A. Preprocessing and Dynamics

A.1. Pre-encoding of Nodes and Edges

Since all datasets we use provide global coordinates and headings of the nodes, we first transform the global coordinates of the state history and state future to local frames fixed at the nodes' current position. For vehicles with heading angle ψ, we use $\cos(\psi)$ and $\sin(\psi)$ as features instead of ψ to prevent the $\pm 2\pi$ issue. For every type of nodes, the raw features are passed through a fully connected layer as pre-encoding, and the network is shared in all modules that require state information, i.e., whenever we use state information, the state vector passes through the pre-encoding layer first.

For edges, we construct a pre-encoding layer for every edge type (e.g., vehicle-vehicle, vehicle-pedestrian, pedestrian-vehicle, and pedestrian-pedestrian). The pre-encoding layer extracts raw features from the states of the two agents then passes them through a fully connected layer. The raw features contain agents' relative position and relative velocity in the local frame as well as their sizes.

A.2. Dynamic Models and Collision Checking

We use a Dubin's car model for vehicles in the scene with

$$s = \begin{bmatrix} X \\ Y \\ v \\ \psi \end{bmatrix}, a = \begin{bmatrix} \dot{v} \\ \dot{\psi} \end{bmatrix}, s^+ = \begin{bmatrix} X + v\cos(\psi)\Delta t \\ Y + v\sin(\psi)\Delta t \\ v + \dot{v}\Delta t \\ \psi + \dot{\psi}\Delta t \end{bmatrix} \text{ where}$$

v and $\dot{v}$ are the longitudinal velocity and acceleration, ψ and $\dot{\psi}$ are the heading angle and yaw rate.

The pedestrians follow a double integrator model with

$$s = \begin{bmatrix} X \\ Y \\ v_x \\ v_y \end{bmatrix}, a = \begin{bmatrix} \dot{v}_x \\ \dot{v}_y \end{bmatrix}, s^+ = \begin{bmatrix} X + v_x\Delta t \\ Y + v_y\Delta t \\ v_x + \dot{v}_x\Delta t \\ v_y + \dot{v}_y\Delta t \end{bmatrix}.$$

Indeed, both models consists of basic differentiable functions that can be incorporated in a neural network. We also put bound on the inputs $\dot{v} \in [-5m/s, 5m/s], \dot{\psi} \in [-1m/s^2, 1m/s^2]$, $v_x, v_y \in [-5m/s, 5m/s]$ so that the generated trajectory predictions are dynamically feasible.

A.3. Collision Check

We model pedestrians as circles with varying radius and vehicles as rectangles. The collision between pedestrians is straightforward to check, simply by taking the Euclidean distance between the two pedestrians. Collisions involving vehicles are checked in the local coordinate frame of the vehicle. Fig. 9 shows the case with a pedestrian and a vehicle, and the collision function is

$$\text{Col}(\Delta X, \Delta Y, L, W) = \max\{|\Delta X| - \frac{L}{2}, |\Delta Y| - \frac{W}{2}\}.$$

Vehicle-to-vehicle collision is a bit tricky since it involves two rectangles. As shown in Fig. 10, we use the four corners to calculate the Col function:

$$\text{Col}(\Delta X_{1:4}, \Delta Y_{1:4}, L, W) = \max \left\{ \begin{matrix} |\Delta X_1| - \frac{L}{2}, ...|\Delta X_4| - \frac{L}{2}, \\ |\Delta Y_1| - \frac{W}{2}, ...|\Delta Y_4| - \frac{L}{2} \end{matrix} \right\}.$$

Note that the collision functions are all differentiable (at least piecewise differentiable), making it convenient to include them in the training process as regularization.

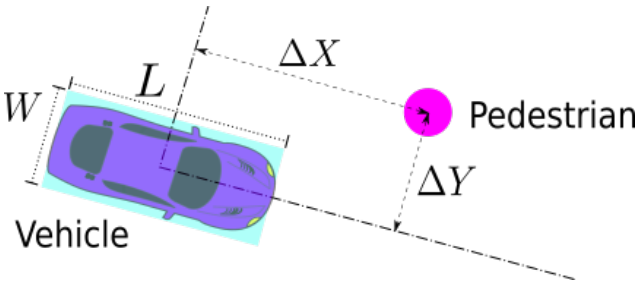

Figure 9. Collision check between a vehicle and a pedestrian

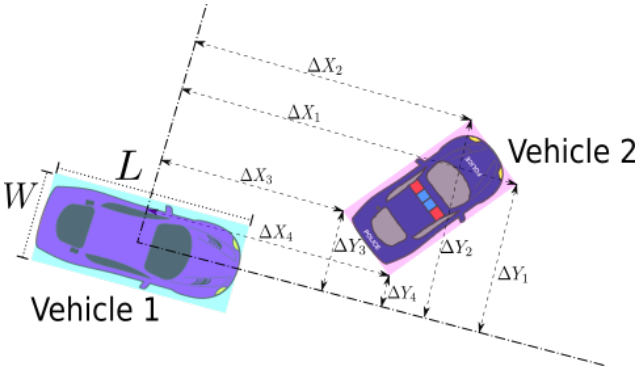

Figure 10. Collision check between two vehicles

A.4. Diversity Scheduling during Training

The parameter α serves as a tuning knob to adjust the tradeoff between encoder accuracy and diversity. During training, we start with a small α so that the decoder can learn diverse trajectory patterns without mode collapse, then increase α to improve the encoder's prediction accuracy. When α is above a threshold, we detach the prediction error loss of all modes but the one with the largest Q in Eq. (4) from the gradient graph to avoid mode collapse. This allows us to reduce the mode collapse under a small α while continue to improve the encoder on the mode probability prediction.

A.5. Diverse Sampling from Product Latent Space

Since $P(\mathbf{z}|\mathbf{x})$ is calculated with factor graphs, two clique modes with similar latent variables, e.g., two $\mathbf{z}$-s that only

differ at one node in the clique, may have similar probabilities,causing the greedy sampling result to lose diversity. This issue is well-known in Markov random fields and we follow the simple diverse sampling scheme in [5]. To be specific, we use a greedy algorithm to pick $\mathbf{z}$ one by one as

$$\begin{aligned}\mathbf{z}^{k+1} = & \arg\max_{\mathbf{z}\in\mathcal{Z}} p(\mathbf{z}|x) \\ & \text{s.t. } \forall \mathbf{z}^i, i = 1, ..., k, \Delta(\mathbf{z}, \mathbf{z}^i) \geq \beta,\end{aligned}$$

where Δ is a distance function, for our setup, $\Delta(\mathbf{z}^1, \mathbf{z}^2) = |\{j|z_j^1 \neq z_j^2\}|$, i.e., the number of nodes with different latent variables under $\mathbf{z}^1$ and $\mathbf{z}^2$. For example, $\Delta([0, 1, 2], [1, 1, 2]) = 1, \Delta([1, 0, 0], [2, 1, 0]) = 2$.