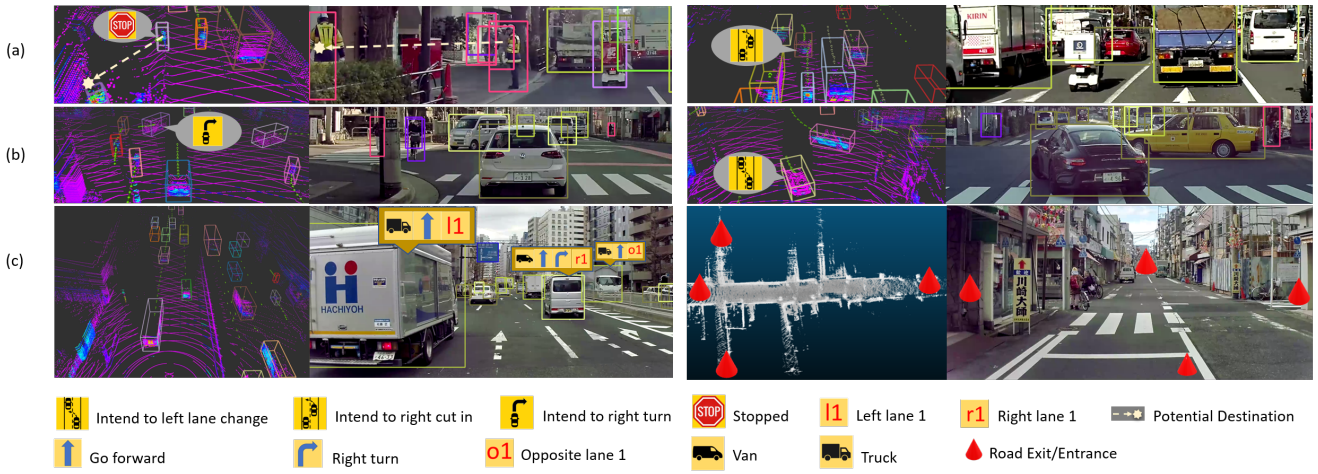

Figure 9: Visualization of three types of labels: (a-b) Intention labels; and (c) Environmental labels. The left part of each image is from laser scan and the right part is from RGB camera. In (a), the current status of pedestrian is "Stopped", and the potential destination shows where the pedestrian may go in the future. In (c) left, the blue arrow indicates the possible action of the vehicle based on the current lane it is on. The red words show the lane position related to the ego-vehicle.

fic light / sign, road entrance / exit) as well as contextual labels (age, gender, weather, road condition). Figure 9 illustrates different types of labels that we annotated in the LOKI dataset.

8. Model Implementation

In this section we provide more information on our data pre-processing, model choices and architecture details of each module.

8.1. Data Pre-processing

The LOKI dataset contains diverse traffic scenarios of up to 20 seconds, with the average recorded scene length of 12.6 seconds. With access to longer recordings, our dataset can be used for a multitude of trajectory prediction settings, from very short-term observations and predictions (3 seconds) to much longer observations and prediction horizons (10+ seconds). In our work we consider a long-term prediction setting with a $3s$ observation horizon and a $5s$ prediction horizon. Thus, we filter all agents that are not observed for at least $8s$ in a given traffic scenario. We use a sliding-window of $0.2s$ to augment tracklets. Furthermore, as in other works [8, 37], we further augment our dataset during training with "rare" examples such as turning and lane changes.

We solve the problem of intention prediction and trajectory prediction jointly as described in the main manuscript. The types of intentions for pedestrian agents and vehicle agents are different due to their different traffic restrictions and trajectory behaviors. For vehicles, we use the following set of discrete actions: *Other/moving*, *Stopped*, *Parked*, *Lane change*, *Turn left*, and *Turn right*. We group *Lane change to the left* and *Lane change to the right* into a single intention type, as the number of instances that contain lane changes are much smaller and we noticed that separating the two did not improve performance. Furthermore, we do not compute a loss or predict on trajectories that contain *Cut in to the left* and *Cut in to the right* , as we noticed that these constitute less than 0.01% of the dataset, making it hard for the model to meaningfully distinguish from turning and lane-changing behavior. For pedestrians, we use the following set of intentions: *Moving*, *Waiting to cross*, *Crossing the road*, and *Stopped*.

Our dataset originally contains frame-wise action labels for each agent. In order to use them as intention labels, we define intention to be a future action [10]. Thus, the intention of an agent at frame m is the agent's action at frame $m + q$ where we fix $q = 4$ frames ($0.8s$) for our work. Note that for the observation period, we do not use intentions and only input observed actions to the model to prevent ground-truth leakage; the intention labels are only used for future trajectory prediction.

8.2. Observation Encoder

The observation encoder outputs a representation of past motion history, observed actions, and lane information for each actor independently. In our paper, we refer to past actions and lane information as observed states.

8.3. Long-term Goal Proposal Network

For each actor, we first independently predict a proposed long-term goal position [9, 5]. The proposed destination is

	Layer	Input shape	Output shape
0	encoder_past.GRUCell.enc	[1, 15, 21]	[1, 64]

Table 5: We use a GRU to encode the observation information for each actor. We use a hidden dimension of 64. The input is 15 observation frames with 21 inputs at each frame (2 from position, 8 from vehicle actions, 5 from pedestrian actions, and 6 from lane information). We use one-hot-encoding to represent action types. We also include a "None" class for both vehicle and pedestrian actions. This allows vehicle agents to choose "None" for pedestrian action types and pedestrian agents to choose "None" for vehicle action types.

similar as in other works [9] and is simply the endpoint of the trajectory, which in our case is $5s$ in the future. Because there are many plausible futures, we capture multi-modality through learning a long-term goal distribution for each actor. To predict multiple trajectories, we sample various goals and condition our Scene Graph + Prediction decoder module on each sampled goal. We follow a similar formulation as proposed in [9] and use a Conditional Variational Autoencoder (CVAE) to learn a latent distribution of the goals.

	Layer	Input shape	Output shape
0	encoder_destination.Linear_1	[1, 2]	[1, 8]
1	encoder_destination.ReLU_1	-	-
2	encoder_destination.Linear_2	[1, 8]	[1, 16]
3	encoder_destination.ReLU_2	-	-
4	encoder_destination.Linear_3	[1, 16]	[1, 16]
5	encoder_latent.Linear_1	[1, 80]	[1, 8]
6	encoder_latent.ReLU_1	-	-
7	encoder_latent.Linear_2	[1, 8]	[1, 50]
8	encoder_latent.ReLU_2	-	-
9	encoder_latent.Linear_3	[1, 50]	[1, 32]
10	decoder_latent.Linear_1	[1, 80]	[1, 1024]
11	decoder_latent.ReLU_1	-	-
12	decoder_latent.Linear_2	[1, 1024]	[1, 512]
13	decoder_latent.ReLU_2	-	-
14	decoder_latent.Linear_3	[1, 512]	[1, 1024]
15	decoder_latent.ReLU_3	-	-
16	decoder_latent.Linear_4	[1, 1024]	[1, 2]

Table 6: Sub-network architectures used for the goal-proposal network, modeled closely from model [9]. Batch size of 1 used for example.

8.4. Scene Graph + Prediction Module

The Scene Graph and prediction module performs joint intention and trajectory prediction while reasoning about various factors that may affect agent intent including i) agent's own will ii) agent-agent interaction and iii) agent-environment influence.

We construct a traffic scene graph to capture interaction and environmental influence. We have two types of nodes: 1) agent nodes 2) road entrance/exit nodes. The agent nodes are for dynamic agents in a scene (vehicles and pedestrians). The road entrance/exit nodes are static nodes that are positional markers that indicate where a road entrance or exit lies. These static nodes are used to provide information regarding map topology. In this work, we assume that these road markers are accessible to the model based on the annotations in our dataset. As described in our main manuscript, we use directional edges to propagate information through the various scene agents. Agents are connected to each other with bidirectional edges if they are within a certain threshold of 20 meters away from each other. Similarly, we connect a directed edge from static nodes to dynamic nodes if the agent is within 35 meters of the road entrance/exit. This graph is flexible in that a variable number of nodes or node types can be added as modification to this graph.

The Scene Graph + Prediction Module is then used to recurrently propagate information, predict intention, and predict trajectory. At each timestep, we first compute edge attributes between each pair of nodes. We use the edge_attr network to embed nodes' velocities and relative positions between each pair of nodes. We then use the transformer_conv layer (with a single attention head) [45] for message passing and update each node's hidden states based on its neighbors. Following this, we use the vehicle_intention_predictor and pedestrian_intention_predictor networks to predict agent intention at that current timestep. The trajectory_predictor is then conditioned on the hidden state of rnn_future_GRUCell.dec and the current intention prediction to predict the next position of each agent. Finally, the predicted positions are then inputted into rnn_future_GRUCell.dec to update the hidden states of each actor. This entire process is repeated for the prediction horizon length to unroll full trajectories while accounting for interactions and environmental information.

	Layer	Input shape	Output shape
0	trajectory_predictor.Linear_1	[1, 93]	[1, 80]
1	trajectory_predictor.ReLU_1	-	-
2	trajectory_predictor.Linear_2	[1, 80]	[1, 40]
3	trajectory_predictor.ReLU_2	-	-
4	trajectory_predictor.Linear_3	[1, 40]	[1, 2]
5	vehicle_intention_predictor.Linear_1	[1, 80]	[1, 256]
6	vehicle_intention_predictor.ReLU_1	-	-
7	vehicle_intention_predictor.Linear_2	[1, 256]	[1, 128]
8	vehicle_intention_predictor.ReLU_2	-	-
9	vehicle_intention_predictor.Linear_3	[1, 128]	[1, 8]
10	pedestrian_intention_predictor.Linear_1	[1, 80]	[1, 256]
11	pedestrian_intention_predictor.ReLU_1	-	-
12	pedestrian_intention_predictor.Linear_2	[1, 256]	[1, 128]
13	pedestrian_intention_predictor.ReLU_2	-	-
14	pedestrian_intention_predictor.Linear_3	[1, 128]	[1, 5]
15	rnn_future.GRUCell.dec	[1, 1, 80]	[1, 1, 80]
16	edge_attr.Linear_1	[1, 8]	[1, 16]
17	edge_attr.ReLU_1	-	-
18	edge_attr.Linear_2	[1, 16]	[1, 16]
19	transformer_conv_1	[1, 80]	[1, 80]

Table 7: Sub-network architectures used for the Scene Graph + Prediction module. Batch size of 1 used for example.

8.5. Training

8.5.1 Loss Functions

For convenience, we copy the loss functions used for training from our manuscript:

$$\mathcal{L}_{GPN} = \alpha_1 D_{KL}(\mathcal{N}(\mu, \sigma) \| \mathcal{N}(0, I)) + \alpha_2 \|\hat{G} - G\|_2^2$$

$$\mathcal{L}_{int} = -\sum_{i=0}^{n} w_i * y_i * log(\hat{y_i})$$

$$\mathcal{L}_{traj} = ||V - \hat{V}||_2$$

$$\mathcal{L}_{Final} = \lambda_1 \mathcal{L}_{GPN} + \lambda_2 \mathcal{L}_{int} + \lambda_3 \mathcal{L}_{traj}$$

We set $\lambda_1 = 1$, $\lambda_2 = 100$, $\lambda_3 = 200$, $\alpha_1 = 1$, $\alpha_2 = 1$

8.5.2 Training details

We train the entire network end-to-end with the $\mathcal{L}_{Final}$ loss using a batch size of 32 scenarios and learning rate of 1×10^{-4} using the ADAM optimizer. The intention prediction and trajectory forecasting tasks are heavily related with one another; thus we observed that training end-to-end helped with performance compared to modular training. Note that our batches are also grouped with an appropriate adjacency list to denote neighbors (connected edges) in a given batch.

During training, we train with the ground-truth destination as the long-term goal, as we noticed that because short-term intentions are influenced by long-term goals, it is important for the intention prediction networks to get a clean signal while training. During testing, we condition on a sampled goal from the Goal-proposal Network. We also adopt the truncation trick as in [9] to appropriately sample based on a varying number of future trajectories. The latent variable is sampled from different distributions depending on the number of future trajectories to be predicted: for $N = 1$ (single-shot) we sample the from $\mathcal{N}(0, 0)$ while for $N = 20$ (multimodal) we sample from $\mathcal{N}(0, 1.1)$.

9. Visualizations

In this section, we provide multiple visualizations that illustrate our proposed model's top-1 predictions (Figure 10), top-5 multimodal predictions (Figure 11), and our model's predictions with and without intention conditioning (Figure 12). Please view the video files provided in the supplementary folder for more detailed visualizations.

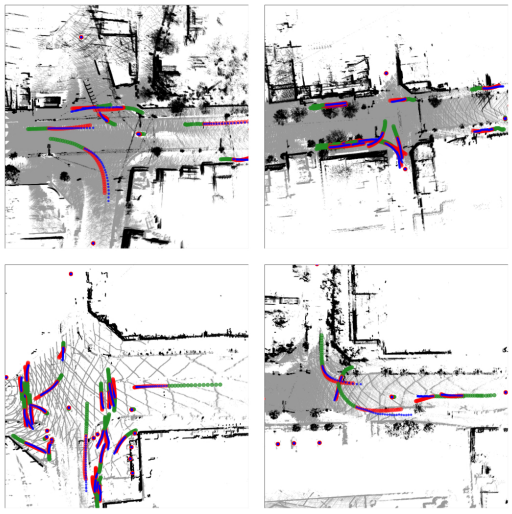

Figure 10: Visualization of our model's (Ours+IC+SG) top-1 (out of N=20 multimodal setting) predictions. Agent's past trajectory is represented in green. Agent's ground truth future is blue. Agent's predicted trajectories are in red (with increasing opacity to indicate better matches to the ground truth). We observe that our model performs reasonably in complex traffic scenarios.

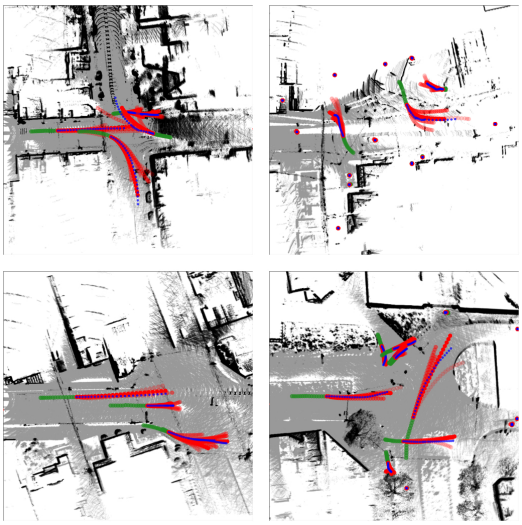

Figure 11: Visualization of our model's (Ours+IC+SG) top-5 (out of N=20 multimodal setting) predictions. Agent's past trajectory is represented in green. Agent's ground truth future is blue. Agent's predicted trajectories are in red (with increasing opacity to indicate better matches to the ground truth).

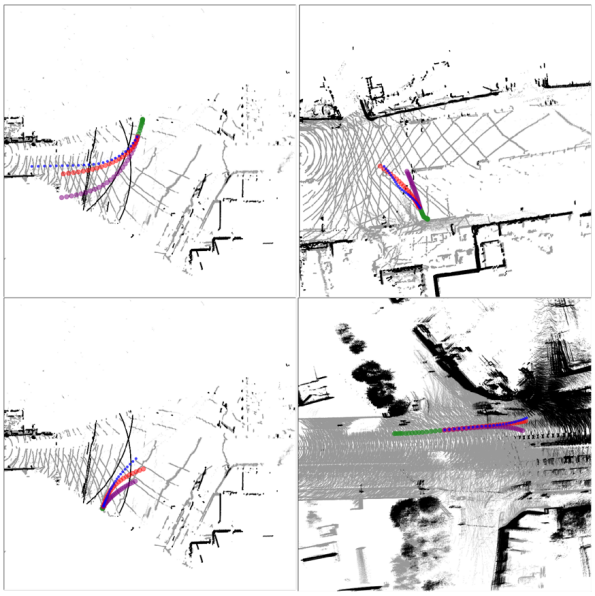

Figure 12: Comparison of with and without intention priors/scene graph for trajectory prediction. Agent's past trajectory is represented in green. Agent's ground truth future is blue. The top-1 predictions by the model without intention conditioning and scene graph are in purple. The top-1 predictions by the model with intention conditioning and scene graph are in red. We can qualitatively observe the efficacy of intention conditioning and incorporating interaction and environmental cues.