NoisyTraj: Robust Trajectory Prediction with Noisy Observations

Anonymous authors
Paper under double-blind review

Abstract

Trajectory prediction aims to forecast an agent's future trajectories based on its historical observed trajectories, which is a critical task for various applications such as autonomous driving, robotics, and surveillance systems. Most existing trajectory prediction methods assume that the observed trajectories collected for forecasting are clean. However, in real-world scenarios, noise is inevitably introduced into the observations due to errors from sensors, detection, and tracking processes, resulting in the collapse of the existing approaches. Therefore, it is essential to perform robust trajectory prediction based on noisy observations, which is a more practical scenario. In this paper, we propose **NoisyTraj**, a noise-agnostic approach capable of tackling the problem of trajectory prediction with arbitrary types of noisy observations. Specifically, we put forward a mutual information-based mechanism to denoise the original noisy observations. This mechanism optimizes the produced trajectories to exhibit a pattern that closely resembles the clean trajectory pattern while deviating from the noisy one. Considering that the trajectory structure may be destroyed through the only optimization of mutual information, we introduce an additional reconstruction loss to preserve the structure information of the produced observed trajectories. Moreover, we further propose a ranking loss based on the intuitive idea that prediction performance using denoised trajectories should surpass that using the original noisy observations, thereby further enhancing performance. Because NoisyTraj does not rely on any specific module tailored to particular noise distributions, it can handle arbitrary types of noise in principle. Additionally, our proposed NoisyTraj can be easily integrated into existing trajectory prediction models. Extensive experiments conducted on the ETH/UCY and Stanford Drone datasets (SDD) demonstrate that NoisyTraj significantly improves the accuracy of trajectory prediction with noisy observations, compared to the baselines.

1 Introduction

The objective of trajectory prediction is to anticipate the future trajectories for agents given their past observed trajectories, which is an essential and emerging task in numerous applications, such as autonomous driving (Phong et al., 2024; Wang et al., 2023b; Zhou et al., 2023; 2022), drones (Corbetta et al., 2019), surveillance systems (Valera & Velastin, 2005), and robotics (Jetchev & Toussaint, 2009; Rösmann et al., 2017). In recent years, trajectory prediction has garnered significant attention in the computer vision and machine learning communities, with numerous methods proposed (Bae et al., 2023; Choi et al., 2023; Chen et al., 2023b;a). Among these methods, they typically assume the observed historical trajectories are clean, and leverage them to predict future trajectories. Recent advances have demonstrated promising performance in trajectory prediction by learning from such clean observed trajectory data.

However, in real-world scenarios, the acquisition of trajectory data is inevitably accompanied by the introduction of noise. For instance, in an autonomous driving system's trajectory acquisition pipeline, object detection is initially performed to determine the positions and categories of objects. This process is susceptible to noise stemming from sensor errors (e.g., cameras or LiDARs) or inaccuracies in the detection algorithm. Subsequently, object tracking algorithms are employed to associate the same object across multiple timestamps, thereby forming trajectories. At this stage, noise may be introduced due to occlusions and inherent errors in the tracking algorithms.

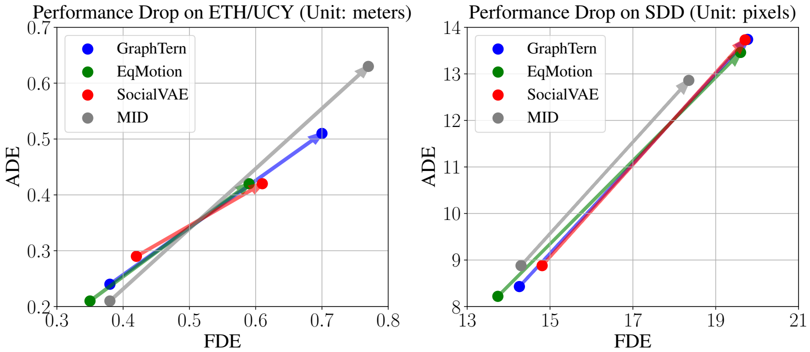


Figure 1: Performance drop of various trajectory prediction methods, including GraphTern (Bae & Jeon, 2023), EqMotion (Xu et al., 2023a), SocialVAE (Xu et al., 2022b) and MID (Gu et al., 2022), on the ETH/UCY (Pellegrini et al., 2009; Leal-Taixé et al., 2014) and Stanford Drone datasets (SDD) (Robicquet et al., 2016). The start of each arrow indicates the performance under clean observations, while the end represents the degraded performance under noisy observations (We add Gaussian noise $\mathcal{N}(0, \sigma = 0.4)$ to the clean observations.). Best viewed in color.

The presence of noise in the observed trajectories significantly hinders the performance of existing trajectory prediction methods. To substantiate this point, we corrupt the input observed trajectories by adding noise in both the training and testing stages. We then conduct experiments on the ETH/UCY (Pellegrini et al., 2009; Leal-Taixé et al., 2014) and Stanford Drone datasets (SDD) (Robicquet et al., 2016) using several recently proposed trajectory prediction approaches. We compare the performance before and after introducing noise to the observations. As shown in Figure 1, the presence of noise in the observed trajectories leads to a significant performance drop for various trajectory prediction methods on both the ETH/UCY and SDD datasets. Specifically, on the ETH/UCY dataset, the Final Displacement Error (FDE) increases from approximately 0.35-0.45 meters in the clean observation setting to 0.60-0.75 meters in the noisy observation setting. Similarly, on the SDD dataset, the FDE rises from around 13-15 pixels to 18-20 pixels This significant performance degradation highlights the detrimental impact of noise on trajectory prediction accuracy, even for state-of-the-art models. Therefore, it is crucial to devise a robust method for predicting future trajectories based on noisy observations.

In this paper, we propose **NoisyTraj**, a noise-agnostic method designed to address the challenge of trajectory prediction with arbitrary types of noisy observations. Specifically, we first propose a mutual information-based mechanism to filter noise from the original observations. This mechanism ensures the produced trajectories closely resemble the patterns of noise-free trajectories while deviating from the noisy patterns. To this end, we maximize the mutual information between the produced trajectories and the clean future trajectories (i.e., ground-truth), while simultaneously minimizing the mutual information between the produced trajectories and the original noisy observations. In this way, the produced trajectories are forced to collate information from both the noisy trajectories and clean future trajectories, thereby preserving the necessary information while filtering out noise. However, solely relying on optimizing mutual information for denoising may disrupt the structure of the trajectory. Therefore, we propose to randomly mask several observations and attempt to reconstruct the masked locations. By jointly optimizing the mutual information and reconstruction losses, the trajectory denoise model can effectively eliminate noise while preserving the structure information of the trajectory. In the meantime, we design a ranking loss to facilitate the ability of the trajectory prediction module based on an intuitive thought: predictions using the produced denoised observations will be superior to those using noisy observations. It is noteworthy that the ranking loss optimizes not only the trajectory prediction module but also the denoising module, which can further assist in filtering noise to some extent. Since NoisyTraj does not rely on any specific module tailored to a particular noise distribution, it can handle arbitrary noise in principle. Essentially, our proposed NoisyTraj is a plug-and-play approach that is compatible with existing trajectory prediction models, enabling them to gracefully handle cases with noisy observations.

Our main contributions are summarized as follows: 1) We investigate a new problem setting for trajectory prediction with noisy observations, addressing a more practical scenario. To tackle this, we propose a noise-agnostic, plug-and-play approach called NoisyTraj. 2) We design a denoising module that incorporates a mutual information-based loss along with a reconstruction loss, effectively denoising observed trajectories while preserving their structural information. 3) We propose a ranking loss to ensure that denoised observations yield superior future predictions compared to their noisy counterparts, thereby enhancing the accuracy of trajectory predictions. 4) We conduct extensive experiments on the ETH/UCY and SDD datasets, demonstrating that our method significantly outperforms the baselines in predicting trajectory with noisy observations.

2 Related Works

2.1 Trajectory Prediction with Clean Observations

Trajectory prediction has been an active area of research in the computer vision and machine learning communities. Early works employ physics-based methods to model the trajectories of agents (Luber et al., 2010; Pellegrini et al., 2009). Subsequently, learning-based approaches are proposed, which significantly enhance the performance of trajectory prediction (Zhu et al., 2023a; Rowe et al., 2023; Wang et al., 2023a; Xu et al., 2023b). They model trajectory temporal information and the interaction between agents (Alahi et al., 2016; Altché & de La Fortelle, 2017; Shi et al., 2022; Xue et al., 2018; Mohamed et al., 2020). One representative approach is social pooling, which aggregates hidden state information of neighbors within a spatial grid (Gupta et al., 2018; Sadeghian et al., 2019). Additionally, attention mechanisms (Fernando et al., 2018; Vemula et al., 2018), graph neural networks (Li et al., 2019; Kosaraju et al., 2019; Sun et al., 2020b) and transformers (Yuan et al., 2021; Zhu et al., 2023b; Shi et al., 2023) have also been exploited to model interactions among agents. To further enhance prediction performance, researchers delve into incorporating the map information. Works such as (Shafiee et al., 2021; Dendorfer et al., 2021; Sun et al., 2020a; Mangalam et al., 2021; Meng et al., 2022) encode RGB scene information, while (Ye et al., 2021; Gu et al., 2021; Zhao et al., 2021; Kang et al., 2024) incorporate lane and road traffic information. Moreover, due to the inherent uncertainty associated with agents, researchers have proposed a series of models to predict multiple plausible future trajectories, including GANs (Liang et al., 2021; Li, 2019; Zhao et al., 2019), VAEs (Lee et al., 2022; 2017; Sun et al., 2021), and diffusion models (Rempe et al., 2023; Li et al., 2024b; Jiang et al., 2023). Recently, several new task settings have been introduced to address more practical trajectory prediction problems, including momentary trajectory prediction (Li et al., 2024a; Monti et al., 2022; Sun et al., 2022; Li et al., 2024b), long-tailed distribution in trajectory prediction (Zhang et al., 2024; Mercurius et al., 2024; Wang et al., 2023c), and distribution shift in trajectory prediction (Stoler et al., 2023; Xu et al., 2022c; Kong et al., 2024).

Despite these methods having shown promising performance, they rely on sufficiently clean observed trajectories. As aforementioned, when the observed trajectories are corrupted by noise, the model performance severely deteriorates. In contrast to these approaches, we attempt to tackle the problem of predicting future trajectories with noisy observed trajectories.

2.2 Trajectory Anomaly Detection

The goal of trajectory anomaly detection is to identify abnormal patterns in trajectories, such as abnormal deviations, trajectory repetitions, and missing segments. Trajectory anomaly detection methods can be categorized as supervised learning, semi-supervised learning, and unsupervised learning approaches (Fan et al., 2009; Quispe-Torres et al., 2021; Zhang et al., 2018; Sillito & Fisher, 2008; Chebiyyam et al., 2018; Jiao et al., 2023; Mondal et al., 2021; Liatsikou et al., 2021). Supervised anomaly detection entails training a deep supervised binary or multi-class classifier using labeled data of both normal and anomalous trajectories. For instance, the work in (Chebiyyam et al., 2018) extracts statistical features from trajectories to train a multi-class SVM for classifying trajectories as normal or anomalous. Despite their promising performance, supervised methods require substantial effort to label trajectory data. To mitigate this labeling burden, researchers have explored semi-supervised anomaly detection, where only normal trajectory data are labeled. A common approach involves employing deep autoencoders trained in a semi-supervised manner (Minhas & Zelek, 2020; Song et al., 2017). These methods assume that the autoencoder will accurately encode and decode normal samples while failing to reconstruct anomalous data. Moreover, in scenarios where labeled data is scarce or unavailable, researchers have proposed unsupervised methods for trajectory anomaly detection by leveraging intrinsic data properties. Clustering is a popular unsupervised technique for trajectory anomaly detection. For example. Hu et al. (2006) clusters trajectories based on spatial and temporal information, and each motion pattern is represented with a chain of Gaussian distributions. Then, they detect anomalies based on these motion patterns. Fan et al. (Fan et al., 2009) propose

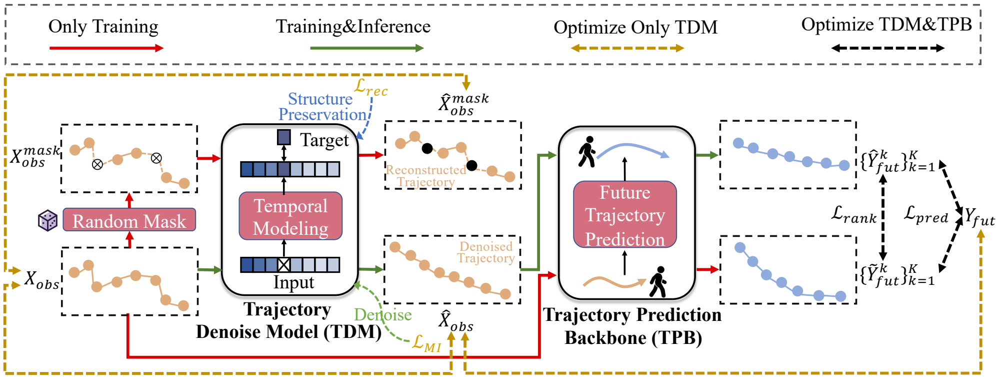


Figure 2: Overview of the proposed NoisyTraj framework. The framework is composed of two modules: a Trajectory Denoise Model (TDM) and a Trajectory Prediction Backbone (TPB). The $\mathcal{L}_{MI}$ denoises the produced trajectories $\hat{X}_{obs}$ by maximizing mutual information between $\hat{X}_{obs}$ and the clean ground-truth Y_{fut}, while minimizing the mutual information between $\hat{X}$ and noisy observations X_{obs}. The $\mathcal{L}_{rec}$ reconstructs the masked location of the original trajectories. By jointly optimizing $\mathcal{L}_{rec}$ and $\mathcal{L}_{MI}$, the trajectory denoise model can learn to denoise trajectories while preserving the structure information. The ranking loss $\mathcal{L}_{rank}$ constrains the predictions based on the denoised observations $\hat{X}_{obs}$ to be superior to those based on the noisy observations X_{obs}, thereby further filtering out noise and enhance the ability of trajectory prediction.

to represent trajectories with Hidden Markov Models (HMMs) and propose a dynamic hierarchical clustering method to differential normal and abnormal patterns.

While trajectory anomaly detection can identify abnormal trajectories, they cannot usually correct these trajectories. In contrast to these methods, our objective is to eliminate noise from observed trajectories and enhance the performance of trajectory prediction.

3 Methods

3.1 Problem Formulation

Let $X_{obs} = \{x_{obs}^1, x_{obs}^2, \ldots, x_{obs}^{T_{obs}}\}$ denote the observed trajectories, where T_{obs} is the observation length, and $x_{obs}^i \in \mathbb{R}^2$ is the i^{th} location. We assume the observations are given by $X_{obs} = S_{obs} + N$, where $S_{obs} = \{s_{obs}^1, s_{obs}^2, \ldots, s_{obs}^{T_{obs}}\}$ are clean observed trajectories, and N is noise sample from an arbitrary distribution, such as Gaussian noise and Possion noise. Moreover, we denote the ground-truth future trajectories as $Y_{fut} = \{y_{fut}^1, y_{fut}^2, \ldots, y_{fut}^{T_{fut}}\}$, where $y_{fut}^i \in \mathbb{R}^2$ represents the i^{th} locations, and T_{fut} is the length of the ground-truth future trajectories. In this work, we simplify the problem by assuming Y_{fut} are clean, and only the observed trajectory is noisy, which is a reasonable assumption [1]. Different from previous works that typically utilize clean observations S_{obs} for future trajectory prediction, our goal is to develop a robust trajectory prediction method using noisy observations, which is a more practical scenario. Specifically, we aim to use noisy observations X_{obs} to forecast K plausible future trajectories $\{\hat{Y}_{fut}\}_{k=1}^K$ under the supervision of Y_{fut}.

3.2 Overall Framework

The overall framework of the proposed NoisyTraj is shown in Figure 2. Our framework consists of two parts: a Trajectory Denoise Model (TDM) and a Trajectory Prediction Backbone (TPB). To eliminate noise from the noisy observations X_{obs}, we first propose a mutual information-based mechanism, which encourages the produced trajectories $\hat{X}_{obs}$ exhibit patterns similar to the noise-

[1] In practical scenarios, we can use an autonomous vehicle equipped with both cameras and LiDAR to collect training data. We can treat camera-derived trajectories as noisy data and LiDAR-derived trajectories as clean ground-truth for training. Once the model is learned based on the training data, we can deploy it on a vehicle equipped with only cameras for trajectory prediction with noisy observations.

free ground-truth future trajectories Y_{fut}, while deviating from the patterns of the noisy observed trajectories X_{obs}. This is achieved through a loss function $\mathcal{L}_{MI}$ that simultaneously maximizes the mutual information between $\hat{X}_{obs}$ and Y_{fut} while minimizing the mutual information between $\hat{X}_{obs}$ and X_{obs}. In this way, the produced trajectories $\hat{X}_{obs}$ are forced to collate information from both the noisy trajectories X_{obs} and clean future trajectories Y_{fut}, thereby filtering out noise. Given that only optimizing mutual information-based loss potentially disrupts the structure of the trajectories, we propose a reconstruction strategy to mitigate this issue. Specifically, we randomly mask M locations of the noisy observations X_{obs} to obtain X_{obs}^{mask}. The X_{obs}^{mask} is then fed into the TDM to reconstruct the masked portion of the original noisy input X_{obs} using $\mathcal{L}_{rec}$. By jointly optimizing $\mathcal{L}_{MI}$ and $\mathcal{L}_{rec}$, the TDM is able to learn to denoise while preserving the structure information of the trajectories. To facilitate more accurate trajectory prediction, we devise a ranking loss. We first input both the denoised observations $\hat{X}_{obs}$ and the original noisy observations X_{obs} into the TPB to forecast future trajectories. Then the ranking loss is applied to encourage the future trajectories predicted from the denoised observations to be more precise than those predicted from the noisy observations, thereby enhancing the trajectory prediction performance. The TDM and TPB modules can benefit from each other: the ranking loss in TPB helps TDM filter noise more effectively, while the denoised trajectories generated by TDM enable TPB to predict future trajectories more accurately. As NoisyTraj is not dependent on any module specialized for a particular noise distribution, it is capable of handling arbitrary noise in principle. In addition, NoisyTraj is essentially a plug-and-play approach and can be readily integrated into existing trajectory prediction models, enabling them to effectively handle scenarios with noisy observations.

3.3 Trajectory Prediction with Noisy Observations

In this section, we introduce the details of our NoisyTraj. We first present the mutual information-based denoising mechanism, followed by the designed ranking loss.

3.3.1 Mutual Information-Based Denoising Mechanism.

Given noisy observations X_{obs}, we expect the noise can be eliminated through a trajectory denoising model $\Phi_{\mathbf{TDM}}$. Inspired by Information Bottleneck (Tishby et al., 2000), we encourage the produced trajectories to exhibit patterns closely resembling noise-free trajectory patterns while deviating from noisy patterns. This is achieved by maximizing the mutual information between the produced trajectories and noise-free ground-truth future trajectories Y_{fut} while minimizing the mutual information between the produced trajectories and the original noisy observations X_{obs}. We define the objective function as:

$$J_{MI} = \min_{\hat{X}_{obs}} \alpha I(X_{obs}; \hat{X}_{obs}) - I(\hat{X}_{obs}; Y_{fut}), \tag{1}$$

where $I(\cdot\,;\,\cdot)$ represents the mutual information and α is a trade-off parameter.

However, directly calculating J_{MI} is intractable. Therefore, we estimate the upper bound of $I(X_{obs}; \hat{X}_{obs})$ by utilizing CLUB (Cheng et al., 2020), and the lower bound of $I(\hat{X}_{obs}; Y_{fut})$ by leveraging the method described in MINE (Belghazi et al., 2018). We first calculate the upper bound of $I(X_{obs}; \hat{X}_{obs})$.

Theorem 3.1. *Given two random variables x and y, the mutual information $I(x; y)$ has the following upper bound*

$$I(x; y) \leq \mathbb{E}_{p(x,y)}[\log p(y|x)] - \mathbb{E}_{p(x)}\mathbb{E}_{p(y)}[\log p(y|x)]. \tag{2}$$

Proof. See proof in the Appendix 6.3.

By substituting X_{obs} and $\hat{X}_{obs}$ to the Equation (2), we can obtain the upper bound of $I(X_{obs}; \hat{X}_{obs})$:

$$I(X_{obs}; \hat{X}_{obs}) \leq \mathbb{E}_{p(X_{obs},\hat{X}_{obs})}[\log p(\hat{X}_{obs}|X_{obs})] - \mathbb{E}_{p(X_{obs})}\mathbb{E}_{p(\hat{X}_{obs})}[\log p(\hat{X}_{obs}|X_{obs})]. \tag{3}$$

Since $p(\hat{X}_{obs}|X_{obs})$ is unknown, we introduce a variational approximation distribution $q_\phi(\hat{X}_{obs}|X_{obs})$ to approximate $p(\hat{X}_{obs}|X_{obs})$ with parameter ϕ, following (Cheng et al., 2020).

Thus, the upper bound can be written as:

$$
\begin{aligned}
I(X_{obs}; \hat{X}_{obs}) &\leq I_{\mu}(X_{obs}; \hat{X}_{obs}) \\
&= \mathbb{E}_{p(X_{obs}, \hat{X}_{obs})}[\log q_{\phi}(\hat{X}_{obs}|X_{obs})] - \mathbb{E}_{p(X_{obs})}\mathbb{E}_{p(\hat{X}_{obs})}[\log q_{\phi}(\hat{X}_{obs}|X_{obs})]. \quad (4)
\end{aligned}
$$

Next, we calculate the lower bound of the mutual information $I(\hat{X}_{obs}; Y_{fut})$.

Theorem 3.2 (Donsker-Varadhan representation (Donsker & Varadhan, 1983))**.** *Given two probability distributions $\mathbb{P}$, $\mathbb{Q}$. The Kullback Liebler Divergence admits the following dual representation:*

$$
D_{KL}(\mathbb{P}||\mathbb{Q}) = \sup_{T:\Omega\to\mathbb{R}} \mathbb{E}_{\mathbb{P}}[T] - \log \mathbb{E}_{\mathbb{Q}}[e^{T}], \quad (5)
$$

Proof. See the proof in the Appendix 6.4.

Based on the Theorem 3.2, we can obtain the $I(\hat{X}_{obs}, Y_{fut})$ by:

$$
\begin{aligned}
I(\hat{X}_{obs}, Y_{fut}) &= D_{KL}(p(\hat{X}_{obs}, Y_{fut})||p(\hat{X}_{obs})p(Y_{fut})) \quad (6) \\
&= \sup_{T:\Omega\to\mathbb{R}} \mathbb{E}_{p(\hat{X}_{obs}, Y_{fut})}[T] - \log \mathbb{E}_{p(\hat{X}_{obs})p(Y_{fut})}[e^{T}], \quad (7)
\end{aligned}
$$

where $\Omega = \hat{X}_{obs} \times Y_{fut}$ is the input space. Let $\mathcal{F}$ be any class of functions $T : \Omega \to \mathbb{R}$, and the lower bound of $I(\hat{X}_{obs}, Y_{fut})$ can be expressed as:

$$
I(\hat{X}_{obs}, Y_{fut}) \geq I_{\mathcal{F}}(\hat{X}_{obs}, Y_{fut}) = \sup_{T\in\mathcal{F}} \mathbb{E}_{p(\hat{X}_{obs}, Y_{fut})}[T] - \log \mathbb{E}_{p(\hat{X}_{obs})p(Y_{fut})}[e^{T}]. \quad (8)
$$

We choose $\mathcal{F}$ to be the family of functions $T_{\psi} : \hat{X}_{obs} \times Y_{fut} \to \mathbb{R}$, parameterized by a neural network ψ. Thus, the lower bound can be written as:

$$
I(\hat{X}_{obs}, Y_{fut}) \geq I_{\psi}(\hat{X}_{obs}, Y_{fut}) = \sup_{\psi} \mathbb{E}_{p(\hat{X}_{obs}, Y_{fut})}[T_{\psi}] - \log \mathbb{E}_{p(\hat{X}_{obs})p(Y_{fut})}[e^{T_{\psi}}]. \quad (9)
$$

Based on Equation (4) and (9), we derive the upper bound $\mathcal{L}_{MI}$ of the J_{MI} as:

$$
\begin{aligned}
J_{MI} \leq \mathcal{L}_{MI} &= \alpha I_{\mu}(X_{obs}; \hat{X}_{obs}) - I_{\psi}(\hat{X}_{obs}, Y_{fut}) \\
&= \alpha \mathbb{E}_{p(X_{obs}, \hat{X}_{obs})}[\log q_{\phi}(\hat{X}_{obs}|X_{obs})] - \mathbb{E}_{p(X_{obs})}\mathbb{E}_{p(\hat{X}_{obs})}[\log q_{\phi}(\hat{X}_{obs}|X_{obs})] \\
&\quad - \sup_{\psi} \mathbb{E}_{p(\hat{X}_{obs}, Y_{fut})}[T_{\psi}] + \log \mathbb{E}_{p(\hat{X}_{obs})p(Y_{fut})}[e^{T_{\psi}}]. \quad (10)
\end{aligned}
$$

By minimizing the upper bound $\mathcal{L}_{MI}$, we can obtain an approximation solution to Equation (1), enabling the trajectory denoise model to learn how to denoise trajectories.

However, only optimizing the mutual information may destroy the structure of the produced trajectories. Therefore, we propose a reconstruction strategy to preserve the structure information. As shown in the left part of Figure 2, we mask locations within the noisy observed trajectories X_{obs} to generate X_{obs}^{mask}, which can be formulated as:

$$
X_{obs}^{mask} = X_{obs} \odot \mathcal{M}_{obs}, \quad (11)
$$

where $\odot$ represents the element-wise multiplication. $\mathcal{M}_{obs}$ is a 0-1 mask vector, where the value **0** represents the corresponding locations are masked. Subsequently, the X_{obs}^{mask} is fed into the trajectory denoise model $\Phi_{\textbf{TDM}}$ to produce observed trajectories $\hat{X}_{obs}^{mask}$. To enable TDM to preserve the structural information of the trajectories, we reconstruct the masked locations in the produced trajectories. We define the reconstruction loss as follows:

$$
\mathcal{L}_{rec} = \mathcal{J}(\hat{X}_{obs}^{mask} \odot (1 - \mathcal{M}_{obs}), X_{obs} \odot (1 - \mathcal{M}_{obs})), \quad (12)
$$

where $\mathcal{J}$ denotes the distance metric, and we empirically adopt L_2 distance in this work. Through optimizing $\mathcal{L}_{rec}$, the trajectory denoise model can learn to preserve the structure information of the observations. By jointly optimizing the reconstruction loss $\mathcal{L}_{rec}$ with $\mathcal{L}_{MI}$, the mutual information-based mechanism effectively denoises the trajectories while preserving their structural information.

3.3.2 Trajectory Prediction Based on Ranking Loss.

After obtaining the denoised observed trajectories $\hat{X}_{obs}$, we design a ranking loss to enhance future trajectory prediction performance. The ranking loss is based on an intuitive thought: leveraging denoised observed trajectories should yield more accurate future predictions compared to using noisy observations. To accomplish this, we first input the denoised observations into the trajectory prediction backbone $\Phi_{\textbf{TPB}}$. Then, we can predict K plausible future trajectories:

$$\{\hat{Y}_{fut}^k\}_{k=1}^K = \Phi_{\textbf{TPB}}(\Phi_{\textbf{TDM}}(X_{obs})), \tag{13}$$

Similarly, we can also predict K possible trajectories based on the noisy observed trajectories:

$$\{\tilde{Y}_{fut}^k\}_{k=1}^K = \Phi_{\textbf{TPB}}(X_{obs}). \tag{14}$$

After obtaining K possible trajectories based on the denoised and noisy observed trajectories, respectively, we then select the minimal distances $d_{denoise}$ and d_{noise} by calculating the distances between each predicted trajectory and ground-truth trajectory, respectively. Formally,

$$d_{denoise} = \min_{1 \leq k \leq K} ||\hat{Y}_{fut}^k - Y_{fut}||_2, \;\; d_{noise} = \min_{1 \leq k \leq K} ||\tilde{Y}_{fut}^k - Y_{fut}||_2. \tag{15}$$

Then, we employ the ground-truth future trajectories as supervision for the best-predicted trajectory:

$$\mathcal{L}_{pred} = ||\hat{Y}_{fut}^{best} - Y_{fut}||_2 + ||\tilde{Y}_{fut}^{best} - Y_{fut}||_2, \tag{16}$$

where $best$ represents the trajectory with a minimal distance to the ground-truth. Subsequently, we design a ranking loss to constrain the best prediction $\hat{Y}_{fut}^{best}$ using the denoised observations to be more accurate than that $\tilde{Y}_{fut}^{best}$ using the noisy observations:

$$\mathcal{L}_{rank} = \max(0, d_{denoise} - d_{noise} + \Delta), \tag{17}$$

where Δ is a margin. Since the ranking loss optimizes both the trajectory prediction backbone and the trajectory denoise model, it not only aids in better trajectory prediction but also facilitates the denoising ability of the trajectory denoise model.

3.4 Optimization and Inference

Optimization. We define the total loss function as:

$$\mathcal{L} = \mathcal{L}_{pred} + \beta\mathcal{L}_{rank} + \delta\mathcal{L}_{rec} + \gamma\mathcal{L}_{MI}, \tag{18}$$

where β, δ, and γ are trade-off hyperparameters. The training details are shown in Appendix 6.7.

Inference. After training, the model can be utilized for trajectory prediction based on noisy observations. As shown in the blue arrow in Figure 2, we first feed the noisy trajectories X_{obs} into the trajectory denoise model $\Phi_{\textbf{TDM}}$ to obtain denoised trajectories $\hat{X}_{obs}$. Subsequently, the trajectory prediction backbone $\Phi_{\textbf{TPB}}$ takes $\hat{X}_{obs}$ as input to obtain the predicted future trajectories $\{\hat{Y}_{fut}^k\}_{k=1}^K$.

4 Experiments

4.1 Experiment Settings

Dataset. We evaluate our proposed NoisyTraj on two widely used datasets: the ETH/UCY (Pellegrini et al., 2009; Leal-Taixé et al., 2014) and SDD dataset (Robicquet et al., 2016). The ETH/UCY dataset is composed of 5 scenes, including ETH, HOTEL, UNIV, ZARA1, and ZARA2, with 1,536 pedestrians recorded in total. Following (Huang et al., 2019; Xu et al., 2022b; Mangalam et al., 2020; Bae et al., 2023), we adopt the "leave-one-out" strategy, where the models are trained on 4 scenes and tested on the remaining scene. SDD consists of 20 scenes captured using a drone in a top-down view around the university campus containing several moving agents such as humans, bicyclists, skateboarders, and vehicles, which contains 5,232 trajectories in total. We follow a common setting among existing works, where 8 frames of trajectories (3.2 seconds) are used as observations to predict the next 12 frames (Wong et al., 2022; Gu et al., 2022). To verify the robustness of NoisyTraj against noise, we add noise into existing publicly available trajectory prediction datasets, ETH/UCY and

Table 1: Comparison of different methods on the ETH/UCY dataset. The evaluation metrics are ADE and FDE (Unit: meters). The best results are highlighted in **bold**.

Noise	Method	ETH		HOTEL		UNIV		ZARA1		ZARA2		AVG	
		ADE	FDE	ADE	FDE	ADE	FDE	ADE	FDE	ADE	FDE	ADE	FDE
$\sigma = 0.2$	GraphTern	0.56	0.81	0.27	0.39	0.39	0.59	0.37	0.58	0.34	0.50	0.39	0.57
	EqMotion	0.51	0.70	0.17	0.24	0.36	0.56	0.33	0.54	0.25	0.37	0.32	0.48
	MID	0.74	0.86	0.37	0.36	0.45	0.58	0.43	0.52	0.40	0.47	0.48	0.56
	SocialImplict	0.67	1.28	0.31	0.49	0.46	0.77	0.39	0.71	0.35	0.63	0.44	0.78
	SocialVAE	0.56	0.89	0.18	0.25	0.40	0.63	0.32	0.49	0.25	0.38	0.34	0.53
	Wavelet+GraphTern	0.50	0.72	0.23	0.35	0.37	0.56	0.32	0.50	0.30	0.42	0.34	0.51
	EMA+GraphTern	0.53	0.76	0.26	0.37	0.38	0.57	0.34	0.54	0.31	0.48	0.36	0.53
	NoisyTraj+GraphTern	**0.48**	**0.68**	**0.19**	**0.26**	**0.35**	**0.53**	**0.28**	**0.44**	**0.27**	**0.36**	**0.31**	**0.45**
	Wavelet+EqMotion	0.48	0.64	**0.16**	0.22	0.32	0.52	0.31	0.48	0.23	0.32	0.30	0.44
	EMA+EqMotion	0.49	0.66	**0.16**	0.22	0.34	0.53	0.31	0.48	0.23	0.33	0.31	0.44
	NoisyTraj+EqMotion	**0.47**	**0.61**	**0.16**	**0.21**	**0.29**	**0.47**	**0.28**	**0.44**	**0.21**	**0.29**	**0.28**	**0.40**
$\sigma = 0.4$	GraphTern	0.67	0.99	0.40	0.51	0.49	0.69	0.51	0.71	0.46	0.61	0.51	0.70
	EqMotion	0.65	0.87	0.25	0.34	0.43	0.64	0.42	0.62	0.34	0.48	0.42	0.59
	MID	0.88	1.08	0.51	0.49	0.60	0.73	0.59	0.84	0.57	0.72	0.63	0.77
	SocialImplict	0.74	1.39	0.40	0.64	0.54	0.86	0.59	0.88	0.53	0.76	0.56	0.61
	SocialVAE	0.67	1.01	0.24	0.32	0.48	0.73	0.41	0.58	0.31	0.43	0.42	0.61
	Wavelet+GraphTern	0.60	0.83	0.33	0.43	0.46	0.64	0.42	0.63	0.37	0.49	0.44	0.60
	EMA+GraphTern	0.64	0.92	0.36	0.46	0.46	0.66	0.47	0.67	0.41	0.58	0.47	0.66
	NoisyTraj+GraphTern	**0.55**	**0.77**	**0.29**	**0.41**	**0.42**	**0.61**	**0.38**	**0.56**	**0.34**	**0.42**	**0.40**	**0.55**
	Wavelet+EqMotion	0.62	0.74	0.23	0.29	0.41	0.58	0.39	0.56	0.32	0.43	0.39	0.52
	EMA+EqMotion	0.63	0.75	0.23	0.29	0.42	0.63	0.40	0.58	0.31	0.42	0.43	0.53
	NoisyTraj+EqMotion	**0.57**	**0.71**	**0.20**	**0.25**	**0.35**	**0.51**	**0.35**	**0.51**	**0.29**	**0.41**	**0.35**	**0.48**

SDD. We employ two settings: i) we first add Gaussian noise $\mathcal{N}(0, \sigma)$ based on the Central Limit Theorem (Kwak & Kim, 2017), which suggests that combination of various noise sources—such as sensor error, detection error, and tracking error—tend to approximate a Gaussian distribution. To verify the effectiveness of NoisyTraj under different levels of noise, we set σ to different values, e.g., 0.2 and 0.4. (Forde & Daniel, 2021); ii) to verify the noise-agnostic property of NoisyTraj, we add various types of noise including Poisson noise, mixed noise, and multiplicative noise.

Evaluation Metrics. Following previous works (Mao et al., 2023; Gu et al., 2022; Sadeghian et al., 2019; Shi et al., 2021), we employ Average Displacement Error (ADE) and Final Displacement Error (FDE) to evaluate the predicted trajectories. ADE is the average L2 error between all future timesteps, and FDE is the error at the final timestamp. We take the best out of $K = 20$ predictions to account for the multi-modality for trajectory prediction, as in (Salzmann et al., 2020; Xu et al., 2022a).

Backbones and Compared Baselines. To validate the efficacy of NoisyTraj, we integrate it into two popular trajectory prediction backbones GraphTern (Bae & Jeon, 2023) and EqMotion (Xu et al., 2023a). We first compare our method against five state-of-the-art trajectory prediction models, including **GraphTern**, **EqMotion**, **MID** (Gu et al., 2022), **SocialImplict** (Mohamed et al., 2022) and **SocialVAE** (Xu et al., 2022b). These methods take original noisy observations as input to predict future trajectories. Considering there are few works focusing on trajectory prediction with noisy observations, we establish two trajectory denoising baselines by integrating Wavelet and EMA with the trajectory prediction backbones, respectively, for a more comprehensive comparison. The **Wavelet** utilizes the wavelet transform to decompose the signal into multiple scales, obtaining wavelet coefficients of different frequencies. Then the noise in high-frequency coefficients is removed by the thresholding method. The **EMA** smooths the current trajectory location by taking an exponentially weighted average of the current location and past locations.

4.2 Results and Analysis

Performance on Trajectory Predictions with Noisy Observations. We evaluate the performance of our proposed NoisyTraj and compare it with various baselines on the ETH/UCY and SDD datasets. The results are listed in Table 1 and Table 2. Based on the two tables, NoisyTraj+GraphTern and NoisyTraj+EqMotion significantly outperforms GraphTern and EqMotion on the two datasets under

Table 2: Comparison of different methods on the SDD dataset. The evaluation metrics are ADE and FDE (Unit: pixels). The best results are highlighted in **bold**.

Noise	Method	SDD	
		ADE	FDE
$\sigma = 0.2$	GraphTern	11.67	18.37
	EqMotion	10.62	15.68
	MID	10.26	15.38
	SocialImplicit	15.92	26.82
	SocialVAE	11.67	17.62
	Wavelet+GraphTern	10.52	16.86
	EMA+GraphTern	11.03	17.42
	NoisyTraj+GraphTern	**10.08**	**15.64**
	Wavelet+EqMotion	10.36	15.24
	EMA+EqMotion	10.32	15.22
	NoisyTraj+EqMotion	**10.06**	**14.67**

Noise	Method	SDD	
		ADE	FDE
$\sigma = 0.4$	GraphTern	13.74	19.77
	EqMotion	13.46	19.60
	MID	12.86	18.35
	SocialImplicit	18.07	29.98
	SocialVAE	13.73	19.71
	Wavelet+GraphTern	12.98	18.50
	EMA+GraphTern	13.21	19.11
	NoisyTraj+GraphTern	**12.35**	**17.28**
	Wavelet+EqMotion	12.38	18.25
	EMA+EqMotion	12.79	18.64
	NoisyTraj+EqMotion	**11.92**	**17.65**

the setting of $\sigma = 0.2$ and $\sigma = 0.4$ meters. This illustrates current state-of-the-art methods cannot well tackle the case of noisy observations. However, when integrating our proposed NoisyTraj into these two models, the performance can be significantly improved. This demonstrates the effectiveness of our method for trajectory prediction with noisy observations, and also highlights its compatibility with different trajectory prediction models. Furthermore, NoisyTraj outperforms the Kalman and EMA denoising methods, further underscoring the superiority of our proposed approach.

Ablation Studies. We conduct ablation studies on the components of our proposed method. We utilize GraphTern as the backbone and set σ to 0.4 meters. The results are listed in Table 3. We first incorporate the mutual information loss $\mathcal{L}_{MI}$ into GraphTern, the performance is improved, demonstrating our denoising mechanism is effective. We then add $\mathcal{L}_{rec}$ into our method to reconstruct the masked locations for preserving the structure information of the trajectories. We observe a further improvement in performance, which demonstrates its effectiveness. Finally, we add the ranking loss $\mathcal{L}_{rank}$, which enables our method to achieve the best performance. This indicates that $\mathcal{L}_{rank}$ enhances the capability of the trajectory prediction model.

Table 3: Ablation Studies on each component of NoisyTraj. The best results are highlighted in **bold.**

Component			ETH/UCY		SDD	
$\mathcal{L}_{MI}$	$\mathcal{L}_{rec}$	$\mathcal{L}_{rank}$	ADE	FDE	ADE	FDE
			0.51	0.70	13.74	19.77
✓			0.47	0.65	13.21	18.79
✓	✓		0.42	0.59	12.74	17.96
✓	✓	✓	**0.40**	**0.55**	**12.35**	**17.28**

Table 4: Comparison of different methods under different noise setting on the SDD dataset. The evaluation metrics are ADE and FDE (Unit: pixels). The best results are highlighted in **bold**.

(a) Possion Noise ($\lambda = 0.4$).

Noise	Method	SDD	
		ADE	FDE
$\lambda = 0.4$	EqMotion	14.05	19.46
	Wavelet+EqMotion	12.95	17.97
	EMA+EqMotion	13.15	17.58
	NoisyTraj+EqMotion	**12.22**	**16.23**

(b) Mixed noise composed of Gaussian noise ($\sigma = 0.2$) and Poisson noise ($\lambda = 0.2$).

Noise	Method	SDD	
		ADE	FDE
$\sigma = 0.2$ $\lambda = 0.2$	EqMotion	14.66	20.27
	Wavelet+EqMotion	13.48	18.30
	EMA+EqMotion	13.82	18.97
	NoisyTraj+EqMotion	**12.96**	**17.09**

(c) Noise randomly multiplied by $\delta \in [0.95, 1.0]$.

Noise	Method	SDD	
		ADE	FDE
$\delta \in [0.95, 1.0]$	EqMotion	17.48	19.10
	Wavelet+EqMotion	16.17	18.18
	EMA+EqMotion	16.25	18.38
	NoisyTraj+EqMotion	**15.66**	**17.39**

(d) Gaussian Noise sampled from $\sigma \in \{0.2, 0.4\}$.

Noise	Method	SDD	
		ADE	FDE
$\sigma \in \{0.2, 0.4\}$	EqMotion	13.33	18.92
	Wavelet+EqMotion	12.82	15.66
	EMA+EqMotion	12.76	15.50
	NoisyTraj+EqMotion	**12.28**	**15.15**

Performance under Different Noise Setting. We conduct experiments to verify the effectiveness of NoisyTraj for various noise settings. we added (1) Poisson noise, (2) Mixed noise (Gaussian + Poisson), (3) Noise randomly multiplied by $\delta \in [0.95, 1]$ and (4) Gaussian noise randomly sampled from $\sigma \in \{0.2, 0.4\}$. The results are listed in Table 4. We observe our method consistently outperforms the baselines across various settings, which demonstrates the effectiveness of Noisytraj and it is agnostic to noise distributions in principle.

Generalizability of NoisyTraj. To verify the generalizability of our method, we conduct additional experiments where the noise in the training and validation/testing set is different. Specifically, we train the model using trajectories with Gaussian noise ($\sigma = 0.4$) and then test it with Gaussian noise ($\sigma = 0.2$) and Poisson noise ($\lambda = 0.4$). The results, as listed in Table 5, show that our NoisyTraj can achieve denoising effectively, and still outperforms the baselines. This also indicates that our method possesses generalization ability when noise is different in the training and testing/validation set. Therefore, we believe our method still works when facing real-world noisy trajectories.

Table 5: Comparison of different methods when noise is different between training and testing on the SDD dataset. The evaluation metrics are ADE and FDE (Unit: pixels). The best results are highlighted in **bold**.

Noise	Method	SDD	
		ADE	FDE
Train:	EqMotion	11.47	16.82
$\sigma = 0.4$	Wavelet+EqMotion	10.86	16.07
Test:	EMA+EqMotion	10.99	16.24
$\sigma = 0.2$	**NoisyTraj+EqMotion**	**10.72**	**15.89**

Noise	Method	SDD	
		ADE	FDE
Train:	EqMotion	15.03	19.35
$\sigma = 0.4$	Wavelet+EqMotion	14.40	18.56
Test:	EMA+EqMotion	14.57	18.98
$\lambda = 0.4$	**NoisyTraj+EqMotion**	**14.26**	**18.32**

Qualitative Results. We visualize the denoised observations and predicted future trajectories generated by EWA, Wavelet, and NoisyTraj, using GraphTern as the backbone. The results are shown in Figure 3. We observe that NoisyTraj can generate less noisy observed trajectories and more accurate future trajectories compared to other methods. This demonstrates the proposed mutual information-based mechanism effectively denoises the observations, and the ranking loss aids in forecasting more precise future trajectories.

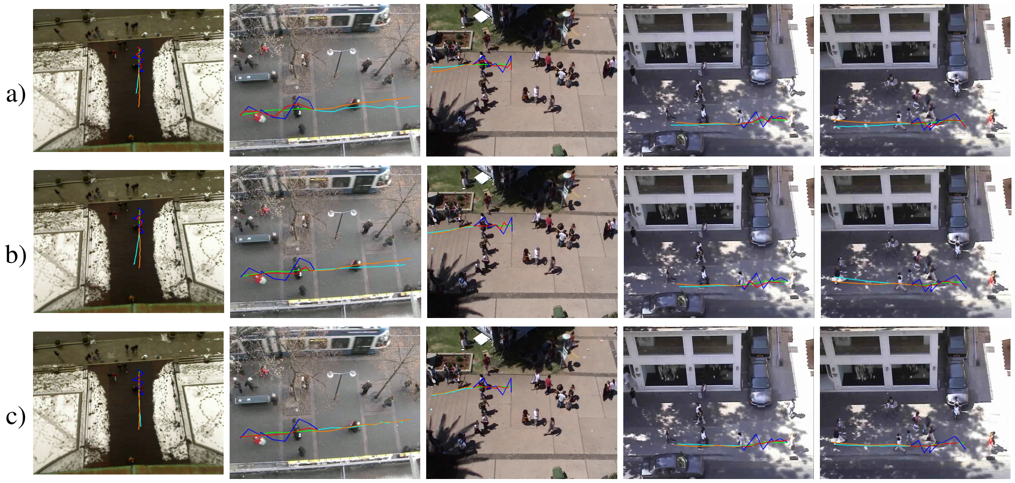


Figure 3: Visualization of predicted trajectories via (a) EMA+GraphTern, (b) Wavelet+GraphTern, (c) NoisyTraj+GraphTern on the ETH/UCY Dataset. The clean, noisy, and denoised observations are shown in green, blue, and red, respectively. The noisy observations are obtained through adding Gaussian noise $\mathcal{N}(0, \sigma = 0.4)$ into clean observations. The ground-truth and predicted future trajectories are shown in orange and cyan, respectively.

5 Conclusion

In this paper, we investigated an extremely challenging task of trajectory prediction with noisy observations. We proposed NoisyTraj, a framework that simultaneously filters out noise and predicts future trajectories, enabling them to benefit from each other. To remove the noise from the observations, we designed a denoising mechanism by jointly optimizing a mutual information-based loss and a reconstruction loss. Moreover, we devised a ranking loss that requires the prediction performance using denoised observed trajectories to be superior to that using the original noisy observations, thereby further improving the performance of the model. Extensive experiments demonstrated the effectiveness of NoisyTraj and its compatibility with various trajectory prediction models.

REFERENCES

Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 961–971, 2016.

Florent Altché and Arnaud de La Fortelle. An lstm network for highway trajectory prediction. In *2017 IEEE 20th international conference on intelligent transportation systems (ITSC)*, pp. 353–359. IEEE, 2017.

Inhwan Bae and Hae-Gon Jeon. A set of control points conditioned pedestrian trajectory prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 6155–6165, 2023.

Inhwan Bae, Jean Oh, and Hae-Gon Jeon. Eigentrajectory: Low-rank descriptors for multi-modal trajectory forecasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10017–10029, 2023.

Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. Mutual information neural estimation. In *International conference on machine learning*, pp. 531–540. PMLR, 2018.

Manaswi Chebiyyam, Rohit Desam Reddy, Debi Prosad Dogra, Harish Bhaskar, and Lyudmila Mihaylova. Motion anomaly detection and trajectory analysis in visual surveillance. *Multimedia Tools and Applications*, 77:16223–16248, 2018.

Guangyi Chen, Zhenhao Chen, Shunxing Fan, and Kun Zhang. Unsupervised sampling promoting for stochastic human trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17874–17884, 2023a.

Hao Chen, Jiaze Wang, Kun Shao, Furui Liu, Jianye Hao, Chenyong Guan, Guangyong Chen, and Pheng-Ann Heng. Traj-mae: Masked autoencoders for trajectory prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8351–8362, 2023b.

Pengyu Cheng, Weituo Hao, Shuyang Dai, Jiachang Liu, Zhe Gan, and Lawrence Carin. Club: A contrastive log-ratio upper bound of mutual information. In *International conference on machine learning*, pp. 1779–1788. PMLR, 2020.

Sehwan Choi, Jungho Kim, Junyong Yun, and Jun Won Choi. R-pred: Two-stage motion prediction via tube-query attention-based trajectory refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8525–8535, 2023.

Matteo Corbetta, Portia Banerjee, Wendy Okolo, George Gorospe, and Dmitry G Luchinsky. Real-time uav trajectory prediction for safety monitoring in low-altitude airspace. In *Aiaa aviation 2019 forum*, pp. 3514, 2019.

Patrick Dendorfer, Sven Elflein, and Laura Leal-Taixé. Mg-gan: A multi-generator model preventing out-of-distribution samples in pedestrian trajectory prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 13158–13167, 2021.

Monroe D Donsker and SR Srinivasa Varadhan. Asymptotic evaluation of certain markov process expectations for large time. iv. *Communications on pure and applied mathematics*, 36(2):183–212, 1983.

Jiang Fan, Ying Wu, and AK Katsaggelos. A dynamic hierarchical clustering method for trajectory-based unusual video event detection. *IEEE Transactions on Image Processing*, 18(4):907–913, 2009.

Tharindu Fernando, Simon Denman, Sridha Sridharan, and Clinton Fookes. Soft+ hardwired attention: An lstm framework for human trajectory prediction and abnormal event detection. *Neural networks*, 108:466–478, 2018.

Albert Forde and Janice Daniel. Pedestrian walking speed at un-signalized midblock crosswalk and its impact on urban street segment performance. *Journal of Traffic and Transportation Engineering (English Edition)*, 8(1):57–69, 2021. ISSN 2095-7564. doi: https://doi.org/10.1016/j.jtte.2019.03.007. URL `https://www.sciencedirect.com/science/article/pii/S209575641830415X`.

Junru Gu, Chen Sun, and Hang Zhao. Densetnt: End-to-end trajectory prediction from dense goal sets. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15303–15312, 2021.

Tianpei Gu, Guangyi Chen, Junlong Li, Chunze Lin, Yongming Rao, Jie Zhou, and Jiwen Lu. Stochastic trajectory prediction via motion indeterminacy diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17113–17122, 2022.

Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2255–2264, 2018.

Weiming Hu, Xuejuan Xiao, Zhouyu Fu, Dan Xie, Tieniu Tan, and Steve Maybank. A system for learning statistical motion patterns. *IEEE transactions on pattern analysis and machine intelligence*, 28(9):1450–1464, 2006.

Yingfan Huang, Huikun Bi, Zhaoxin Li, Tianlu Mao, and Zhaoqi Wang. Stgat: Modeling spatial-temporal interactions for human trajectory prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6272–6281, 2019.

Nikolay Jetchev and Marc Toussaint. Trajectory prediction: learning to map situations to robot trajectories. In *Proceedings of the 26th annual international conference on machine learning*, pp. 449–456, 2009.

Chiyu Jiang, Andre Cornman, Cheolho Park, Benjamin Sapp, Yin Zhou, Dragomir Anguelov, et al. Motiondiffuser: Controllable multi-agent motion prediction using diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9644–9653, 2023.

Ruochen Jiao, Juyang Bai, Xiangguo Liu, Takami Sato, Xiaowei Yuan, Qi Alfred Chen, and Qi Zhu. Learning representation for anomaly detection of vehicle trajectories. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 9699–9706. IEEE, 2023.

Miao Kang, Shengqi Wang, Sanping Zhou, Ke Ye, Jingjing Jiang, and Nanning Zheng. Ffinet: Future feedback interaction network for motion forecasting. *IEEE Transactions on Intelligent Transportation Systems*, 2024.

Hao Kong, Jie Xu, Shenjian Gong, Jian Yang, and Shanshan Zhang. Adaptive pedestrian trajectory prediction via target-directed angle augmentation. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4065–4069. IEEE, 2024.

Vineet Kosaraju, Amir Sadeghian, Roberto Martín-Martín, Ian Reid, Hamid Rezatofighi, and Silvio Savarese. Social-bigat: Multimodal trajectory forecasting using bicycle-gan and graph attention networks. *Advances in neural information processing systems*, 32, 2019.

Sang Gyu Kwak and Jong Hae Kim. Central limit theorem: the cornerstone of modern statistics. *Korean journal of anesthesiology*, 70(2):144–156, 2017.

Laura Leal-Taixé, Michele Fenzi, Alina Kuznetsova, Bodo Rosenhahn, and Silvio Savarese. Learning an image-based motion context for multiple people tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3542–3549, 2014.

Mihee Lee, Samuel S Sohn, Seonghyeon Moon, Sejong Yoon, Mubbasir Kapadia, and Vladimir Pavlovic. Muse-vae: Multi-scale vae for environment-aware long term trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2221–2230, 2022.

Namhoon Lee, Wongun Choi, Paul Vernaza, Christopher B Choy, Philip HS Torr, and Manmohan Chandraker. Desire: Distant future prediction in dynamic scenes with interacting agents. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 336–345, 2017.

Rongqing Li, Changsheng Li, Yuhang Li, Hanjie Li, Yi Chen, Ye Yuan, and Guoren Wang. Itpnet: Towards instantaneous trajectory prediction for autonomous driving. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1643–1654, 2024a.

Rongqing Li, Changsheng Li, Dongchun Ren, Guangyi Chen, Ye Yuan, and Guoren Wang. Bcdiff: Bidirectional consistent diffusion for instantaneous trajectory prediction. *Advances in Neural Information Processing Systems*, 36, 2024b.

Xin Li, Xiaowen Ying, and Mooi Choo Chuah. Grip: Graph-based interaction-aware trajectory prediction. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 3960–3966. IEEE, 2019.

Yuke Li. Which way are you going? imitative decision learning for path forecasting in dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 294–303, 2019.

Rongqin Liang, Yuanman Li, Xia Li, Yi Tang, Jiantao Zhou, and Wenbin Zou. Temporal pyramid network for pedestrian trajectory prediction with multi-supervision. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 2029–2037, 2021.

Maria Liatsikou, Symeon Papadopoulos, Lazaros Apostolidis, and Ioannis Kompatsiaris. A denoising hybrid model for anomaly detection in trajectory sequences. In *EDBT/ICDT Workshops*, 2021.

Matthias Luber, Johannes A Stork, Gian Diego Tipaldi, and Kai O Arras. People tracking with human motion predictions from social forces. In *2010 IEEE international conference on robotics and automation*, pp. 464–469. IEEE, 2010.

Karttikeya Mangalam, Harshayu Girase, Shreyas Agarwal, Kuan-Hui Lee, Ehsan Adeli, Jitendra Malik, and Adrien Gaidon. It is not the journey but the destination: Endpoint conditioned trajectory prediction. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pp. 759–776. Springer, 2020.

Karttikeya Mangalam, Yang An, Harshayu Girase, and Jitendra Malik. From goals, waypoints & paths to long term human trajectory forecasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15233–15242, 2021.

Weibo Mao, Chenxin Xu, Qi Zhu, Siheng Chen, and Yanfeng Wang. Leapfrog diffusion model for stochastic trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5517–5526, 2023.

Mancheng Meng, Ziyan Wu, Terrence Chen, Xiran Cai, Xiang Zhou, Fan Yang, and Dinggang Shen. Forecasting human trajectory from scene history. *Advances in Neural Information Processing Systems*, 35:24920–24933, 2022.

Ray Coden Mercurius, Ehsan Ahmadi, Soheil Mohamad Alizadeh Shabestary, and Amir Rasouli. Amend: A mixture of experts framework for long-tailed trajectory prediction. *arXiv preprint arXiv:2402.08698*, 2024.

Manpreet Singh Minhas and John Zelek. Semi-supervised anomaly detection using autoencoders. *arXiv preprint arXiv:2001.03674*, 2020.

Abduallah Mohamed, Kun Qian, Mohamed Elhoseiny, and Christian Claudel. Social-stgcnn: A social spatio-temporal graph convolutional neural network for human trajectory prediction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 14424–14432, 2020.

Abduallah Mohamed, Deyao Zhu, Warren Vu, Mohamed Elhoseiny, and Christian Claudel. Social-implicit: Rethinking trajectory prediction evaluation and the effectiveness of implicit maximum likelihood estimation. In *European Conference on Computer Vision*, pp. 463–479. Springer, 2022.

Suman Mondal, Arindam Roy, and Sukumar Mandal. A supervised trajectory anomaly detection using velocity and path deviation. In *Proceedings of International Conference on Frontiers in Computing and Systems: COMSYS 2020*, pp. 777–784. Springer, 2021.

Alessio Monti, Angelo Porrello, Simone Calderara, Pasquale Coscia, Lamberto Ballan, and Rita Cucchiara. How many observations are enough? knowledge distillation for trajectory forecasting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6553–6562, 2022.

Stefano Pellegrini, Andreas Ess, Konrad Schindler, and Luc Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *2009 IEEE 12th international conference on computer vision*, pp. 261–268. IEEE, 2009.

Tran Phong, Haoran Wu, Cunjun Yu, Panpan Cai, Sifa Zheng, and David Hsu. What truly matters in trajectory prediction for autonomous driving? *Advances in Neural Information Processing Systems*, 36, 2024.

Gerar F Quispe-Torres, Germain Garcia-Zanabria, Harley Vera-Olivera, and Lauro Enciso-Rodas. Trajectory anomaly detection based on similarity analysis. In *2021 XLVII Latin American Computing Conference (CLEI)*, pp. 1–10. IEEE, 2021.

Davis Rempe, Zhengyi Luo, Xue Bin Peng, Ye Yuan, Kris Kitani, Karsten Kreis, Sanja Fidler, and Or Litany. Trace and pace: Controllable pedestrian animation via guided trajectory diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13756–13766, 2023.

Alexandre Robicquet, Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Learning social etiquette: Human trajectory understanding in crowded scenes. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14*, pp. 549–565. Springer, 2016.

Christoph Rösmann, Malte Oeljeklaus, Frank Hoffmann, and Torsten Bertram. Online trajectory prediction and planning for social robot navigation. In *2017 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, pp. 1255–1260. IEEE, 2017.

Luke Rowe, Martin Ethier, Eli-Henry Dykhne, and Krzysztof Czarnecki. Fjmp: Factorized joint multi-agent motion prediction over learned directed acyclic interaction graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13745–13755, 2023.

Amir Sadeghian, Vineet Kosaraju, Ali Sadeghian, Noriaki Hirose, Hamid Rezatofighi, and Silvio Savarese. Sophie: An attentive gan for predicting paths compliant to social and physical constraints. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1349–1358, 2019.

Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16*, pp. 683–700. Springer, 2020.

Nasim Shafiee, Taskin Padir, and Ehsan Elhamifar. Introvert: Human trajectory prediction via conditional 3d attention. In *Proceedings of the IEEE/cvf Conference on Computer Vision and Pattern recognition*, pp. 16815–16825, 2021.

Liushuai Shi, Le Wang, Chengjiang Long, Sanping Zhou, Mo Zhou, Zhenxing Niu, and Gang Hua. Sgcn: Sparse graph convolution network for pedestrian trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8994–9003, 2021.

Liushuai Shi, Le Wang, Chengjiang Long, Sanping Zhou, Fang Zheng, Nanning Zheng, and Gang Hua. Social interpretable tree for pedestrian trajectory prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 2235–2243, 2022.

Liushuai Shi, Le Wang, Sanping Zhou, and Gang Hua. Trajectory unified transformer for pedestrian trajectory prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9675–9684, 2023.

Rowland R Sillito and Bob Fisher. Semi-supervised learning for anomalous trajectory detection. In *Proceedings British Machine Vision Conference BMVC2008*, pp. 1035–1044, 2008.

Hongchao Song, Zhuqing Jiang, Aidong Men, Bo Yang, et al. A hybrid semi-supervised anomaly detection model for high-dimensional data. *Computational intelligence and neuroscience*, 2017, 2017.

Benjamin Stoler, Ingrid Navarro, Meghdeep Jana, Soonmin Hwang, Jonathan Francis, and Jean Oh. Safeshift: Safety-informed distribution shifts for robust trajectory prediction in autonomous driving. *arXiv preprint arXiv:2309.08889*, 2023.

Hao Sun, Zhiqun Zhao, and Zhihai He. Reciprocal learning networks for human trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7416–7425, 2020a.

Jianhua Sun, Qinhong Jiang, and Cewu Lu. Recursive social behavior graph for trajectory prediction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 660–669, 2020b.

Jianhua Sun, Yuxuan Li, Hao-Shu Fang, and Cewu Lu. Three steps to multimodal trajectory prediction: Modality clustering, classification and synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 13250–13259, 2021.

Jianhua Sun, Yuxuan Li, Liang Chai, Hao-Shu Fang, Yong-Lu Li, and Cewu Lu. Human trajectory prediction with momentary observation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6467–6476, 2022.

Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

Maria Valera and Sergio A Velastin. Intelligent distributed surveillance systems: a review. *IEE Proceedings-Vision, Image and Signal Processing*, 152(2):192–204, 2005.

Anirudh Vemula, Katharina Muelling, and Jean Oh. Social attention: Modeling attention in human crowds. In *2018 IEEE international Conference on Robotics and Automation (ICRA)*, pp. 4601–4607. IEEE, 2018.

Mingkun Wang, Xinge Zhu, Changqian Yu, Wei Li, Yuexin Ma, Ruochun Jin, Xiaoguang Ren, Dongchun Ren, Mingxu Wang, and Wenjing Yang. Ganet: Goal area network for motion forecasting. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1609–1615. IEEE, 2023a.

Xishun Wang, Tong Su, Fang Da, and Xiaodong Yang. Prophnet: Efficient agent-centric motion forecasting with anchor-informed proposals. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21995–22003, 2023b.

Yuning Wang, Pu Zhang, Lei Bai, and Jianru Xue. Fend: A future enhanced distribution-aware contrastive learning framework for long-tail trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1400–1409, 2023c.

Conghao Wong, Beihao Xia, Ziming Hong, Qinmu Peng, Wei Yuan, Qiong Cao, Yibo Yang, and Xinge You. View vertically: A hierarchical network for trajectory prediction via fourier spectrums. In *European Conference on Computer Vision*, pp. 682–700. Springer, 2022.

Chenxin Xu, Maosen Li, Zhenyang Ni, Ya Zhang, and Siheng Chen. Groupnet: Multiscale hypergraph neural networks for trajectory prediction with relational reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6498–6507, 2022a.

Chenxin Xu, Robby T Tan, Yuhong Tan, Siheng Chen, Yu Guang Wang, Xinchao Wang, and Yanfeng Wang. Eqmotion: Equivariant multi-agent motion prediction with invariant interaction reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1410–1420, 2023a.

Pei Xu, Jean-Bernard Hayet, and Ioannis Karamouzas. Socialvae: Human trajectory prediction using timewise latents. In *European Conference on Computer Vision*, pp. 511–528. Springer, 2022b.

Yi Xu, Lichen Wang, Yizhou Wang, and Yun Fu. Adaptive trajectory prediction via transferable gnn. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6520–6531, 2022c.

Yi Xu, Armin Bazarjani, Hyung-gun Chi, Chiho Choi, and Yun Fu. Uncovering the missing pattern: Unified framework towards trajectory imputation and prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9632–9643, 2023b.

Hao Xue, Du Q Huynh, and Mark Reynolds. Ss-lstm: A hierarchical lstm model for pedestrian trajectory prediction. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1186–1194. IEEE, 2018.

Maosheng Ye, Tongyi Cao, and Qifeng Chen. Tpcn: Temporal point cloud networks for motion forecasting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11318–11327, 2021.

Ye Yuan, Xinshuo Weng, Yanglan Ou, and Kris M Kitani. Agentformer: Agent-aware transformers for socio-temporal multi-agent forecasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9813–9823, 2021.

Junrui Zhang, Mozhgan Pourkeshavarz, and Amir Rasouli. Tract: A training dynamics aware contrastive learning framework for long-tail trajectory prediction. *arXiv preprint arXiv:2404.12538*, 2024.

Ya-Lin Zhang, Longfei Li, Jun Zhou, Xiaolong Li, and Zhi-Hua Zhou. Anomaly detection with partially observed anomalies. In *Companion Proceedings of the The Web Conference 2018*, pp. 639–646, 2018.

Hang Zhao, Jiyang Gao, Tian Lan, Chen Sun, Ben Sapp, Balakrishnan Varadarajan, Yue Shen, Yi Shen, Yuning Chai, Cordelia Schmid, et al. Tnt: Target-driven trajectory prediction. In *Conference on Robot Learning*, pp. 895–904. PMLR, 2021.

Tianyang Zhao, Yifei Xu, Mathew Monfort, Wongun Choi, Chris Baker, Yibiao Zhao, Yizhou Wang, and Ying Nian Wu. Multi-agent tensor fusion for contextual trajectory prediction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12126–12134, 2019.

Zikang Zhou, Luyao Ye, Jianping Wang, Kui Wu, and Kejie Lu. Hivt: Hierarchical vector transformer for multi-agent motion prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8823–8833, 2022.

Zikang Zhou, Jianping Wang, Yung-Hui Li, and Yu-Kai Huang. Query-centric trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17863–17873, 2023.

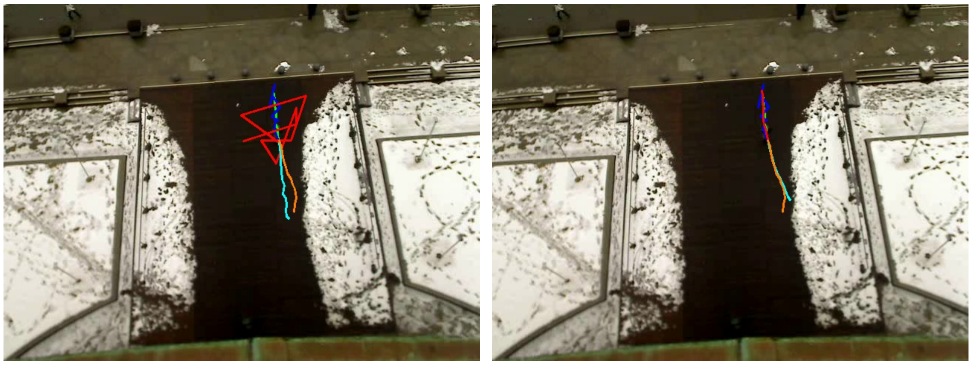
Dekai Zhu, Guangyao Zhai, Yan Di, Fabian Manhardt, Hendrik Berkemeyer, Tuan Tran, Nassir Navab, Federico Tombari, and Benjamin Busam. Ipcc-tp: Utilizing incremental pearson correlation coefficient for joint multi-agent trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5507–5516, 2023a.

Yiyao Zhu, Di Luan, and Shaojie Shen. Biff: Bi-level future fusion with polyline-based coordinate for interactive trajectory prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8260–8271, 2023b.

6 Appendix

6.1 Visualization of Mutual Information-Based Denoising Mechanism

To further demonstrate the efficacy of the proposed Mutual Information-Based Denoising Mechanism, we visualize the denoised trajectory and future predicted trajectory on the ETH dataset. As shown in Figure 4(a), optimizing solely for mutual information leads to the destruction of structural information. However, as depicted in the Figure 4(b), when we incorporate the reconstruction loss $\mathcal{L}_{rec}$, the structure of the trajectory is preserved, and more accurate future trajectory predictions based on these well-structured observations. This underscores the effectiveness of our proposed method.



(a) Results of applying $\mathcal{L}_{MI}$ (b) Results of applying $\mathcal{L}_{MI} + \mathcal{L}_{rec}$

Figure 4: Visualization of trajectories on ETH dataset by employing (a) $\mathcal{L}_{MI}$ and (b) $\mathcal{L}_{MI} + \mathcal{L}_{rec}$. The clean, noisy, and denoised observations are shown in green, blue, and red, respectively. The ground-truth and predicted future trajectories are shown in orange and cyan, respectively.

6.2 More analysis of NoisyTraj

Performance under low/no noise settings. We evaluate NoisyTraj under low or no noise by setting the Gaussian noise σ to 0.05 and 0. The results presented in Table 6 indicate that after integrating NoisyTraj into EqMotion, the performance is still superior to baselines when at a low noise level ($\sigma = 0.05$). Additionally, NoisyTraj+EqMotion performs comparably to EqMotion when $\sigma = 0$. This demonstrates NoisyTraj does not degrade the performance when noise is not introduced.

Table 6: Comparison of different methods under different noise setting on the SDD dataset. The evaluation metrics are ADE and FDE (Unit: pixels). The best results are highlighted in **bold**.

Noise	Method	SDD	
		ADE	FDE
$\sigma = 0.05$	EqMotion	8.48	13.49
	Wavelet+EqMotion	8.39	13.37
	EMA+EqMotion	8.42	13.36
	NoisyTraj+EqMotion	**8.32**	**13.28**

Noise	Method	SDD	
		ADE	FDE
$\sigma = 0$	EqMotion	**8.08**	13.12
	Wavelet+EqMotion	8.16	13.42
	EMA+EqMotion	8.22	13.57
	NoisyTraj+EqMotion	8.11	**13.08**

Table 7: Comparison with baselines using MID backbone. The evaluation metrics are ADE and FDE (Unit: pixels). The best results are highlighted in **bold**.

Noise	Method	SDD	
		ADE	FDE
$\sigma = 0.4$	MID	12.86	18.35
	Wavelet+MID	12.26	17.88
	EMA+MID	12.45	18.01
	NoisyTraj+MID	**11.97**	**17.41**

Performance on diffusion-based backbones. In addition to GraphTern and EqMotion, we integrate NoisyTraj into MID, a diffusion-based model for trajectory prediction. Specifically, we first use

TDM to denoise the noisy observations X_{obs}, obtaining $\hat{X}_{obs}$. Then, using both the denoised and original observations, we sample normal noise from a standard Gaussian distribution to generate $\hat{Y}_{fut}$ and $\tilde{Y}_{fut}$, respectively. To optimize the model, We apply $\mathcal{L}_{pred}$ and $\mathcal{L}_{rank}$ alongside the MID loss . The results shown in Table 7 show that NoisyTraj still outperforms the baselines, which further underscores its adaptability.

Comparison with frozen predictor. We conduct an experiment where we freeze the predictor and only train the denoiser. We first load the predictor trained on clean observations, freeze its parameters, and then integrate NoisyTraj, training only the denoiser. The results, shown in Table 8, reveal a performance decrease when the predictor's parameters are frozen. This indicates the necessity of jointly learning the denoiser and predictor.

Table 8: Comparison with NoiseTraj where the predictor is freezed. The best results are highlighted in **bold**

Noise	Method	SDD	
		ADE	FDE
$\sigma = 0.4$	EqMotion	13.46	19.60
	NoisyTraj+EqMotion (freeze)	12.19	17.95
	NoisyTraj+EqMotion	**11.92**	**17.65**

Comparison with Learning-based baseline. To our knowledge, our work is the first to address trajectory prediction with noisy observations, with no existing learning-based baselines for this problem. We use Noise2Void [1], a learning-based denoiser originally for image denoising, as another baseline. We first denoise the observed trajectories, and then perform future trajectory prediction based on the observations. The results in Table 9 of the attached PDF show that NoisyTraj outperforms Noise2Void, demonstrating the effectiveness of our method.

Table 9: Comparison with baselines on SDD dataset. The evaluation metrics are ADE and FDE (Unit: pixels). The best results are highlighted in **bold**.

Noise	Method	SDD	
		ADE	FDE
$\sigma = 0.4$	EqMotion	13.46	19.60
	Wavelet+EqMotion	12.38	18.25
	EMA+EqMotion	12.79	18.64
	Noise2Void+EqMotion	12.46	18.52
	NoisyTraj+EqMotion	**11.92**	**17.65**

6.3 Proof of Theorem 3.1

Theorem 6.1 (Theorem 3.1 restated)**.** *Given two random variables x and y, the mutual information $I(x;y)$ has the following upper bound*

$$I(x;y) \leq \mathbb{E}_{p(x,y)}[\log p(y|x)] - \mathbb{E}_{p(x)}\mathbb{E}_{p(y)}[\log p(y|x)] \tag{19}$$

Proof. The definition of mutual information between variables x and y is

$$\begin{aligned} I(x;y) &= \mathbb{E}_{p(x,y)}\left[\log \frac{p(x,y)}{p(x)p(y)}\right] = \mathbb{E}_{p(x,y)}\left[\log \frac{p(y|x)}{p(y)}\right] \\ &= \mathbb{E}_{p(x,y)}[\log p(y|x)] - \mathbb{E}_{p(x,y)}[\log p(y)] \\ &= \mathbb{E}_{p(x,y)}[\log p(y|x)] - \mathbb{E}_{p(y)}[\log p(y)] \end{aligned} \tag{20}$$

By the definition of the marginal distribution, we have:

$$p(y) = \int p(y|x)p(x)dx = \mathbb{E}_{p(x)}[p(y|x)]. \tag{21}$$

By substituting Equation (21) to , we have:

$$\begin{aligned} I(x;y) &= \mathbb{E}_{p(x,y)}[\log p(y|x)] - \mathbb{E}_{p(y)}[\log p(y)] \\ &= \mathbb{E}_{p(x,y)}[\log p(y|x)] - \mathbb{E}_{p(y)}[\log \mathbb{E}_{p(x)}[p(y|x)]] \end{aligned} \tag{22}$$

Note that the $\log(\cdot)$ is a concave function, by Jensen's Inequality, we have

$$\begin{aligned} -\mathbb{E}_{p(y)}[\log \mathbb{E}_{p(x)}[p(y|x)]] &\leq -\mathbb{E}_{p(y)}\mathbb{E}_{p(x)}[\log p(y|x)] \\ &= \mathbb{E}_{p(x)}\mathbb{E}_{p(y)}[\log p(y|x)] \end{aligned} \tag{23}$$

By applying this inequality to Equation (22), we obtain:

$$\begin{aligned} I(x;y) &= \mathbb{E}_{p(x,y)}[\log p(y|x)] - \mathbb{E}_{p(y)}[p(y)] \\ &= \mathbb{E}_{p(x,y)}[\log p(y|x)] - \mathbb{E}_{p(y)}[\log \mathbb{E}_{p(x)}[p(y|x)]] \\ &\leq \mathbb{E}_{p(x,y)}[\log p(y|x)] - \mathbb{E}_{p(x)}\mathbb{E}_{p(y)}[\log p(y|x)] \end{aligned} \tag{24}$$

6.4 Proof of Theorem 3.2

Theorem 6.2 (Therorem 3.2 restated)**.** *Given two probability distributions $\mathbb{P}$, $\mathbb{Q}$. The Kullback Liebler Divergence admits the following dual representation:*

$$D_{KL}(\mathbb{P}||\mathbb{Q}) = \sup_{T:\Omega\to\mathbb{R}} \mathbb{E}_{\mathbb{P}}[T] - \log \mathbb{E}_{\mathbb{Q}}[e^T], \tag{25}$$

Proof. The proof comprises two steps. Firstly, we prove the existence of the supremum in the dual representation. Subsequently, we demonstrate that this representation serves as the lower bound of the Kullback-Liebler Divergence.

Lemma 1. *There exist a function $T^* : \Omega \to \mathbb{R}$, such that:*

$$D_{KL}(\mathbb{P}||\mathbb{Q}) = \mathbb{E}_{\mathbb{P}}[T^*] - \log \mathbb{E}_{\mathbb{Q}}[e^{T^*}] \tag{26}$$

Proof. We choose a function $T^* = \log \frac{\mathbb{P}}{\mathbb{Q}}$, then we have:

$$\begin{aligned} \mathbb{E}_{\mathbb{P}}(T^*) - \log \mathbb{E}_{\mathbb{Q}}[e^{T^*}] &= \mathbb{E}_{\mathbb{P}}\left[\log \frac{\mathbb{P}}{\mathbb{Q}}\right] - \log \mathbb{E}_{\mathbb{Q}}[e^{\log \frac{\mathbb{P}}{\mathbb{Q}}}] && (27) \\ &= D_{KL}(\mathbb{P}||\mathbb{Q}) - \log \mathbb{E}_{\mathbb{Q}}\left[\frac{\mathbb{P}}{\mathbb{Q}}\right] && (28) \\ &= D_{KL}(\mathbb{P}||\mathbb{Q}) - \log \int_\Omega \mathbb{Q}\frac{\mathbb{P}}{\mathbb{Q}} d\omega && (29) \\ &= D_{KL}(\mathbb{P}||\mathbb{Q}) - \log \int_\Omega \mathbb{P} d\omega && (30) \\ &= D_{KL}(\mathbb{P}||\mathbb{Q}) - \log 1 && (31) \\ &= D_{KL}(\mathbb{P}||\mathbb{Q}) && (32) \end{aligned}$$

Lemma 2. *For any function $T : \Omega \to \mathbb{R}$, the following equality holds:*

$$D_{KL}(\mathbb{P}||\mathbb{Q}) \geq \mathbb{E}_{\mathbb{P}}[T] - \log \mathbb{E}_{\mathbb{Q}}[e^T] \tag{33}$$

Proof. We define the probability density function $\mathbb{G}$ as:

$$\mathbb{G} \triangleq \frac{\mathbb{Q}e^T}{\mathbb{E}_{\mathbb{Q}}[e^T]} \tag{34}$$

Note that $\mathbb{G}$ satisfies the non-negativity and the integral of its probability density function (PDF) over the input space equals 1:

$$\int_{\Omega} \mathbb{G} d\omega = \int_{\Omega} \frac{\mathbb{Q}e^T}{\mathbb{E}_{\mathbb{Q}}[e^T]} d\omega = \int_{\Omega} \frac{\mathbb{E}_{\mathbb{Q}}[e^T]}{\mathbb{E}_{\mathbb{Q}}[e^T]} d\omega = 1 \tag{35}$$

Then, we calculate the difference between the two sides of 42 to obtain:

$$\begin{aligned} D_{KL}(\mathbb{P}||\mathbb{Q}) - \mathbb{E}_{\mathbb{P}}[T] + \log \mathbb{E}_{\mathbb{Q}}[e^T] &= \mathbb{E}_{\mathbb{P}}\left[\log \frac{\mathbb{P}}{\mathbb{Q}} - T\right] + \log \mathbb{E}_{\mathbb{Q}}[e^T] & (36) \\ &= \mathbb{E}_{\mathbb{P}}\left[\log \frac{\mathbb{P}}{\mathbb{Q}e^T} + \log \mathbb{E}_{\mathbb{Q}}[e^T]\right] & (37) \\ &= \mathbb{E}_{\mathbb{P}}\left[\log \frac{\mathbb{P}\mathbb{E}_{\mathbb{Q}}[e^T]}{\mathbb{Q}e^T}\right] & (38) \\ &= \mathbb{E}_{\mathbb{P}}\left[\log \frac{\mathbb{P}}{\mathbb{G}}\right] & (39) \\ &= D_{KL}(\mathbb{P}||\mathbb{G}) \geq 0 & (40) \end{aligned}$$

Based on the Lemma 1 and Lemma 2, we show that by choosing $T^* = \log \frac{\mathbb{P}}{\mathbb{Q}}$, we obtain:

$$D_{KL}(\mathbb{P}||\mathbb{Q}) = \mathbb{E}_{\mathbb{P}}[T^*] - \log \mathbb{E}_{\mathbb{Q}}[e^{T^*}] \tag{41}$$

Additionally, for any function $T : \Omega \to \mathbb{R}$,

$$D_{KL}(\mathbb{P}||\mathbb{Q}) \geq \mathbb{E}_{\mathbb{P}}[T] - \log \mathbb{E}_{\mathbb{Q}}[e^T] \tag{42}$$

holds. Hence,

$$D_{KL}(\mathbb{P}||\mathbb{Q}) = \sup_{T:\Omega \to \mathbb{R}} \mathbb{E}_{\mathbb{P}}[T] - \log \mathbb{E}_{\mathbb{Q}}[e^T], \tag{43}$$

6.5 Implementation Details

The trajectory denoise model $\Phi_{\mathbf{TDM}}$ is implemented using a 3-layer Transformer with a feature dimension of 256 and the attention head is set to 4. The number of masked locations is set to 2 in our experiments. We empirically set the trade-off parameter β to 0.01 and the margin Δ to 0.05. Additionally, we set the trade-off parameters α, δ, and γ to 0.01, 1 and 0.01, respectively. For the Wavelet denoising method, we utilize the Daubechies wavelet to decompose the signals, and the level is set to 2. We employ the soft-threshold method, with a threshold value set to 0.2. Regarding the EMA method, we empirically determine the Weighted parameter to be 0.75. It is worth noting that these parameter selections are based on experiments aimed at ensuring optimal performance. All experiments are conducted on the PyTorch platform with 4 NVIDIA RTX3090 GPUs.

6.6 Broader Impacts

This work addresses the challenge of trajectory prediction based on noisy observations. It enhances robustness against noise in the trajectory prediction task, benefiting various applications including autonomous driving, robotic navigation, and surveillance systems, thereby contributing to safer deployment.

6.7 Training algorithm of NoisyTraj

We provide the training algorithm of NoisyTraj in the Algorithm 1.

6.8 Discusssion and Limitations

In this paper, we simplify the problem by assuming that only the observed trajectory is noisy, which is a reasonable assumption in certain scenarios. For example, when using an autonomous vehicle equipped with both cameras and LiDAR, we can treat camera-derived trajectories as noisy data and

Algorithm 1: Training Procedure of NoisyTraj

```
Input: Noisy observations $X_{obs}$, ground-truth future trajectories $Y_{fut}$. Four trade-off
        hyper-parameters: $\alpha$, $\beta$, $\delta$ and $\gamma$.
Output: Network parameters: $\Phi_{\textbf{TDM}}$, $\Phi_{\textbf{TPB}}$, $\psi$, and $\phi$.
Initialize: Randomly initialize $\Phi_{\textbf{TDM}}$, $\Phi_{\textbf{TPB}}$, $\psi$, and $\phi$.
while Model not converges do
    Random mask the noisy observations using the mask vector: $X^{mask}_{obs} = X_{obs} \odot \mathcal{M}_{obs}$
    Obtain the trajectories $\hat{X}^{mask}_{obs} = \Phi_{\textbf{TDM}}(X^{mask}_{obs})$
    Calculate reconstruction loss $\mathcal{L}_{rec} = ||\hat{X}^{mask}_{obs} \odot (1 - \mathcal{M}_{obs}) - X_{obs} \odot (1 - \mathcal{M}_{obs})||_2$
    Input noisy observations to $\Phi_{\textbf{TDM}}$ for denoising: $\hat{X}_{obs} = \Phi_{\textbf{TDM}}(X_{obs})$
    Employ Mutual Information-based mechanism for further denoising:
      $\mathcal{L}_{MI} = \alpha \mathbb{E}_{p(X_{obs}, \hat{X}_{obs})}[\log q_\phi(\hat{X}_{obs}|X_{obs})] - \mathbb{E}_{p(X_{obs})}\mathbb{E}_{p(\hat{X}_{obs})}[\log q_\phi(\hat{X}_{obs}|X_{obs})]$
          $- \sup_{\psi} \mathbb{E}_{p(\hat{X}_{obs}, Y_{fut})}[T_\psi] + \log \mathbb{E}_{p(\hat{X}_{obs})p(Y_{fut})}[e^{T_\psi}]$
    Obtain the future predictions based on denoised observations: $\{\hat{Y}^k_{fut}\}^K_{k=1} = \Phi_{\textbf{TPB}}(\hat{X}_{obs})$
    Obtain the future predictions based on noisy observation: $\{\tilde{Y}^k_{fut}\}^K_{k=1} = \Phi_{\textbf{TPB}}(X_{obs})$
    Calculate $d_{denoise}$ and $d_{noise}$:
      $d_{denoise} = \min_{1 \leq k \leq K} ||\hat{Y}^k_{fut} - Y_{fut}||_2, \;\; d_{noise} = \min_{1 \leq k \leq K} ||\tilde{Y}^k_{fut} - Y_{fut}||_2$
    Calculate $\mathcal{L}_{pred}$ and $\mathcal{L}_{rank}$ as
      $\mathcal{L}_{pred} = ||\hat{Y}^{best}_{fut} - Y_{fut}||_2 + ||\tilde{Y}^{best}_{fut} - Y_{fut}||, \;\; \mathcal{L}_{rank} = \max(0, d_{denoise} - d_{noise} + \Delta)$
    Optimizing $\mathcal{L} = \mathcal{L}_{pred} + \beta\mathcal{L}_{rank} + \delta\mathcal{L}_{rec} + \gamma\mathcal{L}_{MI}$ by gradient descent to update the $\Phi_{\textbf{TDM}}$
      and $\Phi_{\textbf{TPB}}$.
end
```

LiDAR-derived trajectories as clean ground-truth for training. Once the model is trained on this data, it can be deployed on a vehicle equipped with only cameras. This camera-only approach is adopted by top industry Tesla to design the Autopilot system, which has been successfully deployed in real-world scenarios [2].

While this work focuses on addressing trajectory prediction based on noisy observed trajectories, it is important to acknowledge that the collected future ground-truth trajectories may also be contaminated with noise. In such cases, the proposed mutual information-based denoising mechanism may not be effective, as NoisyTraj assumes the future trajectories are noise-free and uses them as additional information for denoising the observations. Future research could explore methods for predicting future trajectories based on both noisy observations and noisy future ground-truths.

Reference

[1] Krull, Alexander, Tim-Oliver Buchholz, and Florian Jug. Noise2void-learning denoising from single noisy images. *In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp.2129-2137, 2019.

[2] Tesla AI Day 2021, August 19, 2021, 3:03:20. https://www.youtube.com/watch?v=j0z4FweCy4M.