2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Joint Pedestrian Trajectory Prediction through Posterior Sampling

Haotian Lin[1*,2], Yixiao Wang[1†], Mingxiao Huo[3], Chensheng Peng[1], Zhiyuan Liu[2], Masayoshi Tomizuka[1]

***Abstract*— Joint pedestrian trajectory prediction has long grappled with the inherent unpredictability of human behaviors. Recent works employing conditional diffusion models in trajectory prediction have exhibited notable success. Nevertheless, the heavy dependence on accurate historical data results in their vulnerability to noise disturbances and data incompleteness. To improve the robustness and reliability, we introduce the Guided Full Trajectory Diffuser (GFTD), a novel diffusion-based framework that translates prediction as the inverse problem of spatial-temporal inpainting and models the full joint trajectory distribution which includes both history and the future. By learning from the full trajectory and leveraging flexible posterior sampling methods, GFTD can produce accurate predictions while improving the robustness that can generalize to scenarios with noise perturbation or incomplete historical data. Moreover, the pre-trained model enables controllable generation without an additional training budget. Through rigorous experimental evaluation, GFTD exhibits superior performance in joint trajectory prediction with different data quality and in controllable generation tasks. See more results at https://sites.google.com/andrew.cmu.edu/posterior-sampling-prediction.**

I. INTRODUCTION

Pedestrian trajectory prediction is crucial for human-robot interaction systems such as autonomous driving, etc. The goal is to predict future trajectories based on previous pedestrian movements and environmental contexts. By accurately predicting pedestrian trajectories, autonomous systems can plan their actions accordingly, ensuring safe and efficient navigation in dynamic environments. However, the unpredictable and complicated nature of human behaviors makes this task challenging, especially with multiple agents involved where their interactions need to be considered.

To simplify the problem, early works [1] [2] [3] [4] focused on marginal pedestrian trajectory prediction, which forecasts the trajectory for each pedestrian independently. Such approaches require a downstream planning module to perform safety checks for every combination of the individual predictions. Even so, combination rollouts could still produce unrealistic self-collisions and lead to failure in challenging scenarios. As a result, joint pedestrian trajectory prediction, which predicts consistent trajectories for all agents together, has gained attention in the community. [5] introduced a joint metrics term into supervision loss, transforming the marginal predictor into a joint one. However, joint pedestrian trajectory prediction remains challenging since existing approaches heavily rely on accurate and complete historical data to incorporate temporal and social dynamics. This dependence results in their vulnerability to interference from noisy disturbances and incomplete data, significantly threatening their effectiveness in real-world applications, such as sensors under adverse weather conditions.

To deal with the noises, previous research involved augmenting data with predefined noise and training models on noisy datasets [6], and it can be further improved by adversarial training procedures [7]. In addition, historical trajectories from sensors could be incomplete. Previous studies [8] [9] [10] [11] proposed to reconstruct incomplete data and predict future trajectories during the training phase, requiring a specifically designed model and well-established training strategy. Such approaches optimized per-problem functions, lacking adaptability to diverse tasks across various contexts.

Drawing inspiration from the diffusion model with its remarkable capability of capturing the complicated distribution, we propose a unified framework, named, **Guided Full Trajectory Diffuser** (**GFTD**) for the joint pedestrian trajectory prediction to better handle the disturbances and incompleteness. GFTD represents the entire trajectory distribution, both historical and future, with one diffusion model. We formulate trajectory prediction and controllable generation as inverse problems and solve them through posterior sampling techniques. Specifically, we sample in-distribution full trajectories based on historical trajectories and priors such as physical constraints and behavioral intentions. Without the necessity for explicit training to handle noisy and/or incomplete inputs, GFTD enables robust prediction and controllable generation—both achievable during the inference time. In a nutshell, our proposed framework streamlines the training process and offers adaptability to various scenarios at inference, providing a solution that can address all challenges without extra training requirements.

Our contributions can be summarized as follows:

(1) We introduce a novel permutation-invariant framework for representing the joint distribution of full trajectories (both historical and future), converting trajectory prediction and controllable generation into a unified inverse problem.

(2) We utilize posterior sampling to solve the formulated problem. With our approach, there is no need for specific treatments during the training phase, as it can generalize to various types of data imperfections solely at inference time with one trained model.

(3) Extensive experiments demonstrate that our model not only performs comparably in joint trajectory prediction but also excels in controllable generation, particularly in scenarios with noise injection and incomplete historical data.

[1] University of California, Berkeley, [2] Tsinghua University, [3] Carnegie Mellon University
[*] Research performed while visiting University of California, Berkeley.
[†] Corresponding author: yixiao_wang@berkeley.edu

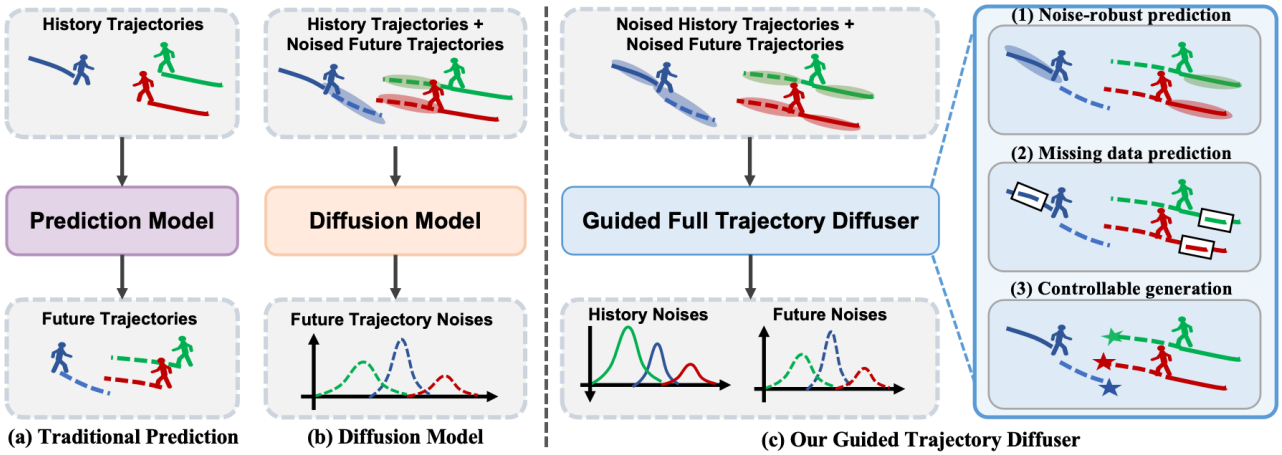

Fig. 1: Illustration of existing trajectory prediction framework and our guided full trajectory diffuser framework. (a) Multi-agent trajectory prediction methods directly generate entire future trajectories through supervised learning. (b) Diffusion-based Multi-agent trajectory prediction methods generate future trajectories step-by-step during the denoising process. (c) Our method for multi-agent trajectory prediction incorporates history guidance within the diffusion framework. It predicts entire trajectories and supports additional capabilities in a single model, including Noise-Robust Trajectory Prediction, Incomplete Data Prediction, and Controllable Trajectory Generation.

II. Related Work

A. Pedestrian Trajectory Prediction

Pedestrian trajectory prediction is crucial for many downstream tasks in autonomous driving, such as tracking [12] and mapping [13]. This task involves forecasting the future movement paths of pedestrians, given their historical movements and the environment. However, it is challenging to predict their motion because of the diverse and unpredictable nature of human behaviors. To address this issue, two mainstream paradigms have been developed. The supervised learning approach [4] aims to minimize the differences between the ground truth trajectories and the predictions using L2 loss, etc. On the other hand, the generative learning approach [1] formulates the prediction as a task of generating conditional distributions of the future trajectory based on past trajectories. Among various generative models, the diffusion model [14] [15] has shown exceptional performance in pedestrian trajectory prediction. Through a conditional reverse diffusion process, the diffusion model generates future trajectory distributions from a standard Gaussian distribution, which captures accurate and diverse predictions of future trajectories.

B. Joint Trajectory Distribution Modelling

Recently, joint pedestrian trajectory prediction has gained significant attention. Unlike marginal trajectory prediction, which treats each pedestrian independently and can result in self-colliding trajectories between agents, joint trajectory prediction considers the interactions between future trajectories of agents, leading to more consistent and feasible predictions. [5] incorporates joint metrics as the training objective, transforming the marginal trajectory predictor into a joint trajectory predictor. In this paper, we propose Guided Full Trajectory Diffuser (GTFD) for joint pedestrian trajectory prediction, leveraging the strengths of diffusion models in generating accurate and diverse future trajectory distributions while considering the interactions between pedestrians.

C. Posterior Sampling for Inverse Problems

Posterior sampling is to infer the underlying distribution conditioned on the measurements, given the unconditional distribution. Previous works [16] [17] [18] have demonstrated great performance in general inverse problems of computer vision, such as image inpainting and denoising. Recent work involves physical constraints [19], the reward in reinforcement learning [20], behavior preference in traffic simulation [21] [22] and bias in trajectory prediction [23], extending a wide range of applications of posterior sampling. In this paper, we gain the insight that, in the task of trajectory prediction, historical trajectories can be regarded as observation or measurement. As a result, we can unleash the potential of posterior sampling in dealing with prediction under uncertainty.

III. Methods

In this section, we undertake a comprehensive exposition of our novel multi-agent trajectory prediction framework, conceptualizing the prediction task through spatial-temporal inpainting. We first introduce preliminaries on diffusion models and problem definition. Subsequently, we explain how we formulate our Guided Full Trajectory Diffuser (GFTD).

A. Preliminaries on Diffusion models

Diffusion models are a class of generative models that operate on the principle of stochastic processes. They define a forward diffusion process that corrupts data by progressively introducing Gaussian noise. Conversely, in the reverse process, diffusion models reconstruct data from noise by iteratively reducing the Gaussian noise.

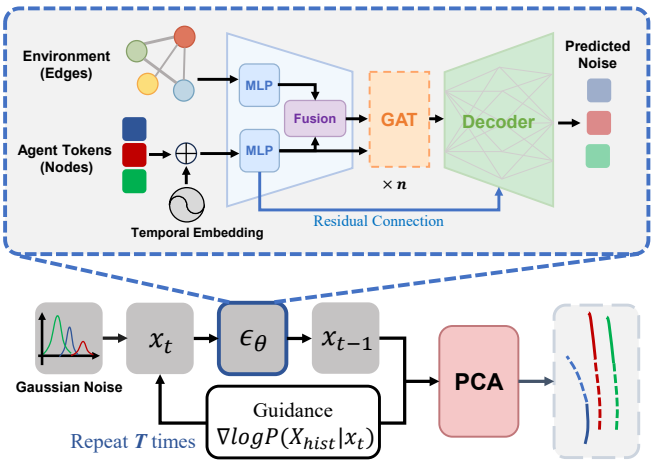

Fig. 2: Architecture of our proposed GFTD framework. During inference, GFTD samples from Gaussian noise, and iteratively recover data with the denoise module ε_θ. It takes in noisy latent node features x_t^i and edge features e. After MLP encoding, the edge is augmented by concatenating the encoded node features and they are both sent into the Processor which consists of stacked Graph Attention (GAT) layers. We then map the nodes to the same dimensions as x_t and add residual connection, resulting in the predicted intermediate noise ε_t. After each denoise step, we address conditions through posterior sampling. Finally, the denoised latent nodes x_0 are converted to trajectory space.

According to Itô Stochastic Differential Equations (SDE) [24], the forward data noising process is defined as the following form:

$$dx = f(x,t)dt + g(t)dw. \tag{1}$$

where x represents the data, w is the standard Wiener process, f is the drift coefficient, g is the diffusion coefficient, $t \in [0,T]$ is the diffusion step.

The corresponding reverse SDE of Eq. (1) is defined as

$$dx = [f(x,t) - g^2(t)\nabla_x \log p_t(x)]dt + g(t)dw. \tag{2}$$

where $p_t(x)$ is the probability density, and $\nabla_x \log p_t(x)$ is the score function, which can be learned by a neural network $s_\theta(x_t,t)$ with score matching. Here, x_t denotes the intermediate noisy data x at denoise time t. The training objective is defined as

$$\mathbb{E}_t\{\lambda(t)\mathbb{E}_{x_0}\mathbb{E}_{x_t|x_0}[\|s_\theta(x_t,t) - \nabla_{x_t}\log p(x_t|x_0)\|_2^2]\}. \tag{3}$$

where $\lambda(t)$ is a weighting function.

In this paper, we follow the implementation introduced by Denoising Diffusion Probabilistic Models (DDPM) [25]. DDPM gradually add noise to the original data x according to a variance schedule β_1, β_2,...,β_T:

$$x_t = \sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta_t}\varepsilon, \quad t \in \{1,2,...,T\} \tag{4}$$

where $\varepsilon \sim \mathcal{N}(0,I)$, $x_0 = x$, $x_T \sim \mathcal{N}(0,I)$. During the training, DDPM tends to learn the added noise,

$$\mathbb{E}_{\varepsilon,x_0,t}\|\varepsilon - \varepsilon_\theta(x_t,t)\|^2. \tag{5}$$

With the learned $\varepsilon_\theta(x_t,t)$, the reverse diffusion process gradually denoises standard Gaussian noise into the original data distribution by

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{\beta_t}{\sqrt{1-\overline{\alpha}_t}}\varepsilon_\theta(x_t,t)) + \sqrt{\beta_t}\varepsilon. \tag{6}$$

where $x_T \sim \mathcal{N}(0,I)$, t from T to 1.

B. Problem Formulation

The objective of the multi-agent trajectory prediction task is to use the observed history trajectories to jointly predict future trajectories of all the agents in the scene. We denote P_k^i as the positions of agent i at time step k, $i = 1,\ldots,N_a$. Joint history trajectory is $c = \{c^i\}_{i=1}^{N_a}$, $c^i = \{P^i_{-T_{his}+1}, P^i_{-T_{his}+2}, \ldots P^i_{-1}\}$. Joint current position is $E = \{E^i\}_{i=1}^{N_a}$, $E^i = P_0^i$. Joint future trajectory is $y = \{y^i\}_{i=1}^{N_a}$, $y^i = \{P_1^i,\ldots,P^i_{T_{fut}}\}$. The task is to generate the future trajectory distribution $p(y|E,c)$. However, the historical trajectory may be incomplete and injected with noise caused by unexpected conditions such as sensor failure and poor weather. We use an elementwise mask to represent incomplete trajectory data and model the injected noise as Gaussian noise.

C. Represent Full Trajectory Distribution with Diffusion

Previous diffusion-based work [14] [23] [26] formulated prediction task as the diffusion process that conditioned on history trajectories c and current position E to generate future trajectory distribution $p(y|E,c)$. Our key insight here is that we can learn to generate full-length trajectories containing both the past and the future and regard the prediction task as an inverse problem of inpainting, which aims to infer and complete the incomplete trajectory (e.g., the future) based on observations (e.g., the past). We denote the generated history trajectory as $\hat{c}_i$ and the corresponding full-length trajectory as $x^i = \hat{c}^i \cup y^i = \{P^i_{-T_{his}},\ldots,P^i_{-1},P^i_1,\ldots,P^i_{T_{fut}}\}$. We train the DDPM by Eq. (5) where $x = \{x^i\}_{i=1}^{N_a}$, and we can generate full trajectory distribution $p(x|E)$ by learned $\varepsilon_\theta(x_t,t,E)$ through Eq. (6).

D. Robust Prediction as Posterior Sampling

Prediction task is to obtain $p(y|E,c)$ which is equivalent to $p(y,c|E,c) = p(x|E,c)$. Given learned full trajectory $p(x|E)$ through diffusion model and Bayes' rule, we have $p(x|E,c) = p(c|x,E)p(x|E)/p(c)$. Thus, the score of the posterior distribution can be calculated as:

$$\nabla_x \log p_t(x|E,c) = \nabla_x \log p_t(x|E) + \nabla_x \log p_t(c|x,E). \tag{7}$$

According to diffusion posterior sampling (DPS) [17], if the condition or measurement has the form of a general noisy inverse problem $c = \phi(x_0) + n$, where ϕ is an arbitrary operator, and $n \sim \mathcal{N}(0,\sigma^2 I)$ is the Gaussian noise, then $\nabla_x \log p_t(c|x,E)$ can be approximated as

$$\nabla_x \log p_t(c|x,E) = -\lambda\nabla_{x_t}\|c - \phi(\hat{x_0}(x_t))\|_2^2 = -\lambda\nabla_{x_t}\mathscr{L}, \tag{8}$$

where $\hat{x_0}(x_t)$ is the posterior mean of $p(x_0|x_t)$:

$$\hat{x_0}(x_t) = \frac{1}{\sqrt{\overline{\alpha}_t}}(x_t + (1-\overline{\alpha}_t)\varepsilon_\theta(x_t,t,E)). \tag{9}$$

Once the condition operator ϕ is known (i.e., $\phi_{his}(x)$ denotes historical portion of x), we can inject such condition by iteratively adding the guidance term Eq. (8) to the intermediate noisy data during the reverse diffusion process so that we drag the sample to the direction that minimizes the guidance loss $\mathscr{L}$ that reflects our preferences, which would be having the generated history trajectories close to the ground truth observation. Specific designs of $\mathscr{L}$ will be elaborated in section IV. With the guidance gradient restricted to a certain range, the generated noisy samples will be lying in the data manifolds, thus avoiding generating out-of-distribution samples. Even though this does not guarantee the exact recovery of history trajectories and may result in some degrees of ill-conditioned prediction, a properly guided sample still achieves high prediction accuracy. We refer to this characteristic as "soft-conditioning" and will show such a design greatly boosts robustness and adaptation abilities in the following sections.

For conventional prediction tasks, we expect data to be clean and complete and do not need to trade condition correctness off for perturbation tolerance. To further enhance performance on these tasks, we can strictly enforce history conditions following RePaint[27]. This plug-and-play modification can be manually enabled. It introduces an additional step that first compromises observed history trajectories to the noise level t through the forward diffusion process Eq. (4), then concatenates them with the corresponding future parts of intermediate samples.

An implementation of the inference process in our proposed framework is summarized in Algorithm 1.

Algorithm 1 Guided Full Trajectory Diffuser (GFTD)

Input: $\mathscr{L}(\cdot,\cdot)$, $\phi(\cdot)$, c, $\{\bar{\alpha}_t,\beta_t,\lambda_t\}_{t=0}^{T-1}$,

$x_T \sim \mathcal{N}(0,I)$
for $t = T-1,...,1$ **do**
  $\varepsilon \sim \mathcal{N}(0,I)$
  $x_{t-1} = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\varepsilon_\theta(x_t,t,E)) + \sqrt{\beta_t}\varepsilon$
  $\hat{x}_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}(x_t - \sqrt{1-\bar{\alpha}_t}\varepsilon_\theta(x_t,t,E))$
  $g = -\nabla_{x_t}\mathscr{L}(\hat{x}_0,c)$
  $x_{t-1} = x_{t-1} + \lambda_t g$
  if RePaint **and** $t > 0$ **then**
    $\varepsilon' \sim \mathcal{N}(0,I)$
    $c_{t-1} = \sqrt{1-\beta_{t-1}}c_0 + \sqrt{\beta}_{t-1}\varepsilon$
    $\phi_{his}(x_{t-1}) = c_{t-1}$
  end if
end for
$y = \phi_{fut}(x_0)$

Output: y

E. Trajectory Latent Representation

Inspired by the advances in image synthesis bought by latent diffusion models [28] and the successful application of low-rank latent representation in trajectory prediction [29] [23], we further consider transforming the spatial-temporal features of trajectory $X^i \in \mathbb{R}^{T*d}$ into latent features $x^i \in \mathbb{R}^k$, $k \ll T*d$ through Principal Component Analysis (PCA). This notation stands for the linear combination of the first principal components. This simple but efficient transformation serves to mitigate the effects of noisy data, leading to more consistent and smoother predicted trajectories. Meanwhile, the latent representation contains its geometry and temporal characteristics, saving the need for additional temporal encoder blocks.

IV. Framework Architecture

As we re-formulate the prediction task as trajectory inpainting, our proposed framework requires a pre-trained DDPM that models the joint distribution of full-length trajectories that is not conditioned on history motion c. Through iteratively adding posterior guidance $-\lambda\nabla_{x_t}\mathscr{L}$ during inference time and extracting the future parts y from generated trajectories x, we achieve the goal to sample future trajectories conditioned on given histories. We argue that our framework is model-agnostic that various reasonable network designs of the denoising module are feasible for our methodology. This section presents a specific implementation of a lightweight denoising module. We will discuss our data representation method, module architecture, and guidance design in the following.

Data Representation. To achieve a rotation-invariant representation of agent motion, for each agent in the scene, we normalized their trajectory according to their current positions and headings. Parallelly, we extract the spatial relativity of agents under current frame by establishing a graph representation, where node $x^i, i = 1,\ldots,N_a$ stands for agent motion and edge $E^{ij}, i,j = 1,\ldots,N_a$ is characterized by a 6D vector describing their relative position at the current timestep: relative distance d^{ij}, direction vector r^{ij} under the reference frame of agent i, and relative heading vector $h^{ij} = \{\theta^{ij}, cos\theta^{ij}, sin\theta^{ij}\}$.

Denoise Module. Following the standard setting in DDPM, we built a denoise module $\varepsilon_\theta(x_t,t,E)$ that predicts the intermediate noise level and denoise the intermediate sample x_t by Eq. (6). For each scenario sample, $x_t = [PCA(X_t^1),\ldots,PCA(X_t^{N_a})]$ is the latent node features of agent trajectories at diffusion step t. And E stands for the edge information mentioned above, it serves as the only condition of the diffusion model and is shared across all diffusion steps.

The core to ε_θ is a Graph Neural Network (GNN). As depicted in Fig. 2, each latent node feature x_t^i is firstly concatenated with diffusion time embeddings and then mapped to higher dimensions by a multi-layer perception (MLP). Since node features do not contain global position information, we need to further fuse the nodes with edge features so that we can model the complex interaction. After MLP encoding, the edge is augmented by concatenating with the encoded node

$$E_t^{ij} = concat\{E^{ij}, x_t^i\}, \tag{10}$$

and they are both sent into the Processor which consists of stacked Graph Attention (GAT) [30] layers. Within each GAT layer, nodes are augmented by shared linear transformation

parameterized by weight $\boldsymbol{W}$. Then, we calculate the attention score by

$$\alpha_t^{ij} = softmax_j(e_t^{ij}) = \frac{exp(e_t^{ij})}{\sum_{k\in N_a} exp(e_t^{ik})}, \tag{11}$$

and

$$e_t^{ij} = LekyReLU(a(\boldsymbol{W}n^i || \boldsymbol{W}n^j)), \tag{12}$$

where $a(\cdot)$ linear transformation that maps a vector to a real number. After acquiring interaction information, we map the nodes to the same dimensions as x_t, resulting in the predicted intermediate noise ε_t.

Guided Generation. For trajectory prediction under the description of the noisy inverse problem, the closed-form dependency between the measurement and sample can be formulated as

$$c = \phi(X) + n. \tag{13}$$

For the prediction task, c is the observed history trajectories, ϕ is the operator that extracts history parts of trajectories from full-length trajectories X, and n is the Gaussian noise with adjustable variance based on how clear our history data is. For the original prediction task, we consider the history trajectories to be reliable and the noise level to be low. Similar to the derivation in section C, we set our guidance loss as

$$\mathscr{L}_{rec} = \left\| c - \phi(\hat{X}_0(x_t)) \right\|_2, \tag{14}$$

which is the reconstruction loss that measures how precisely our generated trajectories fit the conditions. Inspired by [23], we can ensure a more realistic generation by adding repeller guidance that prevents the agents from colliding with each other:

$$\mathscr{L}_{rep} = \frac{1}{N_a} \sum_{i,j,k} \max\{(1 - \frac{1}{r} d_k^{ij}), 0\}, \tag{15}$$

where r is the repeller threshold and $d_k^{ij} \in \mathbb{R}^{N_a \times N_a \times (T_{hist}+T_{fut})}$ is the distance between agent i and j at timestep t. Practically, the repeller loss serves as valuable prior knowledge.

V. Experiments

In this section, we exhibit the capacity and flexibility of our framework by adapting our model to four distinct tasks without retraining: basic trajectory prediction, controllable generation, prediction with noisy history, and prediction with incomplete history. We present two versions of our proposed framework: the foundational **GFTD** and its variant, **GFTD-RePaint**, which is specialized for the basic trajectory prediction task.

A. Experimental Setups

Datasets. We carried out all of our experiments on the ETH/UCY [34] [35] dataset, a popular public pedestrian trajectories forecasting benchmark. The dataset contains pedestrian trajectories in bird's eye view (BEV) from five distinct scenarios (ETH, Hotel, Univ, Zara1, Zara2). We follow the leave-one-out training/evaluation setup that was used in the original S-GAN [31]. Specifically, we partitioned each dataset scene into sliding windows of length 20 steps (8 seconds) at stride 1, with 8 history observation steps (3.2 seconds) followed by 12 prediction steps (4.8 seconds). We then proceed to utilize all sequences containing at least one pedestrian.

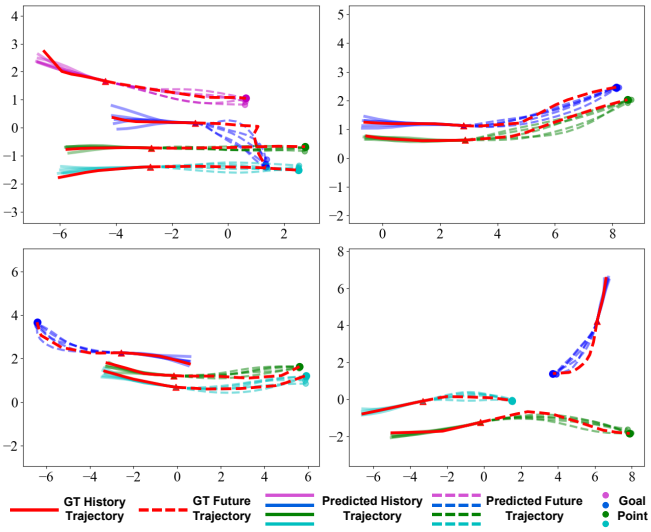

Fig. 3: Qualitative visualization of controllable generation. Red lines represent the ground truth trajectories. With the given goal point and history, our model can generate goal-oriented trajectories with considerable realism.

Metrics and Baselines. For scene-level joint prediction, rather than using marginal metrics Average Displacement Error (ADE) and Final Displacement Error (FDE), we follow [5] to use joint metrics JADE/JFDE:

$$jointADE(Y, Y^*) = \frac{1}{TN} \min_{k=1}^{K} \sum_{n=1}^{N} \sum_{t=1}^{T} \left\| s_{t,n} - s_{t,n}^* \right\|, \tag{16}$$

$$jointFDE(Y, Y^*) = \frac{1}{TN} \min_{k=1}^{K} \sum_{n=1}^{N} \left\| s_{T,n} - s_{T,n}^* \right\|. \tag{17}$$

The joint metrics emphasize the significance of modeling the interactions between agents by introducing a constraint for top-K evaluations: predictions must originate from the same sample. This approach prevents the overestimation of model performance and penalizes models that fail to accurately simulate the realistic behaviors and motions of all agents present in the scene. In the general prediction task, we compared our model to the state-of-the-art models Joint AgentFormer and Joint View Vertically proposed in [5], as well as other famous baselines in pedestrian prediction: S-GAN [31], PECNet[36], and MemoNet [33]. We compute the best-of-20 JADE/JFDE considering the stochasticity of generative prediction models.

Implementation Details. The implementation of the denoising module used in experiments contains a three-layer GAT with a hidden size of 128. Additionally, each feed-forward network is realized as a two-layer MLP equipped with post-layer normalization and activated using the Mish

TABLE I: Trajectory Prediction Performance on ETH/UCY Dataset. The three best scores are marked by red, blue, and green, respectively. † denotes Joint AgentFormer without the diverse sampler (DLow)

Method	minJADE / JFDE (m), K=20 ↓					
	ETH(1.4)	HOTEL(2.7)	UNIV(25.7)	ZARA1(3.3)	ZARA2(5.9)	ETH/UCY Avg.
S-GAN [31]	0.919 / 1.742	0.480 / 0.950	0.744 / 1.573	0.438 / 1.001	0.362 / 0.794	0.589 / 1.212
PECNet [32]	0.618 / 1.097	0.291 / 0.587	0.666 / 1.417	0.408 / 0.896	0.372 / 0.840	0.471 / 0.967
MemoNet[33]	0.499 / 0.859	0.222 / 0.416	0.686 / 1.466	0.349 / 0.723	0.385 / 0.864	0.428 / 0.866
Joint View Vertically [5]	0.652 / 0.839	0.186 / 0.309	0.523 / 1.091	0.331 / 0.634	0.267 / 0.547	0.392 / 0.684
Joint AgentFormer† [5]	0.543 / 0.883	0.211 / 0.377	0.596 / 1.247	0.309 / 0.612	0.282 / 0.584	0.388 / 0.741
Ours (GFTD)	0.505 / 0.873	0.174 / 0.297	0.649 / 1.305	0.340 / 0.667	0.308 / 0.620	0.395 / 0.752
Ours (GFTD + RePaint)	0.514 / 0.906	0.191 / 0.329	0.607 / 1.248	0.327 / 0.646	0.288 / 0.585	0.385 / 0.743

TABLE II: Controllable Generation Performance

Method	minJADE / JFDE (m), K=20 ↓					
	ETH(1.4)	HOTEL(2.7)	UNIV(25.7)	ZARA1(3.3)	ZARA2(5.9)	ETH/UCY Avg.
Baseline	0.505 / 0.873	0.174 / 0.297	0.649 / 1.305	0.340 / 0.667	0.308 / 0.620	0.395 / 0.752
Goal Point Guidance	0.224 / 0.064	0.069 / 0.033	0.280 / 0.394	0.103 / 0.035	0.094 / 0.032	0.154 / 0.112

activation function [37]. In addition, we observed that training our diffusion model on latent space by Eq. (5) can be unstable and somehow inefficient. Thus, following [38], we modified the DDPM training loss with the Min-SNR-γ weighting strategy $\lambda(t) = min\{\gamma, SRN(t)\}/SRN(t)$ to avoid having the model putting too much attention on the final steps of the denoising processing where the noise level is low. We set the training batch size to 32 and we use the Adam optimizer with an initial learning rate of 0.001. All the training is conducted on one GTX-2080Ti GPU.

B. Trajectory Prediction and Controllable Generation

In the context of basic pedestrian trajectory prediction tasks, we present two models within our Guided Trajectory Diffusion (GFTD) framework: the GFTD model and its augmented version, GFTD with RePaint. To assess their efficacy, we conduct a quantitative evaluation, benchmarking these models against the aforementioned baselines with the ETH/UCY dataset. All models are fairly evaluated by joint metrics to ascertain their optimal scene-level performance across 20 samplings. We directly used the reported baseline performance under joint metrics from [5]. We present the results in TABLE I. In specific for Jonit AgentFormer, we use the pre-trained checkpoint for joint prediction provided by their official GitHub repository (https://github.com/ericaweng/joint-metrics-matter). For a fair comparison, we did not apply the Dlow model for performance enhancement. Our model exhibited relative competitive performance, especially on the HOTEL dataset, where our model outperformed all the baselines.

Controllable trajectory generation is another plausible application of our framework. In addition to the fundamental reconstruction loss Eq. (14), we can incorporate any desirable objectives into the guidance term to control the sampling process. For demonstration, we experimented with goal point generation, i.e., given the ground truth history trajectories and desired goal points g, we asked our model to generate future trajectories that reach the goal points. The only modification to adapt our pre-trained model to this particular task is to add an attraction term that measures the L2-norm between the endpoints of generated trajectories and goal points.

We evaluated generation quality through JADE, which represents the realism of the generated scene, and JFDE, which reflects goal-reaching accuracy. As shown in Fig 3, all samples tend to accurately recover the demanded goal point while the generated trajectories are smooth and realistic. The variance of prediction samples tends to be large under cases where the pedestrian performs a sudden turn. However, in most cases, the model successfully covers the modality that most resembles the ground truth. We also present quantitive results in TABLE II, which showcases the effectiveness of goal point guidance.

C. Prediction with Noisy History

To assess the robustness of our proposed method against noisy input, we perturbed the observed history trajectories by introducing random Gaussian noise $n \sim \mathcal{N}(0, \sigma^2 I)$ of varying standard deviation σ. Specifically, we introduced two levels of perturbation: a slight level, characterized by a standard Gaussian noise with a 0.05-meter σ, and a heavy level, with a standard Gaussian noise with a 0.15-meter σ. Subsequently, we compared the performance of our model with the Joint AgentFormer baseline under identical experimental conditions.

As shown in TABLE III, we observed that our model performs better at heavy noise levels, providing trustworthy prediction even if the data is highly corrupted. The average JADE/JFDE performance of our method on 5 benchmarks is close to the baseline but only degenerated by 46.58%/38.38% at the heavy level. By contrast, the performance of the baseline method had degenerated by 57.08%/50.36% at heavy noise level. To reasonably state that the guided posterior sampling accounted for the robustness against noisy data, we further compared it with GFTD-

TABLE III: Prediction with Noisy Data

Noise Std	Model	minJADE / JFDE (m), K=20 ↓					
		ETH(1.4)	HOTEL(2.7)	UNIV(25.7)	ZARA1(3.3)	ZARA2(5.9)	ETH/UCY Avg.
0.00	Joint AgentFormer [5]	0.543 / 0.883	0.211 / 0.377	**0.596 / 1.247**	**0.309 / 0.612**	**0.282 / 0.584**	**0.388 / 0.741**
	Ours	**0.505 / 0.873**	**0.174 / 0.297**	0.649 / 1.305	0.340 / 0.667	0.308 / 0.620	0.395 / 0.752
0.05	Joint AgentFormer [5]	0.588 / **0.992**	0.253 / 0.433	**0.628 / 1.294**	**0.362 / 0.708**	0.361 / 0.714	**0.438** / 0.834
	Ours	**0.567** / 1.009	**0.198 / 0.334**	0.703 / 1.388	0.388 / 0.735	**0.341 / 0.675**	0.439 / **0.828**
0.15	Joint AgentFormer [5]	0.765 / 1.287	0.458 / 0.762	**0.834** / 1.642	0.649 / 1.254	0.736 / 1.312	0.688 / 1.254
	Ours	**0.636 / 1.076**	**0.283 / 0.463**	0.872 / **1.661**	**0.633 / 1.137**	**0.469 / 0.869**	**0.579 / 1.041**

TABLE IV: Prediction with Incomplete History Data

Missing ratio	Model	minJADE / JFDE (m), K=20 ↓					
		ETH(1.4)	HOTEL(2.7)	UNIV(25.7)	ZARA1(3.3)	ZARA2(5.9)	ETH/UCY Avg.
25%	Joint AgentFormer [5]	0.558 / 0.901	0.214 / 0.377	**0.624 / 1.286**	**0.336 / 0.653**	**0.303 / 0.614**	0.407 / 0.766
	Ours	**0.524 / 0.910**	**0.174 / 0.294**	0.662 / 1.321	0.341 / 0.670	0.314 / 0.627	**0.403 / 0.764**
50%	Joint AgentFormer [5]	0.619 / 1.022	0.219 / 0.378	**0.662 / 1.343**	0.374 / 0.721	0.327 / 0.649	0.440 / 0.823
	Ours	**0.511 / 0.897**	**0.176 / 0.302**	0.676 / 1.348	**0.347 / 0.677**	**0.321 / 0.642**	**0.406 / 0.773**
75%	Joint AgentFormer [5]	0.647 / 1.034	0.274 / 0.444	0.752 / 1.488	0.440 / 0.830	0.381 / 0.733	0.499 / 0.906
	Ours	**0.519 / 0.885**	**0.193 / 0.333**	**0.735 / 1.441**	**0.357 / 0.696**	**0.346 / 0.683**	**0.430 / 0.808**

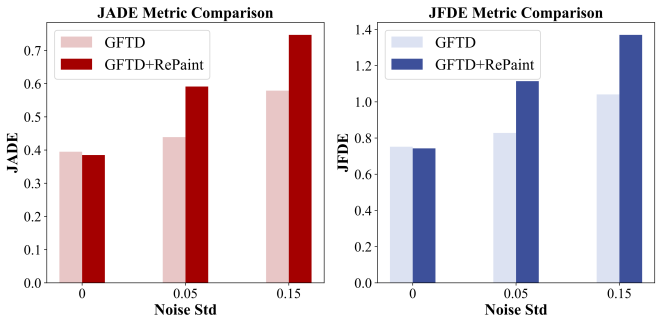

Fig. 4: JADE/JFDE performance comparison between GFDT and GFDT+RePaint with noisy data input.

RePaint. GFTD-RePaint includes gradually replacing the history part of the intermediate noisy sample by ground truth history observation, thus ensuring the generated samples were strictly conditioned on the noisy history data. The experimental results in Fig. 4 show that despite having a near performance to GFTD model under clean data conditions, the GFTD-RePaint performs much worse on noisy data, especially under heavy-level noise. It shows the superiority of our proposed framework in noisy conditions rooted in the design of "soft conditioning".

D. Prediction with Incomplete History

Due to obstacles or sensor failures, historical data may occasionally contain unexpected missing frames. To evaluate model robustness against incomplete data, we randomly select and mask 25%, 50%, and 75% of the historical trajectory frames, while ensuring that the current frame is always retained. With a known random mask, we only consider the available historical frames when calculating the guidance loss. We employ the same Joint AgentFormer model as our baseline. To ensure a fair comparison, we generate an attention mask for each AgentFormer sub-module based on the known data mask and replace the masked frames with zero values. For both models, we directly re-used the same pre-trained versions as in the other two experiments. As shown in TABLE IV, GFTD demonstrates competitive performance under conditions of incomplete data. Even when 75% of the historical observations are missing, the average JADE/JFDE performance only declines by 8.9% and 7.4%, respectively, still surpassing S-GAN with complete data input by a significant margin. In contrast, Joint AgentFormer fails to produce plausible predictions under high missing data rates.

VI. Conclusion & Discussion

In this work, we present the Guided Full Trajectory Diffuser, a novel framework for representing the joint distribution of trajectories leveraging diffusion models, converting trajectory prediction and controllable generation into a unified inverse problem. We formulate the prediction task as spatial-temporal inpainting, a general noisy inverse problem that can be solved through diffusion posterior sampling. Under this framework, the generated trajectories are not rigidly constrained by their historical observations; instead, we gradually enforce such conditions by adjustable posterior guidance. Such a unique design enables flexibility, resulting in not only competitive performance in joint trajectory prediction tasks but also generalizable to scenarios with noise perturbation or incomplete historical data. Moreover, our framework is compatible with plug-and-play modules and various guidance methods, thus expandable to task-oriented enhancements. However, we point out that our current im-

plementation only considers raw trajectories as guidance reference, how to more efficiently exploit the interaction to further improve guidance quality is a topic we will keep working on.

REFERENCES

[1] Ye Yuan, Xinshuo Weng, Yanglan Ou, and Kris M Kitani. Agentformer: Agent-aware transformers for socio-temporal multi-agent forecasting. In *ICCV*, pages 9813–9823, 2021.

[2] Chuhua Wang, Yuchen Wang, Mingze Xu, and David J Crandall. Stepwise goal-driven networks for trajectory prediction. *IEEE Robotics and Automation Letters*, 7(2):2716–2723, 2022.

[3] Jiangbei Yue, Dinesh Manocha, and He Wang. Human trajectory prediction via neural social physics. In *ECCV*, pages 376–394. Springer, 2022.

[4] 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 *ECCV*, pages 682–700. Springer, 2022.

[5] Erica Weng, Hana Hoshino, Deva Ramanan, and Kris Kitani. Joint metrics matter: A better standard for trajectory forecasting. *arXiv preprint arXiv:2305.06292*, 2023.

[6] Simone Zamboni, Zekarias Tilahun Kefato, Sarunas Girdzijauskas, Christoffer Norén, and Laura Dal Col. Pedestrian trajectory prediction with convolutional neural networks. *Pattern Recognition*, 121:108252, 2022.

[7] Yulong Cao, Danfei Xu, Xinshuo Weng, Zhuoqing Mao, Anima Anandkumar, Chaowei Xiao, and Marco Pavone. Robust trajectory prediction against adversarial attacks. In *Conference on Robot Learning*, pages 128–137. PMLR, 2023.

[8] Lívia Almada Cruz, Karine Zeitouni, and José Antonio F de Macedo. Trajectory prediction from a mass of sparse and missing external sensor data. In *MDM*, pages 310–319. IEEE, 2019.

[9] Ziwei Wang, Shiyao Zhang, and JQ James. Reconstruction of missing trajectory data: a deep learning approach. In *ITSC*, pages 1–6. IEEE, 2020.

[10] Mengshi Qi, Jie Qin, Yu Wu, and Yi Yang. Imitative non-autoregressive modeling for trajectory forecasting and imputation. In *CVPR*, pages 12736–12745, 2020.

[11] Yi Xu, Armin Bazarjani, Hyung-gun Chi, Chiho Choi, and Yun Fu. Uncovering the missing pattern: Unified framework towards trajectory imputation and prediction. In *CVPR*, pages 9632–9643, 2023.

[12] Chensheng Peng, Zhaoyu Zeng, Jinling Gao, Jundong Zhou, Masayoshi Tomizuka, Xinbing Wang, Chenghu Zhou, and Nanyang Ye. Pnas-mot: Multi-modal object tracking with pareto neural architecture search. *IEEE Robotics and Automation Letters*, 2024.

[13] Chensheng Peng, Chenfeng Xu, Yue Wang, Mingyu Ding, Heng Yang, Masayoshi Tomizuka, Kurt Keutzer, Marco Pavone, and Wei Zhan. Q-slam: Quadric representations for monocular slam. *arXiv preprint arXiv:2403.08125*, 2024.

[14] Tianpei Gu, Guangyi Chen, Junlong Li, Chunze Lin, Yongming Rao, Jie Zhou, and Jiwen Lu. Stochastic trajectory prediction via motion indeterminacy diffusion. In *CVPR*, pages 17113–17122, 2022.

[15] Weibo Mao, Chenxin Xu, Qi Zhu, Siheng Chen, and Yanfeng Wang. Leapfrog diffusion model for stochastic trajectory prediction. In *CVPR*, pages 5517–5526, 2023.

[16] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in Neural Information Processing Systems*, 35:26565–26577, 2022.

[17] Hyungjin Chung, Jeongsol Kim, Michael T Mccann, Marc L Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. *arXiv preprint arXiv:2209.14687*, 2022.

[18] Benjamin Boys, Mark Girolami, Jakiw Pidstrigach, Sebastian Reich, Alan Mosca, and O Deniz Akyildiz. Tweedie moment projected diffusions for inverse problems. *arXiv preprint arXiv:2310.06721*, 2023.

[19] Alexandros Graikos, Nikolay Malkin, Nebojsa Jojic, and Dimitris Samaras. Diffusion models as plug-and-play priors. *Advances in Neural Information Processing Systems*, 35:14715–14728, 2022.

[20] Michael Janner, Yilun Du, Joshua B Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. *arXiv preprint arXiv:2205.09991*, 2022.

[21] Ziyuan Zhong, Davis Rempe, Danfei Xu, Yuxiao Chen, Sushant Veer, Tong Che, Baishakhi Ray, and Marco Pavone. Guided conditional diffusion for controllable traffic simulation. In *ICRA*, pages 3560–3566. IEEE, 2023.

[22] Hongyi Chen, Jingtao Ding, Yong Li, Yue Wang, and Xiao-Ping Zhang. Social physics informed diffusion model for crowd simulation. *arXiv preprint arXiv:2402.06680*, 2024.

[23] Chiyu Jiang, Andre Cornman, Cheolho Park, Benjamin Sapp, Yin Zhou, Dragomir Anguelov, et al. Motiondiffuser: Controllable multi-agent motion prediction using diffusion. In *CVPR*, pages 9644–9653, 2023.

[24] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[25] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[26] Yixiao Wang, Chen Tang, Lingfeng Sun, Simone Rossi, Yichen Xie, Chensheng Peng, Thomas Hannagan, Stefano Sabatini, Nicola Poerio, Masayoshi Tomizuka, et al. Optimizing diffusion models for joint trajectory prediction and controllable generation. *arXiv preprint arXiv:2408.00766*, 2024.

[27] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *CVPR*, pages 11461–11471, 2022.

[28] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022.

[29] Inhwan Bae, Jean Oh, and Hae-Gon Jeon. Eigentrajectory: Low-rank descriptors for multi-modal trajectory forecasting. In *ICCV*, pages 10017–10029, 2023.

[30] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[31] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *CVPR*, pages 2255–2264, 2018.

[32] Nati Daniel, Ariel Larey, Eliel Aknin, Garrett A Osswald, Julie M Caldwell, Mark Rochman, Margaret H Collins, Guang-Yu Yang, Nicoleta C Arva, Kelley E Capocelli, et al. Pecnet: A deep multi-label segmentation network for eosinophilic esophagitis biopsy diagnostics. *arXiv preprint arXiv:2103.02015*, 2021.

[33] Chenxin Xu, Weibo Mao, Wenjun Zhang, and Siheng Chen. Remember intentions: Retrospective-memory-based trajectory prediction. In *CVPR*, pages 6488–6497, 2022.

[34] Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski. Crowds by example. In *Computer graphics forum*, volume 26, pages 655–664. Wiley Online Library, 2007.

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

[36] 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 *ECCV*, pages 759–776. Springer, 2020.

[37] Diganta Misra. Mish: A self regularized non-monotonic activation function. *arXiv preprint arXiv:1908.08681*, 2019.

[38] Tiankai Hang, Shuyang Gu, Chen Li, Jianmin Bao, Dong Chen, Han Hu, Xin Geng, and Baining Guo. Efficient diffusion training via min-snr weighting strategy. *arXiv preprint arXiv:2303.09556*, 2023.