Appendix for AMD

A. Trajectory Augmentation

Data augmentation techniques have been widely adopted to address imbalanced data to enhance model generalization. Inspired by related research [2], this study designs four novel augmentation methods specifically for short-term trajectories: (1) Simplify, (2) Shift, (3) Mask, and (4) Subset. These methods aim to improve the model's performance in handling real-world uncertainties in trajectory prediction. Each augmentation type generates varied transformations, enabling the model to better adapt to different trajectory patterns during training and enhance prediction accuracy for long-tail trajectories.

(1) **Simplify**. Trajectory simplification is achieved by removing redundant nodes in a trajectory while preserving key shape-defining points. This process helps models focus on the primary trajectory patterns. We use the Ramer-Douglas-Peucker (RDP) algorithm for trajectory simplification. The RDP algorithm iteratively removes points between two endpoints while retaining the endpoints of the trajectory. Given a trajectory $T = \{t_1, t_2, \ldots, t_n\}$, where $t_i = (x_i, y_i)$ represents the i-th point in the trajectory, the RDP algorithm yields a simplified trajectory T'.

$$\begin{aligned} &T' = \{t_1, t_k, t_n\} \\ &k = \arg\max_{i=2}^{n-1} (\text{dist}(t_i, \text{line}(t_1, t_n))) \end{aligned} \tag{1}$$

where t_k is the point that satisfies the maximum distance condition.

(2) **Shift**. In long-tail trajectories, data may undergo small random shifts due to external factors, increasing trajectory diversity. To simulate this variation, the shift method applies a random displacement to each point in the trajectory, helping the model adapt to such perturbations. For a given trajectory T, the shift method adds a random displacement to all points over time.

$$T' = \{t_1 + \Delta, t_2 + \Delta, \ldots, t_n + \Delta\} \tag{2}$$

where $\Delta = (\delta_x, \delta_y)$ is a random displacement vector, $\delta_x, \delta_y \sim u(-\epsilon, \epsilon)$, denotes a uniform distribution in the interval $[-\epsilon, \epsilon]$.

(3) **Mask**. The masking method randomly discards parts of trajectory data to simulate missing data caused by sensor failures or other factors. This method enhances the model's robustness in handling trajectories with partial data loss.

$$T' = \begin{cases} t_i, & \text{if mask}(i) = 1 \\ 0, & \text{if mask}(i) = 0 \end{cases} \tag{3}$$

where, the masking indicator mask(i) follows a Bernoulli distribution $B(p)$, with p representing the probability of retaining each point.

(4) **Subset**. Some long trajectories may contain only partial temporal information, making it challenging for the model to learn complete temporal dependencies during training. The subset selection method randomly selects a consecutive subsequence from the trajectory, simulating scenarios with incomplete or unevenly sampled temporal information. For a given trajectory T, the selected subset T' is defined as:

$$T' = [t_i, \ldots, t_{i+\gamma * n}] \tag{4}$$

where γ is the subset ratio, representing the proportion of the trajectory retained in the subset.

B. Momentum Contrastive Learning

In long-tail prediction tasks, rare or uncommon patterns often lead to difficulty in identifying meaningful samples, potentially resulting in these samples being overlooked. To mitigate the error induced by such rare samples in long-tail judgment, we propose an improved Dynamic Momentum and Top-K Hard Negative Mining method, termed MoCo-DK. This method dynamically adjusts the momentum coefficient and employs a Top-K Hard Negative Mining mechanism to apply additional focus on long-tail samples within the contrastive learning framework.

In the original MoCo approach [3], the momentum encoder parameter is updated based on a fixed momentum coefficient, which limits the model's adaptability across different training stages. Therefore, we introduce a dynamic momentum adjustment mechanism, where the momentum coefficient m changes based on the current training progress t and total training duration T, adapting to the feature learning needs at different stages.

$$m = \begin{cases} m_e, & \text{if } t < 0.3T \\ m_m, & \text{if } 0.3T \leq t < 0.7T \\ m_l, & \text{if } t \geq 0.7T \end{cases} \tag{5}$$

where m_e, m_m, m_l represent the momentum coefficients for early, mid, and late stages of training, respectively.

In each update, the parameters θ_k of the momentum encoder are updated based on the parameter θ_q of the query encoder and the dynamic momentum coefficient m.

$$\theta_k \leftarrow m \cdot \theta_k + (1 - m) \cdot \theta_q \tag{6}$$

To encourage the model focus on hard-to-distinguish negative samples, we introduce a Top-K Hard Negative Mining mechanism. This mechanism identifies the most similar negative samples within the negative sample set, enhancing the model's capacity to discriminate among challenging long-tail samples. Specifically, the similarity l_{pos} between the query sample and its corresponding positive sample is calculated, along with the similarity l_{neg} between the query sample and each negative samples. The Top-K samples with the highest similarity to the query sample are then selected from the negative sample set and designated as hard negative samples:

$$\begin{aligned} &l_{pos} = sim(q, k^+), \quad l_{neg} = sim(q, k^-) \\ &\{k_1^-, \ldots, k_K^-\} = TopK_{k^-}(sim(q, k^-)) \end{aligned} \tag{7}$$

where $sim()$ denotes the similarity function, q is the query encoding, and k^+ and k^- represent positive and negative encodings, respectively. K is the number of hard negative samples selected, and $TopK$ refers to selecting the top K samples with the highest similarity to the query sample among the negative samples.

The final contrastive loss is constructed by comparing the positive sample with the Top-K hard negative samples, calculated as follows:

$$L_{moco} = -\log \frac{\exp(sim(q, k^+)/\tau)}{\exp(sim(q, k^+)/\tau) + \sum_{i=1}^{K} \exp(sim(q, k_i^-)/\tau)} \tag{8}$$

where τ is a temperature parameter. This design allows the model to focus on challenging long-tail features.

C. Scene Interaction Encoder

In the Scene Interaction Encoder, we propose a latent variable mechanism to model the multimodal characteristics of trajectory prediction. By introducing stochasticity into the target agent's representation, this mechanism generates multiple plausible future trajectories from a single input scene. Leveraging latent variability, it effectively captures the inherent uncertainty in agent behavior, proving especially valuable for predicting rare yet critical patterns in long-tail scenarios. The mechanism takes the target agent's encoded state $T \in \mathbb{R}^{1 \times B \times D}$ as input, where B denotes the batch size and D represents the feature dimension. This state is first refined through self-attention operations and subsequently transformed into a latent representation H_{latent} via a linear projection function f_{latent}:

$$H_{\text{latent}} = f_{\text{latent}}(T) \in \mathbb{R}^{B \times (D \cdot K)} \tag{9}$$

Here, K denotes the number of trajectory modes, facilitating the encoding of diverse potential outcomes. Subsequently, H_{latent} is reshaped and combined with the target context T to produce a multimodal feature representation $H_{\text{mode}} \in \mathbb{R}^{K \times B \times D}$.

$$H_{\text{mode}} = H_{\text{latent}} + T \tag{10}$$

This approach ensures that H_{mode} encompasses K distinct trajectory modes, each reflecting a unique variation induced by the latent variable mechanism.

D. Experiments

D.1. Datasets

To evaluate the performance of our proposed model across diverse scenarios, we conducted extensive experiments on the nuScenes and ETH/UCY datasets, which encompass traffic data from various real-world scenes. Below is a brief overview of the two datasets:

(1) **nuScenes**: The nuScenes [1] dataset is a large-scale autonomous driving dataset comprising 1000 scenes, spanning a variety of real-world driving scenarios and enriched with high-definition maps for comprehensive contextual information. It provides 2 seconds of historical trajectory data and 6 seconds of future trajectory data for all agents (vehicles and pedestrians).

(2) **ETH/UCY**: The ETH/UCY dataset is a pedestrian dataset consisting of 5 distinct scenes across 56 segments. The ETH [5] dataset includes two scenes, ETH and HOTEL, with 750 pedestrians, while the UCY [4] dataset comprises three scenes—UNIV, ZARA1, and ZARA2—with 786 pedestrians. Sampled at a frequency of 2.5 Hz (i.e., one data point every 0.4 seconds), this dataset provides 3.2 seconds (8 timesteps) of historical trajectory data and 4.8 seconds (12 timesteps) of future trajectory data.

D.2. Metrics

To assess the trajectory prediction performance of our proposed model, we adopt several widely used metrics: Average Displacement Error (ADE), Final Displacement Error (FDE), and Miss Rate (MR). For long-tail samples, we employ minimum ADE (minADE) and minimum FDE (minFDE) to better evaluate performance on challenging cases. Additionally, for overall multi-modal prediction, we

Prediction Horizon (s)	Top 1%	Top 2%	Top 3%	Top 4%	Top 5%	Rest	All
1	0.13/0.11	0.16/0.13	0.15/0.13	0.14/0.11	0.13/0.11	0.14/0.12	0.14/0.12
2	0.20/0.20	0.24/0.24	0.24/0.23	0.22/0.21	0.21/0.20	0.23/0.22	0.21/0.21
3	0.30/0.32	0.33/0.34	0.33/0.34	0.31/0.32	0.30/0.31	0.32/0.34	0.32/0.34
4	0.40/0.44	0.44/0.48	0.43/0.48	0.41/0.45	0.40/0.45	0.42/0.49	0.42/0.48
5	0.52/0.57	0.56/0.63	0.55/0.65	0.53/0.60	0.52/0.60	0.54/0.66	0.54/0.65
6	0.65/0.72	0.72/0.83	0.70/0.87	0.68/0.87	0.67/0.86	0.69/0.88	0.69/0.88

Table 1. Prediction errors (minADE/minFDE) at different prediction horizons, divided by risk metric for the seven test samples. The top 1%-5% refer to the subset of samples with the highest risk.

Prediction Horizon (s)	Rapid Acceleration	Rapid Deceleration	Sharp Lane Change	Sharp Turn	Normal	All
1	0.19/0.16	0.23/0.18	0.26/0.24	0.21/0.19	0.12/0.09	0.14/0.12
2	0.29/0.27	0.34/0.31	0.39/0.40	0.32/0.34	0.19/0.18	0.21/0.21
3	0.39/0.40	0.46/0.46	0.52/0.56	0.44/0.51	0.27/0.29	0.32/0.34
4	0.51/0.57	0.57/0.63	0.68/0.74	0.58/0.71	0.36/0.42	0.42/0.49
5	0.64/0.75	0.72/0.81	0.87/0.97	0.74/0.94	0.48/0.58	0.54/0.65
6	0.80/1.01	0.90/1.08	1.10/1.41	0.94/1.28	0.61/0.78	0.69/0.88

Table 2. Prediction errors (minADE/minFDE) at different prediction horizons, divided by vehicle state for the six test samples.

use minADE$_k$ and minFDE$_k$ to measure the accuracy of the top-K predicted trajectories. Below, we provide the detailed definitions and formulas for these metrics.

- **Average Displacement Error (ADE)**: ADE measures the average Euclidean distance between the predicted trajectory and the ground-truth trajectory over all time steps. For a predicted trajectory $Y_{\text{pred}} = [y_1, y_2, \dots, y_T]$ and ground-truth trajectory $Y_{\text{gt}} = [y_{\text{gt}1}, y_{\text{gt}2}, \dots, y_{\text{gt}T}]$, where T is the prediction horizon, ADE is computed as:

$$\text{ADE} = \frac{1}{T}\sum_{t=1}^{T} \|y_t - y_{\text{gt}_t}\|_2 \tag{11}$$

  Here, $\|\cdot\|_2$ denotes the L_2 norm (Euclidean distance).
- **Final Displacement Error (FDE)**: FDE measures the Euclidean distance between the predicted endpoint and the ground-truth endpoint at the final time step T:

$$\text{FDE} = \|y_T - y_{\text{gt}_T}\|_2 \tag{12}$$

- **Miss Rate (MR)**: MR evaluates the proportion of predictions where the final displacement exceeds a predefined threshold (e.g., 2 meters). For N samples, it is defined as:

$$\text{MR} = \frac{1}{N}\sum_{i=1}^{N} I(\|y_T^i - y_{\text{gt}_T^i}\|_2 > \text{threshold}) \tag{13}$$

  where $I(\cdot)$ is an indicator function (1 if true, 0 otherwise), and y_T^i and $y_{\text{gt}_T^i}$ are the predicted and ground-truth endpoints for sample i.
- **Top-K Minimum ADE (minADE$_k$)**: minADE$_k$ extends minADE to evaluate the best performance among the top-K predicted trajectories, typically used for multi-modal models. It is equivalent to minADE when considering K candidates:

$$\text{minADE}_k = \min_{k=1,\dots,K}\left[\frac{1}{T}\sum_{t=1}^{T} \|y_t^k - y_{\text{gt}_t}\|_2\right] \tag{14}$$

- **Top-K Minimum FDE (minFDE$_k$)**: minFDE$_k$ assesses the best FDE among the top-K predicted trajectories:

$$\text{minFDE}_k = \min_{k=1,\dots,K} \|y_T^k - y_{\text{gt}_T}\|_2 \tag{15}$$

D.3. Implementation Details

Our experimental hyperparameters are as follows: the trajectory simplification threshold is 0.5, translation distance 0.1, trajectory mask 0.8, and subset ratio 0.6. The encoder embedding dimension is 32, with three Transformer Encoder layers, two GAT layers, and four attention heads in the scene interaction module. The model is trained with the Adam optimizer, and final loss weights γ_1, γ_2, λ_1, and λ_2 are set to 1, 0.5, 1, and 0.1, respectively. In MoCo-DT, m_e, m_m, m_l are set to 0.95, 0.99, and 0.999, respectively. The number of clusters in K-means is set to 5.

D.4. Prediction Horizon Impact

To evaluate the proposed model's performance across different prediction horizons, we conducted experiments on the nuScenes dataset, analyzing the impact of prediction durations from 1s to 6s on trajectory prediction accuracy. The evaluation metrics were minADE and minFDE (as defined earlier), with test samples categorized by risk levels and vehicle states to assess the model's effectiveness in long-tail trajectory prediction.

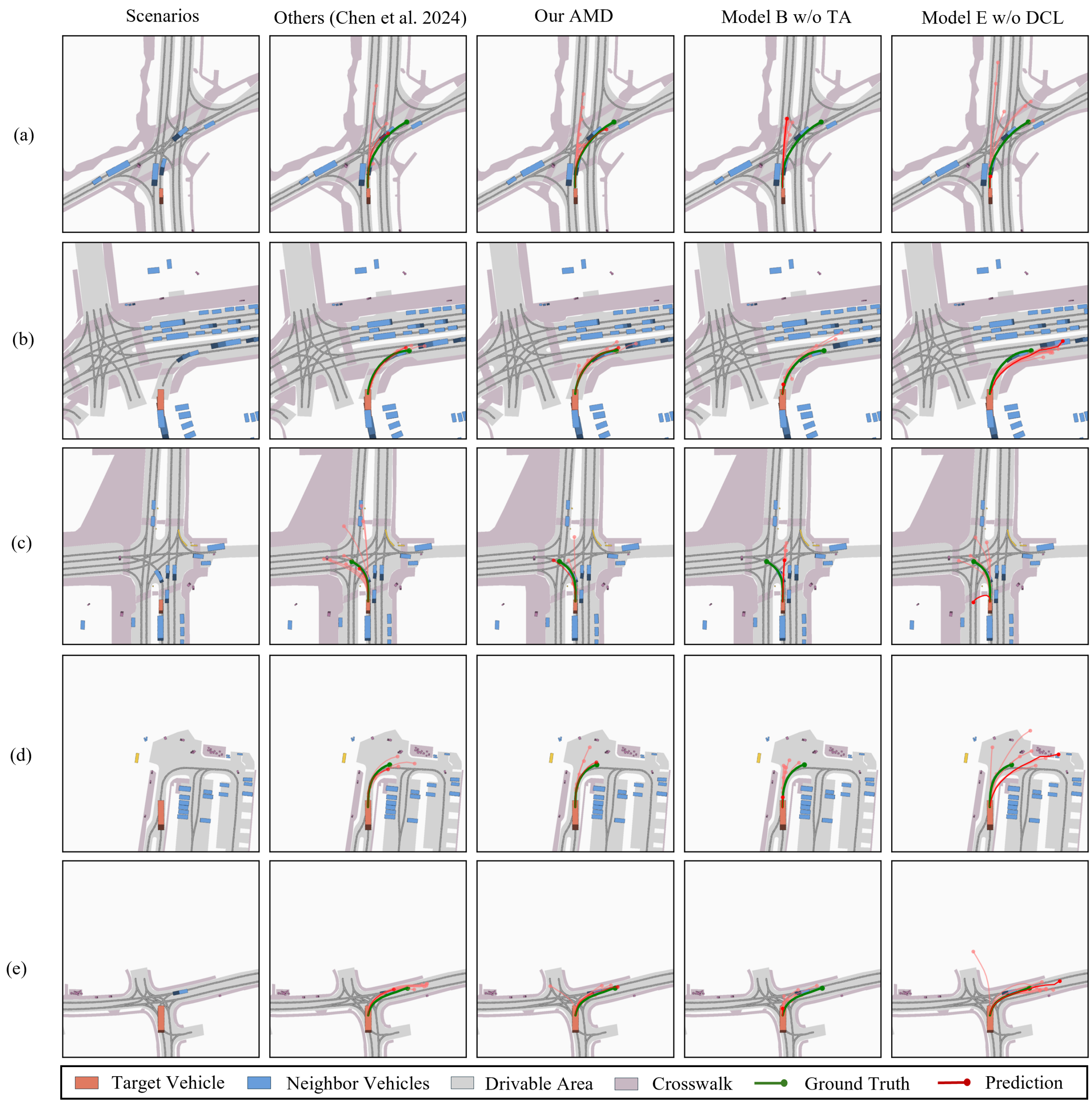

Figure 1. Qualitative results of long-tail trajectory predictions, covering various high-curvature turning trajectories. The red lines show the most probable trajectory, while the light red lines show the predicted multimodal trajectories.

Table 1 shows that minADE and minFDE increase as the prediction horizon extends from 1 to 6 seconds, reflecting accumulated uncertainty. Yet, the model maintains consistency across risk levels; at 6 seconds, the top 1% riskiest samples yield minADE/minFDE of 0.65/0.72, close to the overall average of 0.69/0.88, highlighting its robust performance in high-risk long-tail scenarios.

Table 2 further illustrates the model's strengths across vehicle states like rapid acceleration, deceleration, sharp turns, and normal driving. It achieves lower errors in normal conditions (0.61/0.78 at 6s) while remaining effective in challenging maneuvers like sharp turn (0.94/1.28 at 6s), demonstrating cross-state stability and superior adaptability compared to models overfitting to typical patterns.

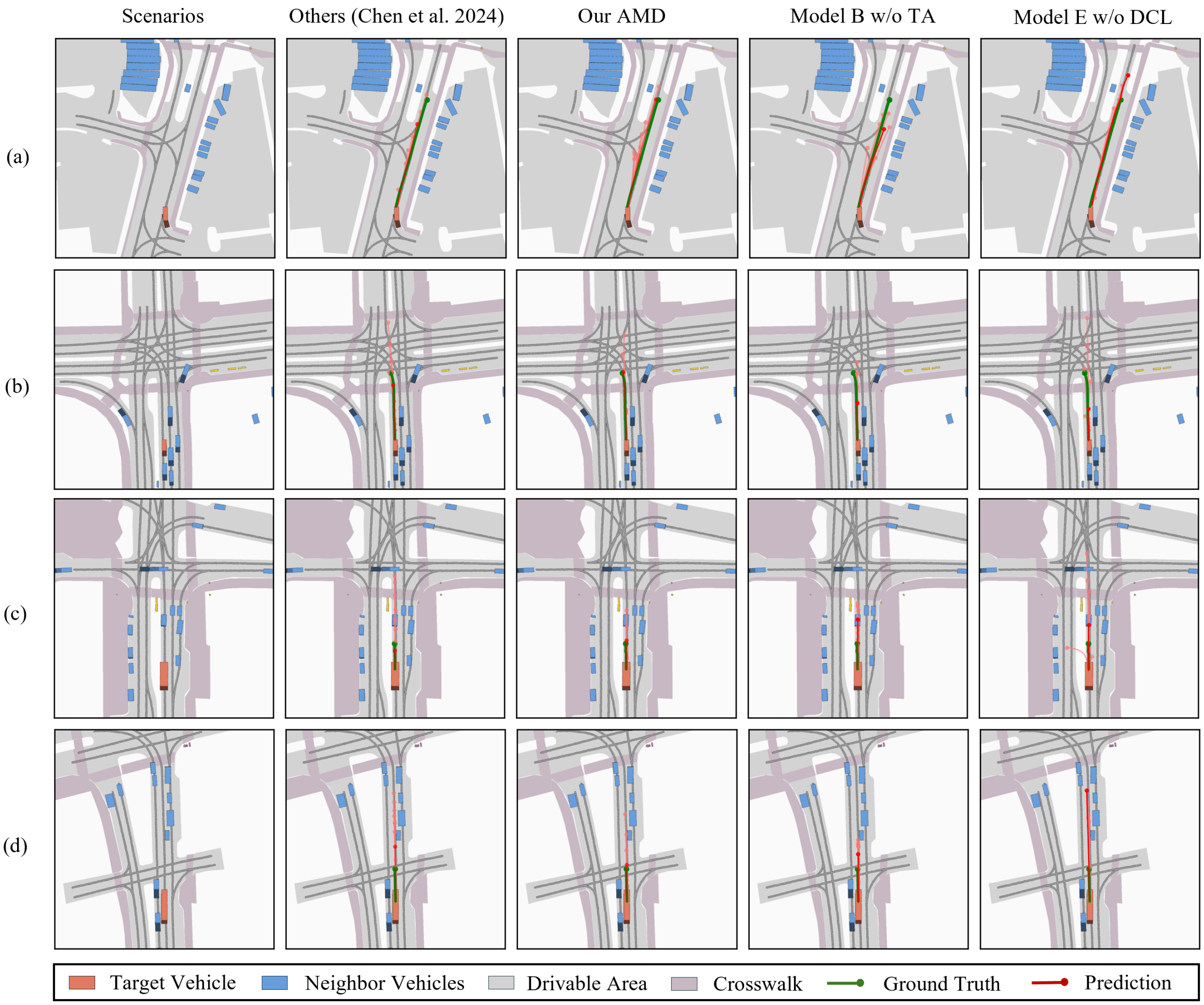

Figure 2. Qualitative results of long-tail trajectory predictions, covering various acceleration and deceleration scenarios. The red lines show the most probable trajectory, while the light red lines show the predicted multimodal trajectories.

D.5. Qualitative Results

To further demonstrate the accuracy of our AMD model in predicting long-tail trajectories, we visualize the prediction results of AMD and its variants on various high-curvature turning trajectories in Figure 1. As illustrated, the AMD model consistently outperforms other variants in scenarios characterized by large turning angles and high trajectory uncertainty, accurately capturing the true vehicle dynamics. Additionally, Figure 2 visualizes vehicle acceleration and deceleration scenarios to evaluate the model's effectiveness under varying speed conditions. The results indicate that AMD effectively captures trajectory fluctuations caused by velocity changes, mitigating prediction latency or overshooting issues commonly observed in the variant models. These observations confirm that our AMD model achieves superior generalization and robustness in long-tail trajectory predictions, exhibiting strong adaptability to complex real-world driving scenarios.

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 *CVPR*, pages 11621–11631, 2020. 2

[2] Yanchuan Chang, Jianzhong Qi, Yuxuan Liang, and Egemen Tanin. Contrastive trajectory similarity learning with dual-feature attention. In *2023 IEEE 39th International conference*

on data engineering (ICDE), pages 2933–2945. IEEE, 2023. 1

[3] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, pages 9729–9738, 2020. 1

[4] Laura Leal-Taixé, Michele Fenzi, Alina Kuznetsova, Bodo Rosenhahn, and Silvio Savarese. Learning an image-based motion context for multiple people tracking. In *CVPR*, pages 3542–3549, 2014. 2

[5] Stefano Pellegrini, Andreas Ess, Konrad Schindler, and Luc Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *ICCV*, pages 261–268. IEEE, 2009. 2