# Unified Uncertainty-Aware Diffusion for Multi-Agent Trajectory Modeling

Guillem Capellera[1,2] Antonio Rubio[2] Luis Ferraz[2] Antonio Agudo[1]
[1]Institut de Robòtica i Informàtica Industrial, CSIC-UPC [2]Kognia Sports Intelligence
{guillem.capellera, antonio.agudo}@upc.edu {antonio.rubio, luis.ferraz}@kogniasports.com

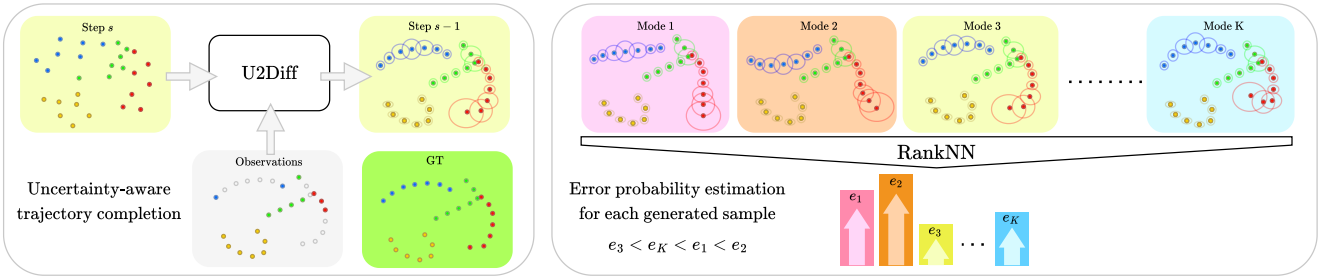


Figure 1. **Uncertainty-aware, unified and interpretable approach for trajectory modeling in multi-agent scenarios. U2Diff** is a diffusion-based model capable of performing trajectory completion tasks such as forecasting, imputation or inferring totally unseen agents, while also jointly estimating state-wise uncertainty. **RankNN** is a post-processing operation that infers an error probability for each generated mode under the same prior observations, which is strongly correlated with the error related to the ground truth.

## Abstract

*Multi-agent trajectory modeling has primarily focused on forecasting future states, often overlooking broader tasks like trajectory completion, which are crucial for real-world applications such as correcting tracking data. Existing methods also generally predict agents' states without offering any state-wise measure of uncertainty. Moreover, popular multi-modal sampling methods lack any error probability estimates for each generated scene under the same prior observations, making it difficult to rank the predictions during inference time. We introduce U2Diff, a* ***unified*** *diffusion model designed to handle trajectory completion while providing state-wise* ***uncertainty*** *estimates jointly. This uncertainty estimation is achieved by augmenting the simple denoising loss with the negative log-likelihood of the predicted noise and propagating latent space uncertainty to the real state space. Additionally, we incorporate a Rank Neural Network in post-processing to enable* ***error probability*** *estimation for each generated mode, demonstrating a strong correlation with the error relative to ground truth. Our method outperforms the state-of-the-art solutions in trajectory completion and forecasting across four challenging sports datasets (NBA, Basketball-U, Football-U, Soccer-U), highlighting the effectiveness of uncertainty and error probability estimation.* video

## 1. Introduction

Modeling trajectories in multi-agent settings is crucial for capturing stochastic human behaviors in various domains, including pedestrian motion prediction [2, 5, 18, 22, 30, 41, 42, 49, 51, 59], human pose estimation [1, 9, 17, 21, 27, 36, 37, 39], and sports analytics [4, 10, 25, 38, 45, 61, 65, 67].

Multi-modal generative approaches primarily focus on forecasting future states based on past trajectories, utilizing models such as Generative Adversarial Networks (GANs) [13, 22], Conditional Variational Auto-Encoders (CVAEs) [57, 64], and, more recently, Denoising Diffusion Probabilistic Models (DDPM) [23]. DDPM have shown particular success in trajectory forecasting for applications like pedestrian and sports modeling [20, 38]. However, their evaluation is often limited to agent-wise metrics, overlooking scene-level dynamics that are crucial for multi-agent contexts. Additionally, these methods generally require fixed temporal window dimensions, which restricts their adaptability across diverse task settings and scenarios.

The task of *trajectory completion* has emerged as a key advancement beyond traditional forecasting, enabling models to infer trajectories by leveraging both past and/or future observations [34, 47, 61]. This task also seeks to predict totally unobserved agents using only the motions of the surrounding observable ones [11, 29, 60]. This capability is especially relevant in sports, where complex multi-agent interactions require models to accurately capture both indi-

vidual and coordinated team tactical behaviors within fixed spatial coordinates.

However, current state-of-the-art methods in both trajectory forecasting and completion focus primarily on predicting locations without estimating the uncertainty associated with each predicted state. This limitation highlights the need for a state-wise uncertainty estimation approach to quantify each state prediction's closeness to the ground truth. Additionally, this gap presents a further challenge in developing methods to extract a scene-level uncertainty or error probability capable of ranking the reliability of multiple generated modes under the same prior.

In this study, we propose a **U**nified **U**ncertainty-aware **Diff**usion (U2Diff) aimed at tackling the general task of multi-agent trajectory completion while predicting per-state uncertainty with a novel variance propagation technique from latent to real space (see our pipeline in Fig. 1). Our method estimates global uncertainty by averaging the variances of each predicted agent's state. We show that this global uncertainty has certain correlation with the scene-level error across modes within the same prior, providing an unsupervised measure of confidence in the generated trajectories. To further refine the model's interpretability, we propose a supervised Rank Neural Network (RankNN) in order to rank modes based on their proximity to ground truth values, providing error probabilities and achieving high Spearman correlation values, with medians around $0.58$ and $0.78$.

We validate the effectiveness of our overall approach using four real-world sports datasets: two of basketball, one of football, and another of soccer; demonstrating substantial improvements over competing methods in scene-level metrics for forecasting and trajectory completion tasks. This work contributes a novel uncertainty-aware approach to trajectory modeling that enhances the reliability of generated trajectories in complex interactive environments like sports. The key contributions are summarized as:

- We propose a diffusion-based approach for general trajectory completion in multi-agent domain, achieving state-of-the-art performance.
- We introduce a simple loss augmentation in diffusion framework that enables direct uncertainty estimation for each predicted state. It ensures consistency across timesteps and moderate correlation with ground truth error, while enhancing prediction robustness.
- We devise a post-processing supervised architecture (RankNN) providing error probability estimates for each generated mode under a shared prior, enabling high-correlation with ground truth error.

## 2. Related Work

We next review the most related work dealing with trajectory modeling, diffusion models and uncertainty estimation.

**Trajectory Modeling.** Multi-modal agent trajectory modeling has traditionally focused on predicting future positions from past observations. Early methods used Variational Recurrent Neural Networks (VRNNs) to capture stochasticity in human long term movement prediction [16, 54, 63, 65, 67]. As the field evolved, GANs [13, 15, 22, 26, 50] and CVAEs [32, 35, 51, 57, 64] enabled more diverse and realistic future predictions by leveraging variational inference. Recently, diffusion models have demonstrated significant potential in generating diverse plausible futures [7, 20, 28, 33, 38, 48, 62], surpassing previous methods in forecasting tasks. However, these approaches are often limited by fixed time horizons. Other methods, such as Graph Variational Neural Networks (GVRNNs) [44, 61] and non-autoregressive techniques [34], have been developed for trajectory imputation tasks. Building on these foundations, our work introduces a diffusion-based architecture that integrates forecasting and imputation in a unified framework, adaptable to multi-agent scenarios without predefined agent or time dimension constraints.

**Unified diffusion models.** Time-series diffusion models have emerged as a viable alternative to Gaussian processes for probabilistic modeling [3, 55]. Our U2Diff architecture is inspired by CSDI [55], which we adapted for multi-agent 2D trajectory modeling by employing a bidirectional version of MambaSSM [19] to enhance temporal processing, replacing Transformer Encoder [56] focused on temporal dynamics processing. The sequential natural processing of MambaSSM allows to remove the temporal positional encoding while obtaining better performance.

**Uncertainty-aware.** Traditional models predict positions but overlook state-level uncertainty. Recent work introduced global uncertainty measures by aggregating individual agent uncertainties [38], but these lack the granularity needed to adapt diverse tasks such as trajectory completion. Inspired by pixel-wise uncertainty method in image generation [31], our model extends this to multi-agent trajectories, providing state-wise uncertainty, enabling finer-grained interpretability at state-level predictions.

**Probability estimation.** In multi-modal trajectory generation, probability estimation for each mode remains relatively unexplored. Existing methods assign probabilities using predefined trajectory anchors [12, 46, 52] or post-process the predicted trajectory [66]; however, they primarily focus on ego-agent scenarios. Latent sequential models [18, 51] introduce the estimation of probabilities at the scene-level requiring a fixed number of modes. To address this and adapt to sampling-based approaches like U2Diff, we propose RankNN, which estimates error probabilities for each scene-mode using all agents' trajectories and uncertainties. Unlike prior methods, RankNN supports a variable number of modes under a shared prior, acting as a post-processing network which provides ranked error probability estimates.

## 3. Revisiting Diffusion Models

We next review DDPM [23] that will be later employed to describe our method for uncertainty-aware multi-agent trajectory completion. They work by gradually adding random Gaussian noise to the original data in a *forward diffusion* process through a series of $S$ steps and then, learning to remove it in a *reverse denoising* one where original data is generated from the noise. To this end, let $\mathbf{X}_0$ be a data point from a real data distribution $q(\mathbf{X})$ where $\mathbf{X}$ is the input data. Some Gaussian noise with variance $\beta_s \in (0,1)$ can be added to $\mathbf{X}_{s-1}$, obtaining a new latent variable $\mathbf{X}_s$ with distribution $q(\mathbf{X}_s \mid \mathbf{X}_{s-1})$ as:

$$q(\mathbf{X}_s \mid \mathbf{X}_{s-1}) = \mathcal{N}\left(\mathbf{X}_s; \sqrt{1-\beta_s}\mathbf{X}_{s-1}, \beta_s \mathbf{I}\right), \quad (1)$$

$$q(\mathbf{X}_{1:S} \mid \mathbf{X}_0) = \prod_{s=1}^{S} q(\mathbf{X}_s \mid \mathbf{X}_{s-1}), \quad (2)$$

where $\mathbf{I}$ denotes an identity matrix, i.e., the distribution is always represented by a diagonal matrix of variances. Assuming a sufficiently large $S$, $\mathbf{X}_S \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. $\mathbf{X}_s$ can be sampled at any arbitrary time step from the distribution:

$$q(\mathbf{X}_s \mid \mathbf{X}_0) = \mathcal{N}(\mathbf{X}_s; \sqrt{\hat{\alpha}_s}\mathbf{X}_0, (1-\hat{\alpha}_s)\mathbf{I}), \quad (3)$$

where $\alpha_s = 1 - \beta_s$ and $\hat{\alpha}_s = \prod_{i=1}^{s} \alpha_i$. Then, $\mathbf{X}_s$ is expressed as:

$$\mathbf{X}_s = \sqrt{\hat{\alpha}_s}\mathbf{X}_0 + \sqrt{1-\hat{\alpha}_s}\boldsymbol{\epsilon}, \quad (4)$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Here $1 - \hat{\alpha_s}$ indicates the variance of the noise for an arbitrary time step, i.e., that could equivalently be used to define the noise schedule instead of $\beta_s$.

In the reverse diffusion process, a neural model is trained to infer the original data by reversing the previous noising process. Estimating $q(\mathbf{X}_{s-1} \mid \mathbf{X}_s)$ is a hard task as it depends on the entire data distribution and, therefore, a neural network $p_\theta(\cdot)$ is used to learn the $\theta$ diffusion parameters by parameterizing both mean and variance as:

$$p_\theta(\mathbf{X}_{s-1} \mid \mathbf{X}_s) = \mathcal{N}(\mathbf{X}_{s-1}; \boldsymbol{\mu}_\theta(\mathbf{X}_s, s), \sigma_\theta(\mathbf{X}_s, s)^2 \mathbf{I}), \quad (5)$$

$$p_\theta(\mathbf{X}_{0:S}) = p(\mathbf{X}_S) \prod_{s=1}^{S} p_\theta(\mathbf{X}_{s-1} \mid \mathbf{X}_s), \quad (6)$$

where $p(\mathbf{X}_S) = \mathcal{N}(\mathbf{X}_S; \mathbf{0}, \mathbf{I})$ and $\boldsymbol{\mu}_\theta(\cdot)$ and $\sigma_\theta(\cdot)^2\mathbf{I}$ represent mean and covariance matrix, respectively.

The mean in Eq. (5) can be obtained by considering the predicted noise $\boldsymbol{\epsilon}_\theta(\mathbf{X}_s, s)$ at $s$ step as:

$$\boldsymbol{\mu}_\theta(\mathbf{X}_s, s) = \frac{1}{\sqrt{\alpha_s}}\left(\mathbf{X}_s - \frac{\beta_s}{\sqrt{1-\hat{\alpha}_s}}\boldsymbol{\epsilon}_\theta(\mathbf{X}_s, s)\right), \quad (7)$$

where $\boldsymbol{\epsilon}_\theta$ is a trainable denoising function. In general, to infer the covariance matrix, the variance is assumed to be $\sigma_\theta(\mathbf{X}_s, s)^2 = \frac{1-\hat{\alpha}_{s-1}}{1-\hat{\alpha}_s}\beta_s$, i.e., it does not depend on the predicted noise. This parametrization is equivalent to rescaled score model for score-based generative models. Then, the reverse process can be trained by minimizing the function:

$$\mathcal{L}_{\text{simple}} = \mathbb{E}_{\mathbf{X}_0, \boldsymbol{\epsilon}, s}||\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{X}_s, s)||_2^2, \quad (8)$$

where $\boldsymbol{\epsilon}$ is a random but known noise. Later, Nichol *et al.* [43] found that learning the variance $\sigma_\theta(\mathbf{X}_s, s)^2$ improved the log-likelihood in images, as we will do in this work.

## 4. Multi-Agent Trajectory modeling by diffusion models

In this section we describe how to exploit probabilistic diffusion models to sort out trajectory completion in multi-agent scenarios. Our work is inspired by [55] that used a diffusion probabilistic framework for handling one-dimensional multivariate time-series imputation, exploiting visible observations to infer the non-visible ones. In contrast, in this work we present a unified approach for two-dimensional scenarios, where the relation between agents is richer and complex to capture.

### 4.1. Problem Statement

Let us consider a set of $N \in \mathbb{N}$ agent observations in a given time instant $t$, denoted as $\mathbf{x}_t = \{\mathbf{x}_t^n\}$ with $n = \{1, \ldots, N\}$, where each observation contains the $(x, y)$ locations. We can now collect $T$ observations along time for every agent, defining a scene tensor $\mathbf{X}$ where all $\mathbf{x}_t^n$ with $t = \{1, \ldots, T\}$ are considered. Trajectory completion aims at inferring missing or unobserved entries of a data structure based on the visible ones. Given a tensor of partial observations defined as $\mathbf{X}^{\text{co}}$ and a $T \times N$ binary conditioning mask $\mathbf{M}$ to encode by 1 the visible observations and by 0 the unobserved ones, the goal is to find a function $f(\cdot)$ to infer the full observations such that:

$$\mathbf{X} = f(\mathbf{X}^{\text{co}}, \mathbf{M}). \quad (9)$$

Particularly, in this paper we propose to model multi-agent trajectories by leveraging per-observation uncertainty estimation as:

$$p(\mathbf{X} \mid \mathbf{X}^{\text{co}}, \mathbf{M}) = \mathcal{N}(\mathbf{X}; f^{\mu}(\mathbf{X}^{\text{co}}, \mathbf{M}), f^{\sigma^2}(\mathbf{X}^{\text{co}}, \mathbf{M})), \quad (10)$$

where $f^{\mu}(\cdot)$ and $f^{\sigma^2}(\cdot)$ denote the function to extract mean and covariance matrix, respectively. As we propose to employ a generative model to handle the previous problem, at inference time the method obtains $K \in \mathbb{N}$ modes or scenes according to the same prior observations such that:

$$p(\mathbf{X}^k \mid \mathbf{X}^{\text{co}}, \mathbf{M}) \quad \forall k \in \{1, \ldots, K\}. \quad (11)$$

Once the trajectory completion problem is addressed, we propose estimating an error probability for each mode, which must be correlated with the ground truth locations.

### 4.2. Unified Uncertainty-aware Diffusion

We now present our Unified Uncertainty-aware Diffusion approach, denoted as U2Diff, to infer the set of distributions in Eq. (11). Our method can capture the uncertainty associated with each predicted agent state, obtaining both mean and variance of the predicted noise at each denoising step $s$.

Existing variance-learning approaches in image processing [43] minimize the variational lower bound by reducing the KL divergence between the predefined true posterior $q(\cdot)$ which follows a scheduled variance $\beta_s$ (see Section 3), and the model-predicted distribution $p_\theta(\cdot)$. In contrast, our method directly maximizes the likelihood of the noise injected during the forward diffusion pass $\boldsymbol{\epsilon}$, by modeling the distribution $\boldsymbol{\epsilon}_\theta(\boldsymbol{\epsilon} \mid \mathbf{X}_s, s, \mathbf{X}^{\text{co}})$ as:

$$\mathcal{N}(\boldsymbol{\epsilon}; \boldsymbol{\epsilon}^\mu_\theta(\mathbf{X}_s, s, \mathbf{X}^{\text{co}}), \boldsymbol{\epsilon}^\sigma_\theta(\mathbf{X}_s, s, \mathbf{X}^{\text{co}})^2 \mathbf{I}), \tag{12}$$

where $\boldsymbol{\epsilon}^\mu_\theta(\mathbf{X}_s, s, \mathbf{X}^{\text{co}})$ and $\boldsymbol{\epsilon}^\sigma_\theta(\mathbf{X}_s, s, \mathbf{X}^{\text{co}})$ are $[T \times N \times 2]$ predicted mean and standard deviation noise, respectively. Particularly, a diagonal covariance matrix across $x$ and $y$ noise components for each agent's state is assumed.

We propose a novel loss term $\mathcal{L}_{\text{NLL}}$ that minimizes the Negative Log-Likelihood (NLL) of the noise distribution as:

$$\mathcal{L}_{\text{NLL}} = \log\left(\sqrt{2\pi}\boldsymbol{\epsilon}^\sigma_\theta(\mathbf{X}_s, s, \mathbf{X}^{\text{co}}_0)\right) + \frac{||\boldsymbol{\epsilon} - \boldsymbol{\epsilon}^\mu_\theta(\mathbf{X}_s, s, \mathbf{X}^{\text{co}}_0)||^2_2}{2\boldsymbol{\epsilon}^\sigma_\theta(\mathbf{X}_s, s, \mathbf{X}^{\text{co}}_0)^2}, \tag{13}$$

This regularizer is added to the objective in Eq. (8), obtaining the total loss function:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{simple}} + \lambda \mathcal{L}_{\text{NLL}}, \tag{14}$$

where $\lambda$ is a weight factor (typically within the range $0.01$ to $0.001$) that balances the influence of $\mathcal{L}_{\text{NLL}}$ without overwhelming the primary learning objective. Following the approach in [43], we apply a stop-gradient to the predicted noise, $\boldsymbol{\epsilon}^\mu_\theta(\mathbf{X}_s, s, \mathbf{X}^{\text{co}}_0)$, so that $\mathcal{L}_{\text{NLL}}$ focuses solely on learning the standard deviation $\boldsymbol{\epsilon}^\sigma_\theta(\mathbf{X}_s, s, \mathbf{X}^{\text{co}}_0)$. This approach enables the model to represent both expected behavior and the associated uncertainties of agents' trajectories.

#### 4.2.1. Variance propagation

During sampling, variance propagation of the predicted noise to the states $(x, y)$ is key. To achieve that, we consider the deterministic Denoising Diffusion Implicit Model (DDIM) [53] with $\zeta$ as the fixed skipping interval:

$$\mathbf{X}_{s-\zeta} = \sqrt{\frac{\hat{\alpha}_{s-\zeta}}{\hat{\alpha}_s}}\mathbf{X}_s + \left(a_{s-\zeta} - \sqrt{\frac{\hat{\alpha}_{s-\zeta}}{\hat{\alpha}_s}}a_s\right)\boldsymbol{\epsilon}^\mu_\theta(\mathbf{X}_s, s, \mathbf{X}^{\text{co}}), \tag{15}$$

where $a_s = \sqrt{1 - \hat{\alpha}_s}$. Following the variance rule and similar to [31], we approximate the corresponding $\text{Var}(\mathbf{X}_{s-\zeta})$ as:

$$\frac{\hat{\alpha}_{s-\zeta}}{\hat{\alpha}_s}\text{Var}(\mathbf{X}_s) + \left(a_{s-\zeta} - \sqrt{\frac{\hat{\alpha}_{s-\zeta}}{\hat{\alpha}_s}}a_s\right)^2 \boldsymbol{\epsilon}^\sigma_\theta(\mathbf{X}_s, s, \mathbf{X}^{\text{co}})^2,$$

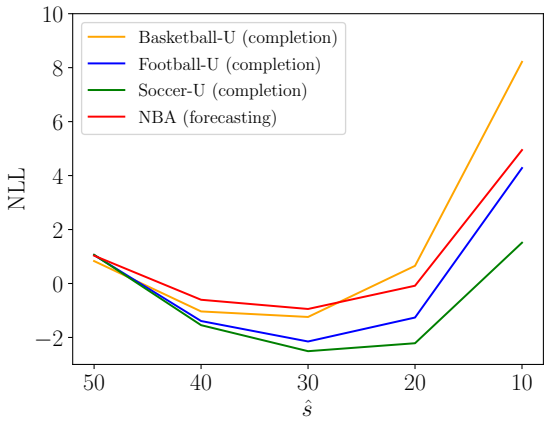


Figure 2. **Evaluation of the NLL over the predicted distribution states** in function of the starting denoising step $\hat{s}$ in which the variance starts propagating.

where $\text{Var}(\cdot)$ denotes the variance operator. The variance is initialized as a null tensor in the first denoising step $S$, i.e. $\text{Var}(\mathbf{X}_S) = \mathbf{0}$. The covariance term can be approximated as null to avoid high computational cost and potential instabilities without compromising performance.

Our analysis further reveals that beginning variance propagation at a smaller denoising step $\hat{s}$ than $S$, yields optimal performance across the datasets. This suggest that assuming $\text{Var}(\mathbf{X}_s) = \mathbf{0}$ for $s \in [\hat{s}, S]$ mitigates the effects of limited data expressivity in the early denoising steps, where the model lacks sufficient information to estimate the meaningful variance. Figure 2 presents the NLL of the predicted state distribution as a function of the starting step $\hat{s}$ for variance propagation across the four datasets. With a total of $S = 50$ denoising steps and a skip interval of $\zeta = 10$ denoising steps, we find that optimal variance propagation consistently starts at diffusion step $\hat{s} = 30$ across all datasets, ensuring robust generalization.

#### 4.2.2. Architecture

Inspired by [55], we introduce our architecture for multi-agent trajectory completion, designed to integrate uncertainty estimation seamlessly. To do so, some modifications are incorporated to enhance performance and adapt to our multi-agent domain. We next present the main ingredients in our contribution.

**Input embedding.** Initially, the observed trajectories $\mathbf{X}^{\text{co}}$ collected in a tensor with dimensions $T \times N \times 2$, which contain zero values $(0, 0)$ for unobserved states, are concatenated with the noised sample $\mathbf{X}_s$ of the same dimensionally, producing a combined tensor with dimensions $T \times N \times 4$. This tensor is then transformed into an embedding tensor $J$ through a linear layer followed by a ReLU activation, resulting in dimensions $T \times N \times 256$. That embedding is subsequently processed sequentially through two identical residual denoising blocks.

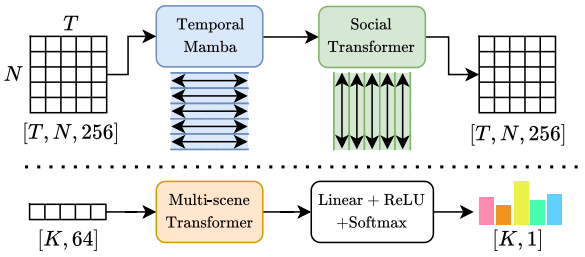


Figure 3. **U2Diff architecture. Top:** Decoupled temporal and social processing in each residual block. **Bottom:** Multi-scene attention processing and projection with Linear+ReLU+Softmax operations in RankNN to obtain the $K$ error probabilities $e$.

**Residual denoising block.** Each residual block comprises two main components. First, temporal processing is performed independently for each agent using a bidirectional version of the original MambaSSM [19], which we term *Temporal Mamba*. In this step, we compute two separate embeddings for each agent coming from the forward and reverse pass through the MambaSSM. These two embeddings are then summed to capture both past and future temporal information. Second, social processing is conducted to capture interactions between agents at each timestep using a Transformer Encoder [14, 56], termed the *Social Transformer*. This decoupling of temporal and social processing is illustrated in Fig. 3-top, giving the ability to infer scenes with variable timesteps and agents without fixing the temporal and social dimensions.

Within each residual block, the binary mask $\mathbf{M}$ is used to specify observed and unobserved states, facilitating accurate trajectory completion. Each block outputs a refined tensor $J$ with the same dimensions as its input, along with a skip-connection output $J_{\text{skip}}$, which is stored from each of the two blocks for later use. Additional details on the implementation of these residual blocks are provided in the supplementary material.

**Output tensor.** The output tensor is derived by summing the skip-connection outputs from the two residual blocks, resulting in a tensor of dimensions $T \times N \times 256$. This tensor is then passed through a linear layer with ReLU activation, producing a tensor of shape $T \times N \times 4$. Finally, that result is split to produce the predicted noise, $\boldsymbol{\epsilon}_\theta^\mu(\mathbf{X}_s, s, \mathbf{X}_0^{\text{co}})$, while the other tensor component is passed through a sigmoid function to generate the standard deviation $\boldsymbol{\epsilon}_\theta^\sigma(\mathbf{X}_s, s, \mathbf{X}_0^{\text{co}})$ with each value bounded in $(0, 1)$.

### 4.3. Rank Neural Network

To compute scene-level uncertainty, we use a simple approach which averages the predicted standard deviations $\sqrt{\text{Var}(\mathbf{X}_0)}$ across all agents and timesteps in a scene. We denote this operation as AvgUcty.

For a given scene prior observations and its $K$ generated modes, ideally the set of AvgUcty per-mode values and the set of corresponding their error values would correlate positively. In other words, higher AvgUcty values correlate with higher error values. The chosen scene-level metric for the error is the Scene Average Displacement Error (SADE) which is defined as:

$$\text{SADE} = \frac{\sum_{n=1}^{N}\sum_{t=1}^{T} \|\hat{\mathbf{x}}_t^n - \mathbf{x}_t^n\|_2 \, (1 - \mathbf{m}_t^n)}{\sum_{n=1}^{N}\sum_{t=1}^{T} (1 - \mathbf{m}_t^n)}, \quad (16)$$

where $\hat{\mathbf{x}}_t^n$ and $\mathbf{x}_t^n$ are the estimation and the corresponding ground truth, respectively, and $\mathbf{m}_t^n$ is the value of $\mathbf{M}$ indicating if the $n$-th agent at timestep $t$ is observed or not.

While AvgUcty provides a straightforward estimation of scene-level uncertainty, it may not fully capture its correlation with SADE. To address this, we introduce a novel learning-based approach that assigns an error probability score, $e$, to each mode, summing to 1 across the $K$ modes. These error probabilities are expected to show a stronger correlation with SADE values compared to AvgUcty. Specifically, we propose a RankNN model, which takes the $K$ generated modes, along with their predicted means and variances, and outputs logits that align with the SADE.

The objective function to maximize is the Spearman correlation coefficient ($\rho$) between the SADE values and the estimated $e$ values across all $K$ modes. This coefficient evaluates the monotonic relationship between these two sets. Let $e^k$ and $\text{SADE}^k$ represent the error probability estimation and the SADE, respectively, for the mode $k \in \{1, \ldots, K\}$. This coefficient, defined as the Pearson correlation between rank variables, is computed by first converting each pair ($e^k$, $\text{SADE}^k$) for all $K$ modes into differentiable ranks, denoted ($R[e^k]$, $R[\text{SADE}^k]$), with $R[\cdot]$ being the differentiable rank operator [8]. Therefore we can express $\rho$ as:

$$\rho = \frac{1}{K}\sum_{k=1}^{K}\left(\frac{(R[e^k] - \overline{R[e]}) \cdot (R[\text{SADE}^k] - \overline{R[\text{SADE}]})}{\|R[e^k] - \overline{R[e]}\| \cdot \|R[\text{SADE}^k] - \overline{R[\text{SADE}]}\|}\right),$$

where the terms $\overline{R[e]}$ and $\overline{R[\text{SADE}]}$ are the mean values over the $K$ generated modes. This correlation the normalization in the denominator ensures that is bounded within the interval $(-1, 1)$.

#### 4.3.1. Architecture

The architecture takes as input the mean $\mathbf{X}_0$ concatenated with its variance $\text{Var}(\mathbf{X}_0)$ for each state, creating a $K \times T \times N \times 4$ tensor. This is then extended to $K \times T \times N \times 5$ by appending the binary mask $\mathbf{M}$, repeated $K$ times.

The resulting tensor is embedded to a dimension of 64 and processed through a Temporal Mamba block to capture individual agent dynamics, with operations repeated across $K \times N$. After that, a Social Transformer models social interactions for each timestep, performing operations across

$K \times T$. After temporal and social processing, the tensor with dimensions $K \times T \times N \times 64$ is averaged across the timesteps and agents axis, resulting in scene-level embedding tensor $K \times 64$. This tensor is then passed through a Transformer Encoder to perform attention operation across the $K$ scenes, facilitating an efficient ranking. We refer this operation as *Multi-scene Transformer* and is depicted in Fig. 3-bottom. Finally, a linear layer with ReLU activation produces a vector of length $K$, which is normalized with a softmax function to yield the error probabilities $\{e^1, \ldots, e^K\}$. Notably, like our U2Diff, this architecture is flexible as it does not require a fixed number of timesteps $T$, agents $N$, or generated modes $K$ (see supplementary).

# 5. Experimental results

## 5.1. Datasets

For trajectory completion, we evaluate on three team sports datasets [60]: Basketball-U, Football-U, and Soccer-U. **Basketball-U** derives from NBA dataset [65] with 93,490 training and 11,543 testing sequences, each spanning 50 frames (8 seconds) capturing $(x, y)$ coordinates for 10 players and the ball. **Football-U**, based on the NFL Big-Data-Bowl[1] dataset, contains 10,762 training and 2,624 testing sequences of 50 frames, tracking $(x, y)$ locations for 22 players and the ball. **Soccer-U**, built from Soccer-Track[2] dataset, provides 9,882 training and 2,448 testing sequences, each also 50 frames, recording $(x, y)$ positions for 22 players and the ball. In [60], five masking strategies are defined for trajectory completion, including forecasting futures, imputing in-between states, and inferring the state of over five fully unobserved agents.

For trajectory forecasting, we use the NBA SportVU dataset (**NBA**) [40], with the same splits and normalization procedure as in LED [38]. The dataset records 30 frames (6 seconds) of $(x, y)$ positions for 10 players and the ball. The prediction task is to observe the first 2 seconds (10 frames) and forecast the subsequent 4 seconds (20 frames).

## 5.2. Implementation details

In our U2Diff, we use $S = 50$ diffusion steps during training, with $\lambda$ values set to $0.001$ for Basketball-U dataset and $0.01$ for the other three datasets. The diffusion noise scheduler starts with an initial value of $\beta_0 = 10^{-4}$ and ends with $\beta_S = 0.5$, following a quadratic function. Sampling is performed using DDIM, with a fixed skipping interval of $\zeta = 10$ denoising steps, reducing the reverse process to only six denoising steps: $s \in \{50, 40, 30, 20, 10, 1\}$. Optimal variance propagation starts at $\hat{s} = 30$. The final step ($s = 1$) follows the standard DDPM sampling and the variance is set as $\mathrm{Var}(\mathbf{X}_0) = \mathrm{Var}(\mathbf{X}_1)$. The Temporal Mamba's forward and reverse blocks are configured with a hidden size of 256, matching the configuration of the Social Transformer, which uses a 1024-dimensional feedforward layer and 8 attention heads. RankNN training involves generating 20 modes per scene online using the trained U2Diff with frozen weights. These generated samples are used to compute rankings based on their proximity to ground truth values. Additional implementation details are provided in the supplementary material.

[1] https://github.com/nfl-football-ops/Big-Data-Bowl

[2] https://github.com/AtomScott/SportsLabKit

## 5.3. Evaluation metrics

The first set of metrics are the commonly used the agent-wise metrics: $\text{minADE}_K$ as the minimum average displacement error, and $\text{minFDE}_K$ as the minimum final displacement error, both calculated over $K$ generated agent-modes. However, these metrics focus only on individual agent modes, lacking a full assessment of inter-agent scene dynamics. To address that, we include scene-level metrics: $\text{minSADE}_K$ as the minimum SADE (see Eq. (16)), and $\text{minSFDE}_K$ as minimum scene final displacement error, both calculated over $K$ generated scene-modes [18, 42].

We also adopt the metric used by [60] in trajectory completion evaluation, here referred to as $\text{minADE}_K$[60]. The Spearman correlation coefficient $\rho$ (see Eq. (17)) is used to assess AvgUcty and RankNN operations. Finally, the Accuracy Rate (AccRate) metric evaluates uncertainty quality by measuring the percentage of ground-truth states that fall within the predicted distribution with 95% confidence.

## 5.4. Comparison in trajectory modeling

In this section, we compare our approach with several state-of-the-art methods in trajectory completion and trajectory forecasting tasks.

In Table 1, we present the results for the $\text{minADE}_{20}$[60] and, for UniTraj [60] and our baselines, the $\text{minSADE}_{20}$ metric (shown in parentheses). Our method outperforms UniTraj, the strongest competing method, across all three completion datasets, achieving over 31% and 42% improvements in $\text{minADE}_{20}$[60] on the Football-U and Soccer-U datasets, respectively. When the number of agents is reduced, as in Basketball-U, we also obtain superior results, with a 27% improvement in $\text{minSADE}_{20}$. The table further includes an ablation study analyzing the impact of loss augmentation ($\lambda = 0$). The results indicate that omitting the loss augmentation does not degrade performance in terms of $\text{minSADE}_{20}$. Moreover, when evaluated using the $\text{minADE}_{20}$[60] metric, loss augmentation leads to improvements across all three datasets.

For the trajectory forecasting task, Table 2 presents the NBA dataset results. Our unified approach ranks second in agent-wise metrics $\text{minADE}_{20}$/$\text{minFDE}_{20}$, while achieving state-of-the-art performance in scene-level metrics $\text{minSADE}_{20}$/$\text{minSFDE}_{20}$, surpassing the diffusion-

| Method | Basketball-U (Feet) | Football-U (Yards) | Soccer-U (Pixels) |
|---|---|---|---|
| Mean | 14.58 | 14.18 | 417.68 |
| Median | 14.56 | 14.23 | 418.06 |
| Linear Fit | 13.54 | 12.66 | 398.34 |
| LSTM [24] | 7.10 | 7.20 | 186.93 |
| Transformer [56] | 6.71 | 6.84 | 170.94 |
| MAT [65] | 6.68 | 6.36 | 170.46 |
| Naomi [34] | 6.52 | 6.77 | 145.20 |
| INAM [47] | 6.53 | 5.80 | 134.86 |
| SSSD [3] | 6.18 | 5.08 | 118.71 |
| GC-VRNN [61] | 5.81 | 4.95 | 105.87 |
| UniTraj [60] | 4.77 (4.29) | 3.55 (4.03) | 94.59 (100.48) |
| **U2Diff** ($\lambda = 0$) | 4.68 (**3.10**) | 2.53 (2.37) | 54.41 (51.27) |
| **U2Diff** | **4.65** / (3.13) | **2.42** (**2.35**) | **53.93** (**51.14**) |

Table 1. **Evaluation in trajectory completion.** We compare our U2Diff with baseline methods in trajectory completion across three datasets. The metrics used are $\text{minADE}_{20}$[60]↓, and in parentheses, we report the $\text{minSADE}_{20}$↓ for both our U2Diff and the UniTraj baseline, computed using their original code and publicly available trained model.

| Method | NBA (Meters) | |
|---|---|---|
| | $\text{minADE}_{20}$ / $\text{minFDE}_{20}$ ↓ | $\text{minSADE}_{20}$ / $\text{minSFDE}_{20}$ ↓ |
| MemoNet [58] | 1.15 / 1.57 | - |
| NPSN [6] | 1.25 / 1.47 | - |
| GroupNet [57]$^{\chi}$ | 0.94 / 1.22 | 2.12 / 3.72 |
| AutoBots [18]$^{\psi}$ | 1.19 / 1.55 | 1.75 / 2.73 |
| MID [20] | 0.96 / 1.27 | - |
| LED [38] | **0.81** / **1.10** | 1.63 / 2.99 |
| **U2Diff** ($\lambda = 0$) | 0.86 / 1.11 | 1.50 / 2.70 |
| **U2Diff** | 0.85 / 1.11 | **1.48** / **2.68** |

Table 2. **Evaluation in trajectory forecasting.** We compare our U2Diff with baseline methods in trajectory forecasting on NBA dataset. We report four metrics, two agent-wise and two scene-level metrics. $^{\chi}$ means a new pretrained model from their codebase is used, with better results than the reported in the original work. $^{\psi}$ means trained using their original code.

based LED [38] method by over 9%. The sequential latent variable model AutoBots [18] also delivers competitive $\text{minSFDE}_{20}$ results. This table includes the same ablation as in Table 1, showing improvement with our proposed loss.

Figure 4 illustrates examples for trajectory completion and forecasting. The depicted trajectories are the modes with the $\text{minSADE}_{20}$. We compare in tajectory completion against the UniTraj [60], where our method delivers more accurate reconstructions and plausible predictions. For NBA forecasting, our model shows generally better future predictions—especially for ball trajectories—compared to LED [38], highlighting the effectiveness of scene-level metrics. Also note that our model is able to estimate variance in both tasks and reconstruct the observed states. Please refer to the supplementary material for additional qualitative results and an ablation study on the U2Diff architecture.

### 5.5. Uncertainty and error probability estimation

As previously mentioned, in Fig. 4 we present the predicted uncertainty for each state based on a 95% confidence interval. For trajectory completion, the ground-truth states fall

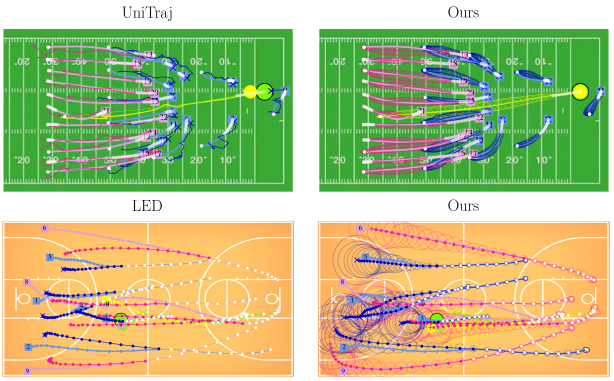


Figure 4. **Qualitative comparisons in trajectory completion (top) and forecasting (bottom).** Our U2Diff is compared with UniTraj [60] for trajectory completion and LED [38] for trajectory forecasting. Ground truth player locations are shown in bright blue and pink, and the ball in green. Model input observations are in white. The predicted mode with the best $\text{minSADE}_{20}$ is shown, with players in dark blue and pink, and the ball in yellow.

| Sampling | Basketball-U | Football-U | Soccer-U | NBA |
|---|---|---|---|---|
| Mean | 82.11 / -1.24 | 92.06 / -2.16 | 94.27 / -2.51 | 76.99 / -0.94 |
| Top-1 $e$ | 84.01 / -1.41 | 92.82 / -2.24 | 94.75 / -2.57 | 79.19 / -1.03 |
| Top-1 SADE | 86.77 / -1.76 | 93.95 / -2.39 | 95.63 / -2.66 | 85.70 / -1.31 |

Table 3. **Uncertainty evaluation using the % of AccRate↑ / NLL↓ metrics**. Results are depicted for all datasets.

within the predicted variance in nearly all cases, indicating robust uncertainty estimation. In contrast, NBA forecasting remains more challenging due to the difficulty in maintaining high confidence over longer prediction horizons.

To evaluate uncertainty quality, we first use Accuracy Rate (AccRate) and the Negative Log-Likelihood (NLL) in Table 3. We compute these metrics across three sampling strategies, depicted in each row: (Mean) average over all $K$ generated modes, (Top-1 $e$) mode with the lowest error probability $e$ predicted using RankNN, and (Top-1 SADE) mode with the minimum SADE. Football-U and Soccer-U datasets achieve AccRate values over 92%, indicating strong variance estimation. However, Basketball-U exhibits slightly lower AccRate and higher NLL values, reflecting the increased dynamics of the sport and the challenge of predicting trajectories for five or more fully unseen agents. The NBA dataset, which involves long-horizon forecasting, naturally exhibits lower AccRate and higher NLL. Notably, Top-1 $e$ selection outperforms Mean sampling, achieving higher AccRate and lower NLL, with results approaching Top-1 SADE. This demonstrates the RankNN's effectiveness in identifying reliable modes.

Another key metric is the Spearman correlation $\rho$ between the ranked $K = 20$ modes and SADE under the same prior. Ranking can be based on either AvgUcty (predicted uncertainty, relative to $\hat{s}$ in our method) or $e$ (error

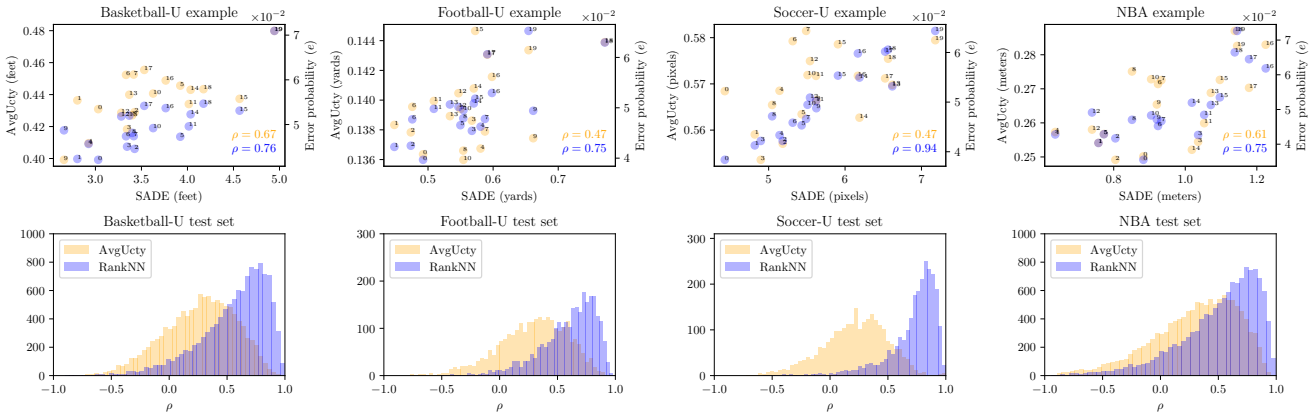


Figure 5. **Qualitative evaluation of the error correlation. Top**: In orange, the AvgUcty versus SADE across the 20 generated modes of a test scene example. In blue, the error probability $e$ versus SADE. **Bottom**: Distribution of Spearman correlation coefficients $\rho$ for all four test datasets, using AvgUcty in orange and RankNN predicting $e$ in blue.

| Method | Rank | $\hat{s}$ | Basketball-U | Football-U | Soccer-U | NBA |
|---|---|---|---|---|---|---|
| AutoBots [18] | AvgUcty | - | - | - | - | 0.09 / 0.10 |
| | $e$ | - | - | - | - | 0.37 / 0.44 |
| U2Diff | AvgUcty | 50 | 0.14 / 0.15 | 0.12 / 0.13 | 0.21 / 0.22 | 0.19 / 0.21 |
| | | 40 | 0.22 / 0.24 | 0.25 / 0.26 | 0.22 / 0.24 | 0.30 / 0.35 |
| | | 30 | 0.27 / 0.29 | 0.28 / 0.31 | 0.23 / 0.25 | 0.30 / 0.35 |
| | | 20 | 0.29 / 0.31 | 0.28 / 0.31 | 0.25 / 0.27 | 0.30 / 0.35 |
| | | 10 | 0.29 / 0.31 | 0.26 / 0.28 | 0.25 / 0.28 | 0.29 / 0.33 |
| | $e$ | - | **0.56 / 0.63** | **0.59 / 0.65** | **0.72 / 0.78** | **0.51 / 0.58** |

Table 4. **Evaluation of the correlation with error.** The results are the Mean / Median of Spearman correlation ($\rho \uparrow$) between the uncertainty or error probability estimations and the SADE.

| Method | Rank | NBA (Meters) | | | | |
|---|---|---|---|---|---|---|
| | | Top-1 | Top-3 | Top-5 | Top-10 | Top-20 |
| LED [38] | Random | 3.80 | 2.17 | 1.92 | 1.73 | 1.63 |
| AutoBots [18] | Random | 2.76 | 2.17 | 2.02 | 1.88 | 1.75 |
| | AvgUcty | 2.37 | 2.19 | 2.09 | 1.94 | 1.75 |
| | $e$ | 2.40 | 2.08 | 1.98 | 1.86 | 1.75 |
| U2Diff | Random | 2.01 | 1.75 | 1.66 | 1.56 | 1.48 |
| | AvgUcty | 1.91 | 1.71 | 1.63 | 1.55 | 1.48 |
| | $e$ | **1.82** | **1.66** | **1.60** | **1.54** | **1.48** |

Table 5. **Evaluation of the rank techniques and baselines comparisons.** From 20 generated modes for each method, the Top-$k$ best ones according the ranking method are selected, then the $\text{minSADE}_k \downarrow$ is computed over this subset.

probabilities). The mean and median values of $\rho$ for each scene across all datasets are shown in Table 4, with comparisons to AutoBots [18]. Notably, our predicted uncertainty and the estimated error probabilities achieve higher correlations in the NBA dataset. Note that the AvgUcty operation alone yields moderate correlations when $\hat{s} = 30$, with median values ranging from 0.25 to 0.35. This correlation improves further when using the RankNN approach. To illustrate this, Fig. 5-top shows four examples where we compare the AvgUcty operation and the error probabilities ($e$) against the SADE for each modes under the same prior. The blue dots corresponding to error probabilities demonstrate better rankings compared to the AvgUcty approach. The distribution of Spearman correlations across the entire test sets is presented in Fig. 5-bottom. See supplementary for the ablation analysis of RankNN inputs and components.

Finally, Table 5 shows results for different ranking strategies to select the Top-$k$ modes from a set of 20 and then compute the $\text{minSADE}_k$ for these selected modes. It is important to note that Top-20 is equivalent to $\text{minSADE}_{20}$. The ranking methods considered include Random, AvgUcty (when the model outputs variance) using $\hat{s} = 30$ in our framework, and $e$ ranking. We present results for the sampling-based LED [38] generative method as well as AutoBots [18], which can infer state-wise uncertainty and error probabilities for the generated modes. Our results show that, by using AvgUcty and the probabilities from RankNN, our method consistently outperforms the others, demonstrating a clear improvement in forecasting accuracy.

## 6. Conclusion

In this paper, we present U2Diff, a unified uncertainty-aware diffusion framework for general trajectory completion tasks. U2Diff not only outperforms state-of-the-art forecasting baselines in scene-level metrics but also sets a new benchmark in trajectory completion. We demonstrated its effectiveness in estimating state-wise uncertainty via a novel loss augmentation, without sacrificing the accuracy of state predictions. Our experiments reveal that U2Diff's uncertainty estimations exhibit a stronger correlation with ground truth errors compared to the scene-level state-of-the-art method. Additionally, we proposed a novel post-processing supervised RankNN model that infers error probability estimates for each mode, achieving a strong correlation with ground truth errors and also surpassing the scene-level based method.

**Acknowledgment.** This work has been supported by the project GRAVATAR PID2023-151184OB-I00 funded by MCIU/AEI/10.13039/501100011033 and by ERDF, UE and by the Government of Catalonia under 2023 DI 00058.

## References


[1] Emre Aksan, Manuel Kaufmann, Peng Cao, and Otmar Hilliges. A spatio-temporal transformer for 3d human motion prediction. In *2021 International Conference on 3D Vision (3DV)*, pages 565–574. IEEE, 2021. 1

[2] 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*, pages 961–971, 2016. 1

[3] Juan Miguel Lopez Alcaraz and Nils Strodthoff. Diffusion-based time series imputation and forecasting with structured state space models. *arXiv preprint arXiv:2208.09399*, 2022. 2, 7

[4] Michael A Alcorn and Anh Nguyen. baller2vec++: A look-ahead multi-entity transformer for modeling coordinated agents. *arXiv preprint arXiv:2104.11980*, 2021. 1

[5] Javad Amirian, Jean-Bernard Hayet, and Julien Pettré. Social ways: Learning multi-modal distributions of pedestrian trajectories with gans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 1

[6] Inhwan Bae, Jin-Hwi Park, and Hae-Gon Jeon. Non-probability sampling network for stochastic human trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6477–6487, 2022. 7

[7] Inhwan Bae, Young-Jae Park, and Hae-Gon Jeon. Singulartrajectory: Universal trajectory predictor using diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17890–17901, 2024. 2

[8] Mathieu Blondel, Olivier Teboul, Quentin Berthet, and Josip Djolonga. Fast differentiable sorting and ranking. In *International Conference on Machine Learning*, pages 950–959. PMLR, 2020. 5

[9] Yujun Cai, Lin Huang, Yiwei Wang, Tat-Jen Cham, Jianfei Cai, Junsong Yuan, Jun Liu, Xu Yang, Yiheng Zhu, Xiaohui Shen, et al. Learning progressive joint propagation for human motion prediction. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII 16*, pages 226–242. Springer, 2020. 1

[10] Guillem Capellera, Luis Ferraz, Antonio Rubio, Antonio Agudo, and Francesc Moreno-Noguer. Footbots: A transformer-based architecture for motion prediction in soccer. In *2024 IEEE International Conference on Image Processing (ICIP)*, pages 2313–2319. IEEE, 2024. 1

[11] Guillem Capellera, Luis Ferraz, Antonio Rubio, Antonio Agudo, and Francesc Moreno-Noguer. Transportmer: A holistic approach to trajectory understanding in multi-agent sports. In *Proceedings of the Asian Conference on Computer Vision*, pages 1652–1670, 2024. 1

[12] Yuning Chai, Benjamin Sapp, Mayank Bansal, and Dragomir Anguelov. Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. *arXiv preprint arXiv:1910.05449*, 2019. 2

[13] 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*, pages 13158–13167, 2021. 1, 2

[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 5

[15] Liangji Fang, Qinhong Jiang, Jianping Shi, and Bolei Zhou. Tpnet: Trajectory proposal network for motion prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6797–6806, 2020. 2

[16] Panna Felsen, Patrick Lucey, and Sujoy Ganguly. Where will they go? predicting fine-grained adversarial multi-agent motion using conditional variational autoencoders. In *Proceedings of the European conference on computer vision (ECCV)*, pages 732–747, 2018. 2

[17] Katerina Fragkiadaki, Sergey Levine, Panna Felsen, and Jitendra Malik. Recurrent network models for human dynamics. In *Proceedings of the IEEE international conference on computer vision*, pages 4346–4354, 2015. 1

[18] Roger Girgis, Florian Golemo, Felipe Codevilla, Martin Weiss, Jim Aldon D'Souza, Samira Ebrahimi Kahou, Felix Heide, and Christopher Pal. Latent variable sequential set transformers for joint multi-agent motion prediction. *arXiv preprint arXiv:2104.00563*, 2021. 1, 2, 6, 7, 8

[19] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023. 2, 5

[20] 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*, pages 17113–17122, 2022. 1, 2, 7

[21] Wen Guo, Yuming Du, Xi Shen, Vincent Lepetit, Xavier Alameda-Pineda, and Francesc Moreno-Noguer. Back to mlp: A simple baseline for human motion prediction. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 4809–4819, 2023. 1

[22] 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*, pages 2255–2264, 2018. 1, 2

[23] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 1, 3

[24] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 7

[25] Bo Hu and Tat-Jen Cham. Entry-flipped transformer for inference and prediction of participant behavior. In *European Conference on Computer Vision*, pages 439–456. Springer, 2022. 1

[26] Yue Hu, Siheng Chen, Ya Zhang, and Xiao Gu. Collaborative motion prediction via neural motion message passing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6319–6328, 2020. 2

[27] Ashesh Jain, Amir R Zamir, Silvio Savarese, and Ashutosh Saxena. Structural-rnn: Deep learning on spatio-temporal graphs. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 5308–5317, 2016. 1

[28] 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*, pages 9644–9653, 2023. 2

[29] Hyunsung Kim, Han-Jun Choi, Chang Jo Kim, Jinsung Yoon, and Sang-Ki Ko. Ball trajectory inference from multi-agent sports contexts using set transformer and hierarchical bi-lstm. *arXiv preprint arXiv:2306.08206*, 2023. 1

[30] 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. 1

[31] Siqi Kou, Lei Gan, Dequan Wang, Chongxuan Li, and Zhijie Deng. Bayesdiff: Estimating pixel-wise uncertainty in diffusion via bayesian inference. *arXiv preprint arXiv:2310.11142*, 2023. 2, 4

[32] 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*, pages 2221–2230, 2022. 2

[33] 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: 14400–14413, 2023. 2

[34] Yukai Liu, Rose Yu, Stephan Zheng, Eric Zhan, and Yisong Yue. Naomi: Non-autoregressive multiresolution sequence imputation. *Advances in neural information processing systems*, 32, 2019. 1, 2, 7

[35] 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*, pages 759–776. Springer, 2020. 2

[36] Wei Mao, Miaomiao Liu, Mathieu Salzmann, and Hongdong Li. Learning trajectory dependencies for human motion prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9489–9497, 2019. 1

[37] Wei Mao, Miaomiao Liu, and Mathieu Salzmann. History repeats itself: Human motion prediction via motion attention. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*, pages 474–489. Springer, 2020. 1

[38] 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*, pages 5517–5526, 2023. 1, 2, 6, 7, 8

[39] Julieta Martinez, Michael J Black, and Javier Romero. On human motion prediction using recurrent neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2891–2900, 2017. 1

[40] Alessio Monti, Alessia Bertugli, Simone Calderara, and Rita Cucchiara. Dag-net: Double attentive graph neural network for trajectory forecasting. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 2551–2558. IEEE, 2021. 6

[41] Ingrid Navarro and Jean Oh. Social-patternn: Socially-aware trajectory prediction guided by motion patterns. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9859–9864. IEEE, 2022. 1

[42] Jiquan Ngiam, Benjamin Caine, Vijay Vasudevan, Zhengdong Zhang, Hao-Tien Lewis Chiang, Jeffrey Ling, Rebecca Roelofs, Alex Bewley, Chenxi Liu, Ashish Venugopal, et al. Scene transformer: A unified architecture for predicting multiple agent trajectories. *arXiv preprint arXiv:2106.08417*, 2021. 1, 6

[43] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pages 8162–8171. PMLR, 2021. 3, 4

[44] Shayegan Omidshafiei, Daniel Hennes, Marta Garnelo, Zhe Wang, Adria Recasens, Eugene Tarassov, Yi Yang, Romuald Elie, Jerome T Connor, Paul Muller, et al. Multiagent off-screen behavior prediction in football. *Scientific reports*, 12 (1):8638, 2022. 2

[45] Marc Peral, Guillem Capellera, Antonio Rubio, Luis Ferraz, Francesc Moreno-Noguer, and Antonio Agudo. Temporally accurate events detection through ball possessor recognition in soccer. In *Proceedings of the International conference on Computer Vision Theory and Applications*, 2025. 1

[46] Tung Phan-Minh, Elena Corina Grigore, Freddy A Boulton, Oscar Beijbom, and Eric M Wolff. Covernet: Multimodal behavior prediction using trajectory sets. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14074–14083, 2020. 2

[47] Mengshi Qi, Jie Qin, Yu Wu, and Yi Yang. Imitative non-autoregressive modeling for trajectory forecasting and imputation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12736–12745, 2020. 1, 7

[48] 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*, pages 13756–13766, 2023. 2

[49] Saeed Saadatnejad, Yang Gao, Kaouther Messaoud, and Alexandre Alahi. Social-transmotion: Promptable human trajectory prediction. *arXiv preprint arXiv:2312.16168*, 2023. 1

[50] 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 con-*

*ference on computer vision and pattern recognition*, pages 1349–1358, 2019. 2

[51] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Multi-agent generative trajectory forecasting with heterogeneous data for control. *arXiv preprint arXiv:2001.03093*, 2, 2020. 1, 2

[52] 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*, pages 9675–9684, 2023. 2

[53] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 4

[54] Chen Sun, Per Karlsson, Jiajun Wu, Joshua B Tenenbaum, and Kevin Murphy. Stochastic prediction of multi-agent interactions from partial observations. *arXiv preprint arXiv:1902.09641*, 2019. 2

[55] Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. Csdi: Conditional score-based diffusion models for probabilistic time series imputation. *Advances in Neural Information Processing Systems*, 34:24804–24816, 2021. 2, 3, 4

[56] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2, 5, 7

[57] 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*, pages 6498–6507, 2022. 1, 2, 7

[58] Chenxin Xu, Weibo Mao, Wenjun Zhang, and Siheng Chen. Remember intentions: Retrospective-memory-based trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6488–6497, 2022. 7

[59] 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*, pages 1410–1420, 2023. 1

[60] Yi Xu and Yun Fu. Sports-traj: A unified trajectory generation model for multi-agent movement in sports. In *The Thirteenth International Conference on Learning Representations*, 2025. 1, 6, 7

[61] 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*, pages 9632–9643, 2023. 1, 2, 7

[62] Brian Yang, Huangyuan Su, Nikolaos Gkanatsios, Tsung-Wei Ke, Ayush Jain, Jeff Schneider, and Katerina Fragkiadaki. Diffusion-es: Gradient-free planning with diffusion for autonomous driving and zero-shot instruction following. *arXiv preprint arXiv:2402.06559*, 2024. 2

[63] Raymond A Yeh, Alexander G Schwing, Jonathan Huang, and Kevin Murphy. Diverse generation for multi-agent sports games. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4610–4619, 2019. 2

[64] 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*, pages 9813–9823, 2021. 1, 2

[65] Eric Zhan, Stephan Zheng, Yisong Yue, Long Sha, and Patrick Lucey. Generating multi-agent trajectories using programmatic weak supervision. *arXiv preprint arXiv:1803.07612*, 2018. 1, 2, 6, 7

[66] 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*, pages 895–904. PMLR, 2021. 2

[67] Stephan Zheng, Yisong Yue, and Jennifer Hobbs. Generating long-term trajectories using deep hierarchical networks. *Advances in Neural Information Processing Systems*, 29, 2016. 1, 2