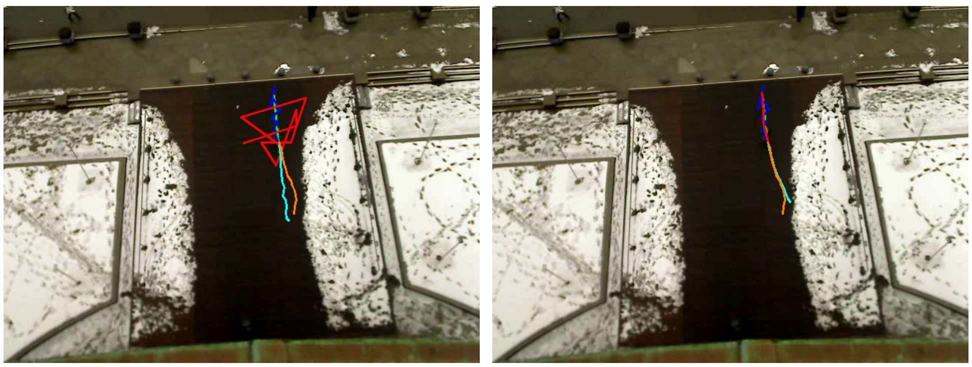
6 Appendix

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:

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

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:

$$D_{KL}(\mathbb{P}||\mathbb{Q}) - \mathbb{E}_\mathbb{P}[T] + \log \mathbb{E}_\mathbb{Q}[e^T] = \mathbb{E}_\mathbb{P}\left[\log \frac{\mathbb{P}}{\mathbb{Q}} - T\right] + \log \mathbb{E}_\mathbb{Q}[e^T] \tag{36}$$

$$= \mathbb{E}_\mathbb{P}\left[\log \frac{\mathbb{P}}{\mathbb{Q}e^T} + \log \mathbb{E}_\mathbb{Q}[e^T]\right] \tag{37}$$

$$= \mathbb{E}_\mathbb{P}\left[\log \frac{\mathbb{P}\mathbb{E}_\mathbb{Q}[e^T]}{\mathbb{Q}e^T}\right] \tag{38}$$

$$= \mathbb{E}_\mathbb{P}\left[\log \frac{\mathbb{P}}{\mathbb{G}}\right] \tag{39}$$

$$= D_{KL}(\mathbb{P}||\mathbb{G}) \geq 0 \tag{40}$$

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: α, β, δ and γ.
Output: Network parameters: $\Phi_{\mathbf{TDM}}$, $\Phi_{\mathbf{TPB}}$, ψ, and ϕ.
Initialize: Randomly initialize $\Phi_{\mathbf{TDM}}$, $\Phi_{\mathbf{TPB}}$, ψ, and ϕ.
while *Model not converges* **do**
  Random mask the noisy observations using the mask vector: $X_{obs}^{mask} = X_{obs} \odot \mathcal{M}_{obs}$
  Obtain the trajectories $\hat{X}_{obs}^{mask} = \Phi_{\mathbf{TDM}}(X_{obs}^{mask})$
  Calculate reconstruction loss $\mathcal{L}_{rec} = ||\hat{X}_{obs}^{mask} \odot (1 - \mathcal{M}_{obs}) - X_{obs} \odot (1 - \mathcal{M}_{obs})||_2$
  Input noisy observations to $\Phi_{\mathbf{TDM}}$ for denoising: $\hat{X}_{obs} = \Phi_{\mathbf{TDM}}(X_{obs})$
  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: $\{\hat{Y}_{fut}^k\}_{k=1}^K = \Phi_{\mathbf{TPB}}(\hat{X}_{obs})$
  Obtain the future predictions based on noisy observation: $\{\tilde{Y}_{fut}^k\}_{k=1}^K = \Phi_{\mathbf{TPB}}(X_{obs})$
  Calculate $d_{denoise}$ and d_{noise}:
  $$d_{denoise} = \min_{1\le k\le K} ||\hat{Y}_{fut}^k - Y_{fut}||_2, \quad d_{noise} = \min_{1\le k\le K} ||\tilde{Y}_{fut}^k - Y_{fut}||_2$$
  Calculate $\mathcal{L}_{pred}$ and $\mathcal{L}_{rank}$ as
  $$\mathcal{L}_{pred} = ||\hat{Y}_{fut}^{best} - Y_{fut}||_2 + ||\tilde{Y}_{fut}^{best} - Y_{fut}||, \quad \mathcal{L}_{rank} = \max(0, d_{denoise} - d_{noise} + \Delta)$$
  Optimizing $\mathcal{L} = \mathcal{L}_{pred} + \beta\mathcal{L}_{rank} + \delta\mathcal{L}_{rec} + \gamma\mathcal{L}_{MI}$ by gradient descent to update the $\Phi_{\mathbf{TDM}}$ and $\Phi_{\mathbf{TPB}}$.
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.