

Uncovering the Missing Pattern: Unified Framework Towards Trajectory Imputation and Prediction

Yi Xu^{1,2,*} Armin Bazarjani^{2,3} Hyung-gun Chi^{2,4} Chiho Choi^{2,5} Yun Fu¹

¹Northeastern University ²Honda Research Institute, USA

³University of Southern California ⁴Purdue University ⁵Samsung Semiconductor US

xu.yi@northeastern.edu, bazarjan@usc.edu, hgchi@purdue.edu

chiho1.choi@samsung.com, yunfu@ece.neu.edu

Abstract

Trajectory prediction is a crucial undertaking in understanding entity movement or human behavior from observed sequences. However, current methods often assume that the observed sequences are complete while ignoring the potential for missing values caused by object occlusion, scope limitation, sensor failure, etc. This limitation inevitably hinders the accuracy of trajectory prediction. To address this issue, our paper presents a unified framework, the Graph-based Conditional Variational Recurrent Neural Network (GC-VRNN), which can perform trajectory imputation and prediction simultaneously. Specifically, we introduce a novel Multi-Space Graph Neural Network (MS-GNN) that can extract spatial features from incomplete observations and leverage missing patterns. Additionally, we employ a Conditional VRNN with a specifically designed Temporal Decay (TD) module to capture temporal dependencies and temporal missing patterns in incomplete trajectories. The inclusion of the TD module allows for valuable information to be conveyed through the temporal flow. We also curate and benchmark three practical datasets for the joint problem of trajectory imputation and prediction. Extensive experiments verify the exceptional performance of our proposed method. As far as we know, this is the first work to address the lack of benchmarks and techniques for trajectory imputation and prediction in a unified manner.

1. Introduction

Modeling and predicting future trajectories play an indispensable role in various applications, i.e., autonomous driving [23, 25, 65], motion capture [57, 59], behavior understanding [20, 38], etc. However, accurately predicting movement patterns is challenging due to their complex and

*Work done during Yi’s internship at Honda Research Institute, under Chiho Choi’s supervision.

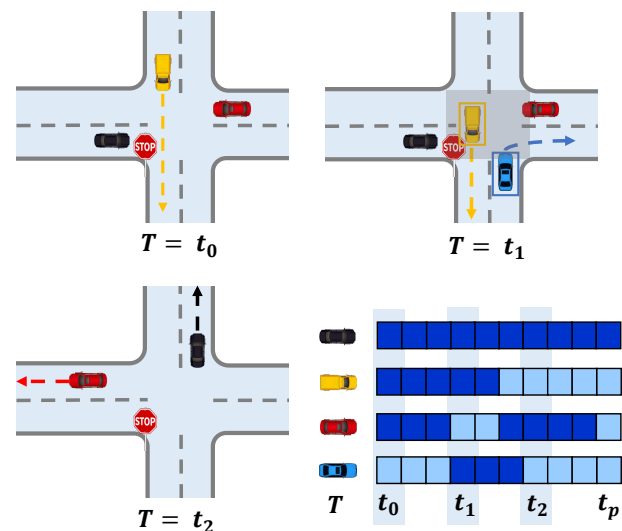


Figure 1. A typical example of incomplete observed trajectory. The black car (ego-vehicle) is waiting at the intersection at time step t_0 , and the yellow car is moving. At time step t_1 , the red car is occluded by the yellow car, and the blue car appears. The bottom right figure indicates the “visibility” of four cars, where dark means visible and light color means invisible.

subtle nature. Despite significant attention and numerous proposed solutions [5, 34, 62, 67–69] to the trajectory prediction problem, existing methods often assume that agent observations are entirely complete, which is too strong an assumption to satisfy in practice.

For example, in sports games such as football or soccer, not all players are always visible in the live view due to the limitation of the camera view. In addition, the live camera always tracks and moves along the ball, resulting in some players appearing and disappearing throughout the view, depending on their relative locations to the ball. Fig. 2 illustrates this common concept. Similar situations also arise in autonomous driving where occlusion or sensor failure can cause missing data. As illustrated in Fig. 1, at time step

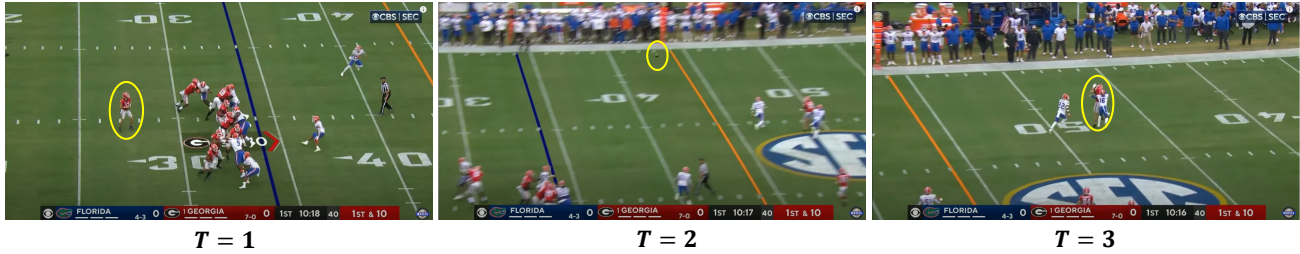


Figure 2. Three continuous frames of a “throw-and-catch” sequence in a live football match, where the ball is being circled. During this attacking play, the camera follows and zooms in on the ball. Only a subset of players is visible within the camera’s field of view.

t_0 , there is no observation of the blue car. At time step t_1 , the red car is occluded by the yellow car from the black car perspective, and the blue car appears at the intersection to turn right. Predicting future trajectories of entities under these circumstances will no doubt hinder the performance and negatively influence behavior understanding of moving agents or vehicle safety operations.

Although various recent works [15, 17, 37, 53, 73, 74] have investigated the time series imputation problem, most are autoregressive models that impute current missing values from previous time steps, making them highly susceptible to compounding errors for long-range temporal modeling. Additionally, commonly used benchmarks [29, 50, 54, 82] do not contain interacting entities. Some recent works [36, 78] have studied the imputation problem in a multi-agent scenario. Although these methods have achieved promising imputation performance, they fail to explore the relevance between the trajectory imputation and the prediction task. In fact, complete trajectory observation is essential for prediction, and accurate trajectory prediction can offer valuable information about the temporal correlation between past and future, ultimately aiding in the imputation process.

In this paper, we present a unified framework, Graph-based Conditional Variational Recurrent Neural Network (GC-VRNN), that simultaneously handles the trajectory imputation and prediction. Specifically, we introduce a novel Multi-Space Graph Neural Network (MS-GNN) to extract compact spatial features of incomplete observations. Meanwhile, we adopt a Conditional VRNN (C-VRNN) to model the temporal dependencies, where a Temporal Decay (TD) module is designed to learn the missing patterns of incomplete observations. The critical idea behind our method is to acquire knowledge of the spatio-temporal features of missing patterns, and then unite these two objectives through shared parameters. Sharing valuable information allows these two tasks to support and promote one another for better performance mutually. In addition, to support the joint evaluation of multi-agent trajectory imputation and prediction, we curate and benchmark three practical datasets from different domains, *Basketball-TIP*, *Football-TIP*, and *Vehicle-TIP*, where the incomplete trajectories are generated via reasonable and practical strategies. The main con-

tributions of our work can be summarized as follows:

- We investigate the multi-agent trajectory imputation and prediction problem and develop a unified framework, GC-VRNN, for imputing missing observations and predicting future trajectories simultaneously.
- We propose a novel MS-GNN that can extract comprehensive spatial features of incomplete observations and adopt a C-VRNN with a specifically designed TD module for better learning temporal missing patterns, and valuable information is shared via temporal flow.
- We curate and benchmark three datasets for the multi-agent trajectory imputation and prediction problem. Strong baselines are set up for this joint problem.
- Thorough experiments verify the consistent and exceptional performance of our proposed method.

2. Related Work

2.1. Trajectory Prediction

The objective of trajectory prediction is to predict the future positions of agents conditioned on their observations. A pioneering study, Social-LSTM [2], introduces a pooling layer that facilitates the sharing of human-human interaction features. Following this, some methods [26, 64, 71, 79] have been proposed to extract comprehensive interaction features. Considering the uncertainty of human trajectory, some works use generative models such as Generative Adversarial Networks (GANs) [3, 22, 31, 33, 48] and Variational Autoencoders (VAEs) [28, 41, 49, 68, 69] to generate multiple trajectory predictions. Recently, Transformer structure [63] is applied in this task [21, 62, 76, 77] to model the spatio-temporal relations via an attention mechanism. Moreover, various viewpoints have emerged towards more practical applications, i.e., goal-driven idea [13, 40, 60, 81], long-tail situation [39], interpretability [32], robustness [9, 66, 70, 80], counterfactual analysis [11], planning-driven [12], generalization ability to new environment [6, 27, 72], and knowledge distillation [44].

Typically, in graph-based models [5, 31, 34, 43, 51, 55, 67], each individual is considered as a single node, while the

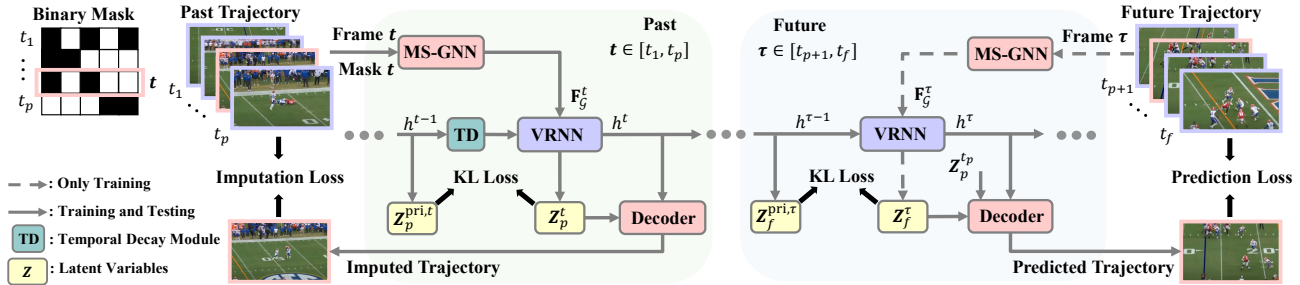


Figure 3. Overview of our GC-VRNN. The inputs are the past incomplete trajectory, the corresponding mask, and the future trajectory (only used for training). The outputs are imputed trajectory and predicted trajectory. Our model jointly handles the imputation and prediction problem, meanwhile, is trained in an end-to-end fashion.

connections between them are depicted as edges. Graph Convolutional Layers (GCLs) and a Message Passing (MP) mechanism are utilized to extract spatio-temporal characteristics. However, these methods presume that observations are complete, which can be difficult to meet in real-world situations. Furthermore, these graph-based models are unable to identify missing patterns, whereas our proposed approach can reveal incomplete spatio-temporal patterns.

2.2. Trajectory Imputation

Some statistical imputation techniques substitute missing values with the mean or median value [1]. Other alternatives, such as linear fit [4], k-nearest neighbours [7, 61], and expectation-maximization (EM) algorithm [19, 45], are also adopted. One of the biggest limitations of such methods is using rigid priors, which hinders the generalization ability. A more flexible framework is utilized with generative methods to learn the missing pattern. For instance, some deep autoregressive methods based on RNNs [8, 10, 35, 75] are proposed to impute the sequential data. Some other methods [18, 37, 42, 47, 58, 74] have been proposed to leverage GANs or VAEs to generate reconstructed sequences.

Few works explore the trajectory imputation problem in the multi-agent scenario. Notably, NAOMI [36] presents a non-autoregressive imputation method that exploits the multi-resolution structure of sequential data for imputation. GMAT [78] designs a hierarchical model to produce weak macro-intent labels for sequence generation. However, these two methods only focus on the trajectory imputation task and fail to investigate the prediction task. In work INAM [47], an imitation learning paradigm is proposed to handle the imputation and prediction in an asynchronous mode. While our model handles these two tasks simultaneously and is trained in an end-to-end fashion. Furthermore, method INAM is solely assessed on a single multi-agent dataset, with missing instances being generated at random. We argue that this arbitrary masking technique is not practical in real-world scenarios. Conversely, our work has been validated as effective across various multi-agent domains. Most importantly, we also leverage missing patterns in the

realm of spatio-temporal features.

3. Problem Definition

Consider an observed set of N agents $\Omega = \{1, 2, \dots, N\}$ over time step t_1 to t_p . Let $X_i^{\leq t_p} = \{x_i^1, \dots, x_i^t, \dots, x_i^{t_p}\}$ denote the observed trajectory of agent i , where $x_i^t \in \mathbb{R}^2$ represents the 2D coordinates of agent i at time step t . The observed trajectory set is thus defined as $X_\Omega^{\leq t_p} = \{X_i^{\leq t_p} | \forall i \in \Omega\}$. Because some observations for any subset of agents could be missing at any time due to occlusion, sensor failure, *etc.* The missing locations are represented by a masking matrix $M_i^{\leq t_p} = \{m_i^1, \dots, m_i^t, \dots, m_i^{t_p}\}$ valued in $\{0, 1\}$. The variable m_i^t is assigned a value of 1 if the observation is available at time step t and 0 otherwise.

The goal of the joint problem of multi-agent trajectory imputation and prediction is to impute missing values of all agents observations from time step t_1 to t_p , and also predict their future trajectory from time step t_{p+1} to t_f conditioned on their incomplete observations. More formally, that is to learn a model $f(\cdot)$ with parameter W^* that outputs $\hat{X}_\Omega^{\leq t_p}$ and $\hat{Y}_\Omega^{t_{p+1} \leq t \leq t_f}$, where $\hat{X}_\Omega^{\leq t_p}$ refers to the imputed trajectory and $\hat{Y}_\Omega^{t_{p+1} \leq t \leq t_f}$ refers to the predicted trajectory.

4. Proposed Method

Fig. 3 illustrates our proposed method at a high level. The fundamental idea is to address the task of multi-agent trajectory imputation and prediction in a unified framework while promoting information exchange via temporal flow.

4.1. MS-GNN

Applying a graph-based approach is a natural decision to model the spatial correlations of multiple agents. However, as we are addressing the unique challenge of incomplete observations of multiple agents, we develop a novel approach called Multi-Space Graph Neural Network (MS-GNN) that enhances the capability of Graph Convolutional Layers (GCL).

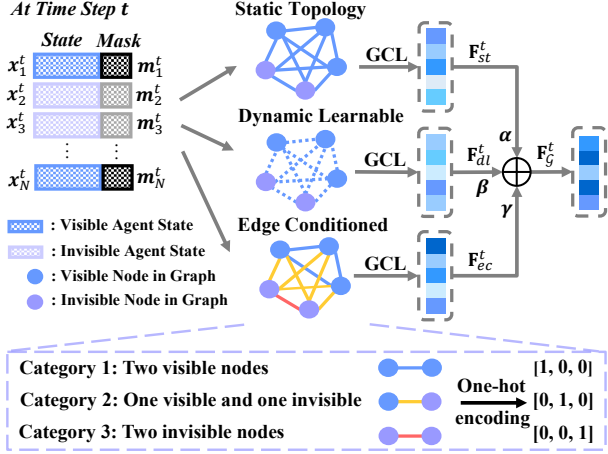


Figure 4. Schematic diagram of three different agent-wise graph convolutional layers at time step t . The output features of three GCLs are finally integrated to \mathbf{F}_G^t via weighted sum.

Graph Construction. Each agent is considered as a single node in the graph, and the graph at time step t is defined as $\mathcal{G}^t = (\mathcal{V}^t, \mathcal{E}^t)$, where $\mathcal{V}^t = \{v_i^t | i \in \Omega\}$ denotes the vertex set of agents, $\mathcal{E}^t = \{e_{i,j}^t | i, j \in \Omega\}$ denotes the edge set captured by an adjacency matrix $\mathbf{A}^t = \{a_{i,j}^t | i, j \in \Omega\}$. The graph feature representation is defined as $\mathbf{F}^t = \{\mathbf{f}_i^t \in \mathbb{R}^D | i \in \Omega\}$, where \mathbf{f}_i^t is the feature vector of node i at time step t , D denotes the dimension of node feature vector.

Graph Input. The inputs of MS-GNN are the incomplete observed trajectory, the corresponding binary mask that indicates the missing status, and the future trajectory. Note that the future trajectory is only used in the training phase. For instance, consider agent i at time step t , the node feature is initialized by projecting inputs to high-dimensional feature vectors, which are defined as follows:

$$\mathbf{f}_i^t = \begin{cases} \varphi_p((\mathbf{x}_i^t \odot \mathbf{m}_i^t) \oplus \mathbf{m}_i^t; \mathbf{W}_p) & t \in [t_1, t_p] \\ \varphi_f(\mathbf{y}_i^t; \mathbf{W}_f) & t \in [t_{p+1}, t_f] \end{cases}, \quad (1)$$

where $\varphi_p(\cdot)$ and $\varphi_f(\cdot)$ are different projection functions with weights $\mathbf{W}_p \in \mathbb{R}^{3 \times D}$ and $\mathbf{W}_f \in \mathbb{R}^{2 \times D}$, respectively. In our implementation, we achieve this using MLPs. \odot denotes the element-wise multiplication, and \oplus denotes the concatenation operation. \mathbf{y}_i^t represents the future location of agent i , which is only used in the training phase.

In MS-GNN, we define three different GCLs to extract spatial features from different feature spaces at each time step. Each GCL is designed to extract primary features for intuitive purposes, and meanwhile emphasize the spatial correlations of missing patterns in observations. To avoid confusion and for simplicity, we omit the **superscript** t of node feature \mathbf{f}_i^t as \mathbf{f}_i , and graph feature \mathbf{F}^t as \mathbf{F} .

Static Topology GCL. In the vanilla GCL [30], the adjacency matrix only indicates the connectivity of node pairs,

where $\mathbf{A}_{i,j} = 1$ if an edge directs from node i to j and 0 otherwise. While in our case, some of the agents (nodes) are missing in the observed trajectory. Therefore, we define a different adjacency matrix \mathbf{A}_{st} not only to indicate the connectivity but also the visibility in static topology GCL. The identity matrix \mathbf{I}_{st} is adjusted accordingly with a constraint for adding the self-loop. Constraints are defined as follows:
Constraint 1: $\mathbf{A}_{i,j}^t = \mathbf{A}_{j,i}^t = 1$ if node i and j are both visible at time step t . Otherwise, $\mathbf{A}_{i,j}^t = \mathbf{A}_{j,i}^t = 0$.
Constraint 2: $\mathbf{I}_{i,i}^t = 1$ if node i is visible at time step t . Otherwise, $\mathbf{I}_{i,i}^t = 0$.

Similar to [30], the propagation rule of graph feature $\mathbf{F}_{st}^{(l)}$ of the l -th static topology layer is defined as follows:

$$\mathbf{F}_{st}^{(l+1)} = \sigma \left(\hat{\mathbf{A}}_{st} \mathbf{F}_{st}^{(l)} \mathbf{W}_{st}^{(l)} \right), \quad (2)$$

where normalized adjacency matrix $\hat{\mathbf{A}}_{st} = \tilde{\mathbf{D}}^{-1/2} (\mathbf{A}_{st} + \mathbf{I}_{st}) \tilde{\mathbf{D}}^{1/2}$, $\tilde{\mathbf{D}}$ is the diagonal degree of $\mathbf{A}_{st} + \mathbf{I}_{st}$, $\mathbf{W}_{st}^{(l)} \in \mathbb{R}^{D^{(l)} \times D^{(l+1)}}$ are learnable parameters of the l -th static topology layer, and $\sigma(\cdot)$ denotes the ReLU activation function. This static topology GCL models the connectivity and visibility features of agents in a fixed way.

Dynamic Learnable GCL. In contrast to the \mathbf{A}_{st} in topology GCL with fixed values (0 or 1), inspired by [52], we define a simple, learnable, and unconstrained \mathbf{A}_{dl} to dynamically learn the strength of relations between nodes, and to improve the flexibility of GCL. The matrix \mathbf{A}_{dl} is initialized with random values and is trained to modify the edges by either strengthening, weakening, adding, or removing them. Similar to Eq. (2), the propagation rule of graph feature $\mathbf{F}_{dl}^{(l)}$ is defined as follows:

$$\mathbf{F}_{dl}^{(l+1)} = \sigma \left(\mathbf{A}_{dl} \mathbf{F}_{dl}^{(l)} \mathbf{W}_{dl}^{(l)} \right), \quad (3)$$

where $\mathbf{W}_{dl}^{(l)} \in \mathbb{R}^{D^{(l)} \times D^{(l+1)}}$ are learnable parameters of the l -th dynamic learnable layer. Since all the elements in \mathbf{A}_{dl} are learnable with no constraint, the \mathbf{A}_{dl} will be asymmetric that allows each edge to select the best suitable relation strength to update its corresponding node features. Intuitively, compared to the static topology GCL, the relations among agents (nodes) are better captured by this GCL.

Edge Conditioned GCL. The aforementioned two GCLs focus on learning spatial relations among nodes and the different strengths of such relations by two different definitions of the adjacency matrix. While in our case, one challenge is that node features of some agents are missing in the incomplete observations. In order to better understand the spatial missing patterns, we leverage an edge conditioned GCL, where we assign a label to each edge based on its category and integrate such category information in graph propagation. As shown in Fig. 4, three types of edges exist, which are determined by the visibility of the corresponding node pair. We first encode three categories into

one-hot vectors $\vartheta_{i,j} \in \mathbb{R}^3$, and then define a mapping network $\varphi_{ec}(\cdot)$ to output the edge-specific weight matrix $\Theta_{i,j} \in \mathbb{R}^{D_G \times D}$ for updating the node features. The updating rule in edge conditioned GCL is defined as:

$$\begin{aligned} \mathbf{f}_{ec;i} &= \frac{1}{|\mathcal{V}(i)|} \sum_{j \in \mathcal{V}(i)} \varphi_{ec}(\vartheta_{i,j}; \mathbf{W}_{ec}) \mathbf{f}_j + \mathbf{b}_{ec} \\ &= \frac{1}{|\mathcal{V}(i)|} \sum_{j \in \mathcal{V}(i)} \Theta_{i,j} \mathbf{f}_j + \mathbf{b}_{ec}, \end{aligned} \quad (4)$$

where $\mathbf{f}_{ec;i}$ denotes the feature vector for node i in \mathbf{F}_{ec} , while \mathbf{W}_{ec} represents the learnable parameters of the l -th edge conditioned layer, \mathbf{b}_{ec} is the learnable bias, and $\mathcal{V}(i)$ denotes the neighboring nodes set of node i . In our implementation, we utilize a Conv2D block consisting of two Conv2D layers and one average pooling layer. Note that this GCL is only employed for the observations.

Graph Feature Fusion. Upon obtaining the last layer graph feature representations, namely \mathbf{F}_{st} , \mathbf{F}_{dl} , and \mathbf{F}_{ec} , we integrate them into a final graph representation denoted as $\mathbf{F}_G \in \mathbb{R}^{N \times D_G}$ through the following equation. This is achieved by setting the feature dimension of the final layer of each GCL as D_G .

$$\mathbf{F}_G = \alpha \mathbf{F}_{st} + \beta \mathbf{F}_{dl} + \gamma \mathbf{F}_{ec}, \quad (5)$$

where α , β , and γ are three learnable parameters for feature fusion with the same size \mathbb{R}^{D_G} .

4.2. C-VRNN with TD

In our work, we leverage a C-VRNN for modeling both past and future trajectory temporal dependencies. In the imputation stream, we introduce a Temporal Decay (TD) module that is dependent on the time interval between the preceding observation and the current time step. Furthermore, we recurrently update the priors of trajectory and the latent variables of imputation and prediction streams via a parameter-shared temporal flow. Valuable information is promoted to exchange implicitly with one another.

The vanilla VRNN [14] can be considered as a basic VAE conditioned on the hidden states of an RNN and it is trained by maximizing the Sequential ELBO as follows:

$$\begin{aligned} \mathbb{E}_{q_\phi(z^{\leq T} | \mathbf{x}^{\leq T})} \left[\sum_{t=1}^T \log p_\theta(\mathbf{x}^t | \mathbf{z}^{\leq t}, \mathbf{x}^{< t}) \right. \\ \left. - \text{KL}(q_\phi(\mathbf{z}^t | \mathbf{x}^{\leq t}, \mathbf{z}^{< t}) || p_\theta(\mathbf{z}^t | \mathbf{x}^{< t}, \mathbf{z}^{< t})) \right]. \end{aligned} \quad (6)$$

Prior. The distribution for the prior on the latent variables \mathbf{z}^t follows the following format at each time step:

$$\mathbf{z}^t \sim \mathcal{N}(\boldsymbol{\mu}^{\text{pri},t}, \boldsymbol{\sigma}^{\text{pri},t^2}), \quad (7)$$

where the distribution parameters $\boldsymbol{\mu}^{\text{pri},t}$ and $\boldsymbol{\sigma}^{\text{pri},t^2}$ are conditioned on the hidden states h^{t-1} of RNN as follows:

$$[\boldsymbol{\mu}^{\text{pri},t}, \boldsymbol{\sigma}^{\text{pri},t^2}] = \varphi^{\text{pri}}(h^{t-1}; \mathbf{W}^{\text{pri}}), \quad (8)$$

where $\varphi^{\text{pri}}(\cdot)$ is a mapping function that maps hidden state to a prior distribution with weights \mathbf{W}^{pri} .

Generation. At time step t , the generation process aims to decode imputed trajectory or future prediction from latent variables. Similarly, we assume that the location (2D coordinates) of agents follows a bi-variate Gaussian distribution as $\mathbf{x}^t \sim \mathcal{N}(\boldsymbol{\mu}^t, \boldsymbol{\sigma}^t, \boldsymbol{\rho}^t)$, where $\boldsymbol{\mu}^t$ is the mean, $\boldsymbol{\sigma}^t$ is the standard deviation, and $\boldsymbol{\rho}^t$ is the correlation coefficient.

For imputation, the generating distribution is conditioned on \mathbf{z}^t and the previous hidden state h^{t-1} such that:

$$[\hat{\boldsymbol{\mu}}^t, \hat{\boldsymbol{\sigma}}^t, \hat{\boldsymbol{\rho}}^t] = \varphi_p^{\text{dec}}(\varphi_p^z(\mathbf{z}^t) \oplus h^{t-1}; \mathbf{W}_p^{\text{dec}}) \quad t \in [t_1, t_p] \quad (9)$$

Differently, apart from \mathbf{z}^t and the previous hidden state h^{t-1} , the predicting distribution is also conditioned on the latent variables \mathbf{z}^{t_p} of the last observed time step such that:

$$[\hat{\boldsymbol{\mu}}^t, \hat{\boldsymbol{\sigma}}^t, \hat{\boldsymbol{\rho}}^t] = \varphi_f^{\text{dec}}(\varphi_f^z(\mathbf{z}^t \oplus \mathbf{z}^{t_p}) \oplus h^{t-1}; \mathbf{W}_f^{\text{dec}}) \quad t \in [t_{p+1}, t_f] \quad (10)$$

where $\varphi^{\text{dec}}(\cdot)$ is a decoding function with weights \mathbf{W}^{dec} , and $\varphi^z(\cdot)$ is feature extractor of \mathbf{z}^t . In the prediction decoder, we first concatenate latent variable \mathbf{z}^t to \mathbf{z}^{t_p} , and then extract the joint features. Intuitively, the feature information encoded by the imputation stream is also considered when decoding the predicting distribution.

Temporal Decay. In order to extract temporal features of missing patterns in observations, here we first introduce a temporal lag δ_i^t that indicates the relative distance between the last observable time step and the current time step t of agent i . The temporal lag is calculated as follows:

$$\delta_i^t = \begin{cases} t - (t-1) + \delta_i^{t-1} & \text{if } t > 1 \text{ and } \mathbf{m}_i^t = 0 \\ t - (t-1) & \text{if } t > 1 \text{ and } \mathbf{m}_i^t = 1 \\ 0 & \text{if } t = 1 \end{cases} \quad (11)$$

Concatenate temporal lags δ_i^t of all the agents at time step t , we can obtain the temporal lag vector $\boldsymbol{\delta}^t$. Then, the temporal decay vector $\boldsymbol{\Delta}^t$ is calculated as follows:

$$\boldsymbol{\Delta}^t = 1 / \exp(\max(0, \mathbf{W}_\delta \boldsymbol{\delta}^t + \mathbf{b}_\delta)), \quad (12)$$

where \mathbf{W}_δ and \mathbf{b}_δ are learnable parameters and bias. The insights behind this design lie in several points. In sequential modeling, if a variable has been missing for a while, its influence from the input will gradually decrease over time.

Since the temporal lag δ^t represents the distance from the last observation to the current time step, the temporal lag and temporal decay should be negatively correlated. Therefore, we chose a negative exponential function to ensure that the temporal decay decreases monotonically within a reasonable range of 0 and 1. Note that the decay vector is only calculated and applied for the past incomplete trajectory.

Recurrence. To capture more complex patterns from missing data, simply relying on temporal decay vectors may not be sufficient. Therefore, we propose to enhance the information obtained from the temporal decay vectors by element-wise multiplying them with the hidden states during the recurrence updating process as follows:

$$h^{t-1'} = \Delta^t \odot h^{t-1}, \quad (13)$$

Intuitively, this operation can decay the extracted features rather than temporal decayed values. Finally, for the imputation stream, the RNN is updated as follows:

$$h^t = \text{RNN} \left((\mathbf{F}_G^t \oplus \varphi_p^z(z^t)), h^{t-1'} \right), \quad (14)$$

$$t \in [t_1, t_p]$$

while for the prediction stream, the RNN is updated as:

$$h^t = \text{RNN} \left((\mathbf{F}_G^t \oplus \varphi_f^z(z^t \oplus z^{t_p})), h^{t-1} \right), \quad (15)$$

$$t \in [t_{p+1}, t_f]$$

where we also concatenate latent variable z^t to z^{t_p} when updating the hidden states in the prediction stream.

Inference. At each time step, the approximate posterior distribution of latent variables follows the distribution as:

$$\begin{aligned} z^t | \mathbf{x}^t &\sim \mathcal{N}(\boldsymbol{\mu}^{\text{enc},t}, \boldsymbol{\sigma}^{\text{enc},t^2}) & t \in [t_1, t_p] \\ z^t | \mathbf{y}^t &\sim \mathcal{N}(\boldsymbol{\mu}^{\text{enc},t}, \boldsymbol{\sigma}^{\text{enc},t^2}) & t \in [t_{p+1}, t_f] \end{aligned} \quad (16)$$

The approximate posterior distribution is conditioned on graph representation and hidden states of RNN as follows:

$$[\boldsymbol{\mu}^{\text{enc},t}, \boldsymbol{\sigma}^{\text{enc},t^2}] = \varphi^{\text{enc}} \left((\mathbf{F}_G^t \oplus h^{t-1}); \mathbf{W}^{\text{enc}} \right), \quad (17)$$

where $\varphi^{\text{enc}}(\cdot)$ is the encoding function with weights \mathbf{W}^{enc} .

Loss Function. The loss function formed by Eq. (6) of our proposed model consists of two parts: \mathcal{L}_{imp} for imputation and \mathcal{L}_{pre} for prediction, which are defined as follows:

$$\begin{aligned} \mathcal{L}_{\text{imp}} = & - \sum_{t=t_1}^{t_p} \log(\mathbb{P}(\mathbf{x}^t | \hat{\boldsymbol{\mu}}^t, \hat{\boldsymbol{\sigma}}^t, \hat{\boldsymbol{\rho}}^t)) \\ & + \lambda_1 \text{KL}(\mathcal{N}(\boldsymbol{\mu}^{\text{enc},t}, \boldsymbol{\sigma}^{\text{enc},t^2}) || \mathcal{N}(\boldsymbol{\mu}^{\text{pri},t}, \boldsymbol{\sigma}^{\text{pri},t^2})) \end{aligned} \quad (18)$$

$$\begin{aligned} \mathcal{L}_{\text{pre}} = & - \sum_{t=t_{p+1}}^{t_f} \log(\mathbb{P}(\mathbf{y}^t | \hat{\boldsymbol{\mu}}^t, \hat{\boldsymbol{\sigma}}^t, \hat{\boldsymbol{\rho}}^t)) \\ & + \lambda_2 \text{KL}(\mathcal{N}(\boldsymbol{\mu}^{\text{enc},t}, \boldsymbol{\sigma}^{\text{enc},t^2}) || \mathcal{N}(\boldsymbol{\mu}^{\text{pri},t}, \boldsymbol{\sigma}^{\text{pri},t^2})) \end{aligned} \quad (19)$$

The overall loss is defined in the following manner:

$$\mathcal{L} = \mathcal{L}_{\text{imp}} + \lambda_3 \mathcal{L}_{\text{pre}}, \quad (20)$$

where $\{\lambda_1, \lambda_2, \lambda_3\}$ are weighting factors, and we set them as 1 in our model. Note that the loss is calculated over all agents in each trajectory, and our model is trained end-to-end for the joint problem of imputation and prediction.

5. Experiments

5.1. Benchmarks and Setup

Datasets. We curate three benchmarks for the joint problem of Trajectory Imputation and Prediction (TIP), and we name these three datasets with *-TIP* as the suffix. More details are presented in the supplementary material.

Basketball-TIP: We construct Basketball-TIP using NBA dataset [78], which consists of 104,003 training sequences and 13,464 testing sequences. We design two strategies, “circle mode” and “camera mode”, to replicate the realistic appearance and disappearance of players. We established six scenarios by defining three different radii r (feet) or three different angles θ (degree) for better evaluation.

Football-TIP: Football-TIP is established from NFL Football Dataset¹, which contains 10,780 training sequences and 2,492 testing sequences. Similar to Basketball-TIP, we curate six scenarios by defining three different radii r (yard) and three different angles θ (degree) for evaluation.

Vehicle-TIP: We use the Omni-MOT dataset [56] to simulate incomplete observations. Three difficulty levels are associated with the camera viewpoints: Easy, Ordinary, and Hard. With the Easy viewpoint, we have 29,239 training sequences and 6,419 testing sequences. With the Ordinary viewpoint, we have 33,831 training sequences and 7,427 testing sequences. Lastly, with the Hard viewpoint, we have 31,714 training sequences and 6,962 testing sequences.

Evaluation Protocol. For Basketball-TIP and Football-TIP, we observe the first 40 frames and predict the next 10 frames. For Vehicle-TIP, we observe the first 60 frames and predict the next 30 frames. The observations are incomplete and have corresponding masks to indicate the visibility.

Metrics. To assess the imputation, we determine the average L_2 distance ($\mathbf{I}-L_2$) between each agent’s imputed trajectory and its corresponding ground truth over time. Similarly, for prediction evaluation, we calculate the average L_2

¹<https://github.com/nfl-football-ops/Big-Data-Bowl>

Datasets	Methods	$r = 3$ ft.		$r = 5$ ft.		$r = 7$ ft.		$\theta = 10^\circ$		$\theta = 20^\circ$		$\theta = 30^\circ$		
		I- L_2	P- L_2	I- L_2	P- L_2	I- L_2	P- L_2	I- L_2	P- L_2	I- L_2	P- L_2	I- L_2	P- L_2	
Basketball-TIP (In Feet)	Mean	9.07	—	9.53	—	9.51	—	8.83	—	8.64	—	8.47	—	
	Median	9.32	—	9.82	—	9.81	—	9.16	—	8.96	—	8.75	—	
	GMAT [78]	7.36	—	6.89	—	6.73	—	6.42	—	5.99	—	6.01	—	
	NAOMI [36]	7.68	—	7.08	—	7.04	—	6.33	—	6.11	—	5.91	—	
	Linear Fit	14.90	21.14	14.06	20.36	13.58	18.94	12.78	21.01	11.47	16.38	11.26	14.40	
	Vanilla LSTM [24]	7.33	20.07	6.73	14.91	6.51	10.07	6.28	9.34	6.01	7.52	5.67	6.10	
	Vanilla VRNN [14]	7.43	12.26	6.90	11.38	6.68	10.07	6.38	8.49	6.09	7.47	5.92	7.36	
	INAM [47]	7.35	8.93	6.93	8.24	6.80	7.68	6.50	7.32	6.13	7.10	5.92	6.96	
GC-VRNN (Ours)	7.03	7.50	6.41	6.80	6.24	5.93	5.86	6.29	5.56	4.74	5.39	4.28		
Football-TIP (In Yards)		$r = 2$ yd.		$r = 4$ yd.		$r = 6$ yd.		$\theta = 2^\circ$		$\theta = 6^\circ$		$\theta = 8^\circ$		
		I- L_2	P- L_2	I- L_2	P- L_2	I- L_2	P- L_2	I- L_2	P- L_2	I- L_2	P- L_2	I- L_2	P- L_2	
		Mean	8.94	—	9.28	—	9.69	—	8.74	—	9.12	—	9.27	—
		Median	8.99	—	9.39	—	9.86	—	8.80	—	9.23	—	9.39	—
		GMAT [78]	4.63	—	5.37	—	6.44	—	7.37	—	6.98	—	6.92	—
		NAOMI [36]	4.48	—	4.95	—	5.83	—	7.21	—	6.82	—	6.70	—
		Linear Fit	7.18	7.58	7.01	6.97	7.08	9.88	8.48	9.77	7.17	8.40	7.12	8.04
		Vanilla LSTM [24]	5.26	6.98	5.96	5.13	6.47	6.83	7.33	10.21	7.85	7.90	7.75	7.88
		Vanilla VRNN [14]	4.61	5.63	5.31	5.48	6.29	5.94	6.70	8.27	6.38	6.98	6.21	6.72
		INAM [47]	4.32	6.01	4.94	5.52	6.11	6.34	7.19	8.31	6.85	7.33	6.62	7.26
		GC-VRNN (Ours)	3.95	4.50	4.68	4.42	5.19	4.66	5.54	7.58	5.37	5.88	5.42	5.71

Table 1. Quantitative results on datasets Basketball-TIP (in feet) and Football-TIP (in yards). Each dataset comprises six scenarios derived from two mode settings with different radius r and angel θ . The best results are highlighted in bold.

Dataset	Methods	Easy		Ordinary		Hard	
		I- L_2	P- L_2	I- L_2	P- L_2	I- L_2	P- L_2
Vehicle-TIP (In Pixels)	Mean	337.38	—	282.24	—	319.68	—
	Median	336.94	—	281.58	—	318.82	—
	Linear Fit	100.53	139.86	86.69	97.24	106.43	113.61
	Vanilla LSTM [24]	83.50	125.05	75.23	82.76	87.59	91.61
	Vanilla VRNN [14]	88.36	103.21	70.89	73.54	95.66	104.34
	GC-VRNN (Ours)	65.48	72.44	58.36	62.03	74.28	78.12

Table 2. Quantitative results (in pixels) on dataset Vehicle-TIP with three scenarios. The best results are highlighted in bold.

ID	GCL			$r = 3$ ft.		$r = 7$ ft.		$\theta = 10^\circ$		$\theta = 30^\circ$	
	ST	DL	EC	I- L_2	P- L_2	I- L_2	P- L_2	I- L_2	P- L_2	I- L_2	P- L_2
1	✓			7.24	9.46	7.16	9.01	6.20	7.30	5.78	5.28
2		✓		7.16	9.44	6.74	8.93	6.15	7.27	5.70	5.19
3	✓	✓		7.11	9.33	6.55	8.54	5.98	7.18	5.51	5.08
4	✓		✓	7.10	9.09	6.30	7.08	5.90	6.96	5.41	4.90
5		✓	✓	7.07	8.10	6.26	6.04	5.87	6.32	5.88	6.09
Ours	✓	✓	✓	7.03	7.50	6.24	5.93	5.86	6.29	5.39	4.28

Table 3. Component study of three GCLs in MS-GNN. ST denotes the Static Topology GCL, DL represents the Dynamic Learnable GCL, and EC represents the Edge Conditioned GCL.

distance ($P-L_2$) between each agent’s predicted trajectory and its corresponding ground truth over time.

Baselines. For both trajectory imputation and prediction, we choose the following methods: Linear Fit, Vanilla LSTM [24], Vanilla VRNN [14], INAM [47]. We also implement Mean, Median, GMAT [78], and NAOMI [36] to compare the trajectory imputation performance.

Implementation Details. In MS-GNN, we stack 3 static topology GCLs of Eq. (2) and 1 dynamical learnable GCL

of Eq. (3) for encoding both observed trajectory and future trajectory. The edge-conditioned GCL is only employed for observed trajectory and $\varphi_{ec}(\cdot)$ is a Conv2D block that includes 2 Conv2D layers with kernel size as 1, and 1 average pooling layers operating at channel level. $\varphi^{pri}(\cdot)$, $\varphi^{enc}(\cdot)$, $\varphi^{dec}(\cdot)$, and $\varphi^z(\cdot)$ are all implemented by MLPs. We set node feature dimension D in Eq. (1) and all three GCLs as 16. The RNN dimension is set as 256, and the latent variables dimension is set as 64. The experiments are conducted using PyTorch [46] on the Nvidia A100 GPU. The model is trained for 200 epochs, with a batch size of 64, utilizing the Adam optimizer [16] with an initial learning rate of 0.001, which decayed by 0.9 for every 20 epochs.

5.2. Quantitative Results

Tab. 1 shows the quantitative results on Basketball-TIP and Football-TIP with six scenarios. In all six scenarios, our method can achieve better performance compared to other baselines, regardless of whether it pertains to the imputation or prediction task. In particular, we can observe that the improvement on the prediction task is much more significant than that on the imputation task. Unlike other baselines, valuable information is promoted to be shared via temporal flow in our framework, the imputation task can help the prediction task to a certain extent. Sec. 5.3 empirically explores the connection between these two tasks.

Tab. 2 shows the quantitative results on Vehicle-TIP with three scenarios. Note that baselines [36, 47, 78] are only applicable to the sequences with a fixed number of agents,

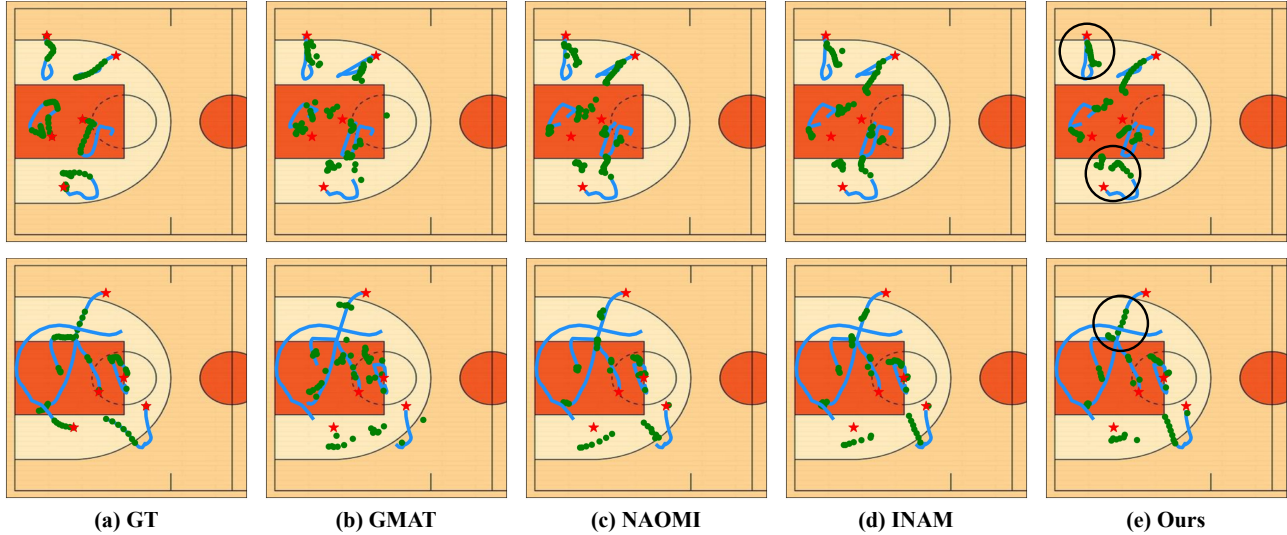


Figure 5. Visualizations of imputed results on Basketball-TIP ($\theta = 30^\circ$). The red star denotes the starting point, the blue line represents the visible observation, and the green point represents the missing point. Note that we only plot five defenders for brevity here.

Variants	$r = 3$ ft.		$r = 7$ ft.		$\theta = 10^\circ$		$\theta = 30^\circ$	
	I- L_2	P- L_2	I- L_2	P- L_2	I- L_2	P- L_2	I- L_2	P- L_2
w/ IMP	7.21	28.74	6.58	30.06	5.94	31.84	5.56	31.57
w/ PRE	29.15	7.78	29.07	6.48	30.61	6.83	32.35	4.76
wo/ CON	7.11	16.38	6.40	13.32	5.96	15.05	5.48	13.96
wo/ TD	7.25	7.70	6.37	6.26	5.98	6.61	5.51	4.52
Ours	7.03	7.50	6.24	5.93	5.86	6.29	5.39	4.28

Table 4. Ablation study of the temporal decay module and the connection between the imputation and prediction stream.

while the numbers of vehicles vary in each sequence in Vehicle-TIP. Nevertheless, our method greatly outperforms other baselines, validating that our method can effectively tackle this problem in diverse multi-agent domains.

5.3. Ablation Study

Three GCLs. Tab. 3 shows the results of variants with different combinations of GCLs. It can be observed that each GCL contributes to the final performance of our GC-VRNN. Specifically, we can see that the edge conditioned GCL provides a greater relative improvement to the accuracy of the model than the ST and DL GCLs. This validates the effectiveness of our designed EC GCL in extracting spatial missing patterns from incomplete observations.

Temporal Decay. The TD module is designed to decipher the temporal missing patterns of incomplete observations. To study the functionality of the TD module, we remove this module for comparison, which we refer to as “wo/ TD” in Tab. 4. It can be observed that the absence of the TD module leads to a notable decline in the performance of both tasks, confirming the effectiveness of modeling temporal missing patterns from incomplete observations.

Connection Between Two Streams. We investigate the benefits of connecting the imputation and the prediction

task. We conduct the following experiments: “w/ IMP” means we only make imputations, and “w/ PRE” means we only make predictions. We also cut off the connection by introducing two different RNNs for imputation and prediction separately, which we refer to as “wo/ CON”. It can be observed from Tab. 4 that considering these two tasks simultaneously can boost the performance of both tasks, especially for the prediction task. It validates the necessity of considering these two tasks in a unified framework.

5.4. Qualitative Results

In Fig. 5, we present visual results of imputation on Basketball-TIP ($\theta = 30^\circ$). The results demonstrate that our GC-VRNN produces more precise imputations than other baselines, thus confirming the superiority of our approach. Additional experimental outcomes, including visualizations, are available in the supplementary material.

6. Conclusion

Our study highlights a prevalent issue in the trajectory prediction literature, which assumes complete agent observations. We introduce a new avenue of research by jointly learning trajectory imputation and prediction. We propose a novel GC-VRNN method that uncovers spatio-temporal missing patterns and handles both tasks in a unified framework. Through experiments, we demonstrate the superiority of our designs and the benefits of simultaneously learning these tasks. To further research in this domain, we curate and benchmark three practical datasets, *Basketball-TIP*, *Football-TIP*, and *Vehicle-TIP*. As far as we know, our study is the first to bridge the gap in benchmarks and techniques for this joint problem.

References

- [1] Edgar Acuna and Caroline Rodriguez. The treatment of missing values and its effect on classifier accuracy. In *Classification, Clustering, and Data Mining Applications*, pages 639–647, 2004. 3
- [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. 2
- [3] Javad Amirian, Jean-Bernard Hayet, and Julien Pettré. Social ways: Learning multi-modal distributions of pedestrian trajectories with GANs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 2964–2972, 2019. 2
- [4] Craig F Ansley and Robert Kohn. On the estimation of arima models with missing values. *Lecture Notes in Statistics*, page 9. 3
- [5] Inhwan Bae, Jin-Hwi Park, and Hae-Gon Jeon. Learning pedestrian group representations for multi-modal trajectory prediction. In *Proceedings of the European Conference on Computer Vision*, pages 270–289, 2022. 1, 2
- [6] Mohammadhossein Bahari, Saeed Saadatnejad, Ahmad Rahimi, Mohammad Shaverdikondori, Amir Hossein Shahidzadeh, Seyed-Mohsen Moosavi-Dezfooli, and Alexandre Alahi. Vehicle trajectory prediction works, but not everywhere. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 17123–17133, 2022. 2
- [7] Lorenzo Beretta and Alessandro Santaniello. Nearest neighbor imputation algorithms: a critical evaluation. *BMC Medical Informatics and Decision Making*, 16(3):197–208, 2016. 3
- [8] Wei Cao, Dong Wang, Jian Li, Hao Zhou, Yitan Li, and Lei Li. Brits: bidirectional recurrent imputation for time series. In *Proceedings of the Advances in Neural Information Processing Systems*, pages 6776–6786, 2018. 3
- [9] Yulong Cao, Chaowei Xiao, Anima Anandkumar, Danfei Xu, and Marco Pavone. Advdo: Realistic adversarial attacks for trajectory prediction. In *Proceedings of the European Conference on Computer Vision*, 2022. 2
- [10] Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent neural networks for multivariate time series with missing values. *Scientific Reports*, 8(1):1–12, 2018. 3
- [11] Guangyi Chen, Junlong Li, Jiwen Lu, and Jie Zhou. Human trajectory prediction via counterfactual analysis. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9824–9833, 2021. 2
- [12] Yuxiao Chen, Boris Ivanovic, and Marco Pavone. Scept: Scene-consistent, policy-based trajectory predictions for planning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 17103–17112, 2022. 2
- [13] Luigi Filippo Chiara, Pasquale Coscia, Sourav Das, Simone Calderara, Rita Cucchiara, and Lamberto Ballan. Goal-driven self-attentive recurrent networks for trajectory prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2518–2527, 2022. 2
- [14] Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron Courville, and Yoshua Bengio. A recurrent latent variable model for sequential data. In *Proceedings of the Advances in Neural Information Processing Systems*, pages 2980–2988, 2015. 5, 7
- [15] Andrea Cini, Ivan Marisca, and Cesare Alippi. Multivariate time series imputation by graph neural networks. In *Proceedings of the International Conference on Learning Representations*, 2021. 2
- [16] P. Kingma Diederik and Ba Jimmy. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations*, 2015. 7
- [17] William Fedus, Ian Goodfellow, and Andrew M Dai. Maskgan: Better text generation via filling in the .. In *Proceedings of the International Conference on Learning Representations*, 2018. 2
- [18] Vincent Fortuin, Dmitry Baranchuk, Gunnar Raetsch, and Stephan Mandt. Gp-vae: Deep probabilistic time series imputation. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pages 1651–1661, 2020. 3
- [19] Zoubin Ghahramani and Michael Jordan. Supervised learning from incomplete data via an em approach. In *Proceedings of the Advances in Neural Information Processing Systems*, volume 6, 1993. 3
- [20] Harshayu Girase, Haiming Gang, Srikanth Malla, Jiachen Li, Akira Kanehara, Karttikeya Mangalam, and Chiho Choi. Loki: Long term and key intentions for trajectory prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9803–9812, 2021. 1
- [21] Francesco Giuliani, Irtiza Hasan, Marco Cristani, and Fabio Galasso. Transformer networks for trajectory forecasting. In *Proceedings of the IEEE International Conference on Pattern Recognition*, pages 10335–10342, 2020. 2
- [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. 2
- [23] Christopher Hazard, Akshay Bhagat, Balarama Raju Bud-dharaju, Zhongtao Liu, Yunming Shao, Lu Lu, Sammy Omari, and Henggang Cui. Importance is in your attention: agent importance prediction for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2532–2535, 2022. 1
- [24] Sepp Hochreiter and Jurgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. 7
- [25] Shengchao Hu, Li Chen, Penghao Wu, Hongyang Li, Junchi Yan, and Dacheng Tao. St-p3: End-to-end vision-based autonomous driving via spatial-temporal feature learning. In *Proceedings of the European Conference on Computer Vision*, pages 533–549, 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 Conference on Computer Vision and Pattern Recognition*, pages 6319–6328, 2020. 2

- [27] Boris Ivanovic, James Harrison, and Marco Pavone. Expanding the deployment envelope of behavior prediction via adaptive meta-learning. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2023. 2
- [28] Boris Ivanovic and Marco Pavone. The trajectron: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2375–2384, 2019. 2
- [29] Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific Data*, 3(1):1–9, 2016. 2
- [30] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *Proceedings of the International Conference on Learning Representations*, 2017. 4
- [31] Vineet Kosaraju, Amir Sadeghian, Roberto Martín-Martín, Ian Reid, Hamid Rezatofighi, and Silvio Savarese. Socialbigat: Multimodal trajectory forecasting using bicycle-gan and graph attention networks. In *Proceedings of the Advances in Neural Information Processing Systems*, pages 137–146, 2019. 2
- [32] Parth Kothari, Brian Siffringer, and Alexandre Alahi. Interpretable social anchors for human trajectory forecasting in crowds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 15556–15566, 2021. 2
- [33] Jiachen Li, Hengbo Ma, and Masayoshi Tomizuka. Conditional generative neural system for probabilistic trajectory prediction. *arXiv preprint arXiv:1905.01631*, 2019. 2
- [34] Lihuan Li, Maurice Pagnucco, and Yang Song. Graph-based spatial transformer with memory replay for multi-future pedestrian trajectory prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2231–2241, 2022. 1, 2
- [35] Zachary C Lipton, David Kale, and Randall Wetzell. Directly modeling missing data in sequences with rnns: Improved classification of clinical time series. In *Machine Learning for Healthcare Conference*, pages 253–270, 2016. 3
- [36] Yukai Liu, Rose Yu, Stephan Zheng, Eric Zhan, and Yisong Yue. Naomi: Non-autoregressive multiresolution sequence imputation. 32, 2019. 2, 3, 7
- [37] Yonghong Luo, Xiangrui Cai, Ying Zhang, Jun Xu, and Xiaojie Yuan. Multivariate time series imputation with generative adversarial networks. In *Proceedings of the Advances in Neural Information Processing Systems*, pages 1603–1614, 2018. 2, 3
- [38] Hengbo Ma, Yaofeng Sun, Jiachen Li, Masayoshi Tomizuka, and Chiho Choi. Continual multi-agent interaction behavior prediction with conditional generative memory. *IEEE Robotics and Automation Letters*, 6(4):8410–8417, 2021. 1
- [39] Osama Makansi, Özgün Cicek, Yassine Marrakchi, and Thomas Brox. On exposing the challenging long tail in future prediction of traffic actors. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 13127–13137, 2021. 2
- [40] Karttikeya Mangalam, Yang An, Harshayu Girase, and Jitendra Malik. From goals, waypoints & paths to long term human trajectory forecasting. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 15233–15242, 2021. 2
- [41] Karttikeya Mangalam, Harshayu Girase, Shreyas Agarwal, Kuan Hui Lee, Ehsan Adeli, Jitendra Malik, and Adrien Gaidon. It is not the journey but the destination: End-point conditioned trajectory prediction. In *Proceedings of the European Conference on Computer Vision*, pages 759–776, 2020. 2
- [42] Xiaoye Miao, Yangyang Wu, Jun Wang, Yunjun Gao, Xudong Mao, and Jianwei Yin. Generative semi-supervised learning for multivariate time series imputation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 8983–8991, 2021. 3
- [43] Abdulllah Mohamed, Kun Qian, Mohamed Elhoseiny, and Christian Claudel. Social-STGCNN: A social spatiotemporal graph convolutional neural network for human trajectory prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 14424–14432, 2020. 2
- [44] Alessio Monti, Angelo Porrello, Simone Calderara, Pasquale Coscia, Lamberto Ballan, and Rita Cucchiara. How many observations are enough? Knowledge distillation for trajectory forecasting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6553–6562, 2022. 2
- [45] Fulufhelo V Nelwamondo, Shakir Mohamed, and Tshilidzi Marwala. Missing data: A comparison of neural network and expectation maximization techniques. *Current Science*, pages 1514–1521, 2007. 3
- [46] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Proceedings of the Advances in Neural Information Processing Systems*, 2019. 7
- [47] Mengshi Qi, Jie Qin, Yu Wu, and Yi Yang. Imitative non-autoregressive modeling for trajectory forecasting and imputation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12736–12745, 2020. 3, 7
- [48] Amir Sadeghian, Vineet Kosaraju, Ali Sadeghian, Noriaki Hirose, Hamid Rezatofighi, and Silvio Savarese. SoPhic: An attentive GAN for predicting paths compliant to social and physical constraints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1349–1358, 2019. 2
- [49] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *Proceedings of the European Conference on Computer Vision*, pages 683–700, 2020. 2
- [50] Omer Berat Sezer, Mehmet Ugur Gudelek, and Ahmet Murat Ozbayoglu. Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. *Applied soft computing*, 90:106181, 2020. 2

- [51] Liushuai Shi, Le Wang, Chengjiang Long, Sanping Zhou, Mo Zhou, Zhenxing Niu, and Gang Hua. Sgcn: Sparse graph convolution network for pedestrian trajectory prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8994–9003, 2021. 2
- [52] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 12026–12035, 2019. 4
- [53] Satya Narayan Shukla and Benjamin Marlin. Multi-time attention networks for irregularly sampled time series. In *Proceedings of the International Conference on Learning Representations*, 2020. 2
- [54] Ikaro Silva, George Moody, Daniel J Scott, Leo A Celi, and Roger G Mark. Predicting in-hospital mortality of icu patients: the physionet/computing in cardiology challenge 2012. In *Computing in Cardiology*, pages 245–248, 2012. 2
- [55] Jianhua Sun, Qinhong Jiang, and Cewu Lu. Recursive social behavior graph for trajectory prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 660–669, 2020. 2
- [56] ShiJie Sun, Naveed Akhtar, XiangYu Song, HuanSheng Song, Ajmal Mian, and Mubarak Shah. Simultaneous detection and tracking with motion modelling for multiple object tracking. In *Proceedings of the European Conference on Computer Vision*, pages 626–643, 2020. 6
- [57] Supasorn Suwajanakorn, Steven M Seitz, and Ira Kemelmacher-Shlizerman. Synthesizing obama: learning lip sync from audio. *ACM Transactions on Graphics*, 36(4):1–13, 2017. 1
- [58] Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. Csd: Conditional score-based diffusion models for probabilistic time series imputation. In *Proceedings of the Advances in Neural Information Processing Systems*, volume 34, pages 24804–24816, 2021. 3
- [59] Sarah Taylor, Taehwan Kim, Yisong Yue, Moshe Mahler, James Krahe, Anastasio Garcia Rodriguez, Jessica Hodgins, and Iain Matthews. A deep learning approach for generalized speech animation. *ACM Transactions on Graphics*, 36(4):1–11, 2017. 1
- [60] Hung Tran, Vuong Le, and Truyen Tran. Goal-driven long-term trajectory prediction. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, pages 796–805, 2021. 2
- [61] Olga Troyanskaya, Michael Cantor, Gavin Sherlock, Pat Brown, Trevor Hastie, Robert Tibshirani, David Botstein, and Russ B Altman. Missing value estimation methods for dna microarrays. *Bioinformatics*, 17(6):520–525, 2001. 3
- [62] Li-Wu Tsao, Yan-Kai Wang, Hao-Siang Lin, Hong-Han Shuai, Lai-Kuan Wong, and Wen-Huang Cheng. Social-ssl: Self-supervised cross-sequence representation learning based on transformers for multi-agent trajectory prediction. In *Proceedings of the European Conference on Computer Vision*, pages 234–250, 2022. 1, 2
- [63] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the Advances in Neural Information Processing Systems*, pages 5998–6008, 2017. 2
- [64] Anirudh Vemula, Katharina Muelling, and Jean Oh. Social attention: Modeling attention in human crowds. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1–7, 2018. 2
- [65] Jingke Wang, Tengju Ye, Ziqing Gu, and Junbo Chen. Ltp: Lane-based trajectory prediction for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 17134–17142, 2022. 1
- [66] Xinshuo Weng, Boris Ivanovic, Kris Kitani, and Marco Pavone. Whose track is it anyway? improving robustness to tracking errors with affinity-based trajectory prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6573–6582, 2022. 2
- [67] 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 Conference on Computer Vision and Pattern Recognition*, pages 6498–6507, 2022. 1, 2
- [68] Chenxin Xu, Weibo Mao, Wenjun Zhang, and Siheng Chen. Remember intentions: Retrospective-memory-based trajectory prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6488–6497, 2022. 1, 2
- [69] Pei Xu, Jean-Bernard Hayet, and Ioannis Karamouzas. Socialvae: Human trajectory prediction using timewise latents. In *Proceedings of the European Conference on Computer Vision*, 2022. 1, 2
- [70] Yi Xu, Dongchun Ren, Mingxia Li, Yuehai Chen, Mingyu Fan, and Huaxia Xia. Robust trajectory prediction of multiple interacting pedestrians via incremental active learning. In *Proceedings of the International Conference on Neural Information Processing*, pages 141–150, 2021. 2
- [71] Yi Xu, Dongchun Ren, Mingxia Li, Yuehai Chen, Mingyu Fan, and Huaxia Xia. Tra2tra: Trajectory-to-trajectory prediction with a global social spatial-temporal attentive neural network. *IEEE Robotics and Automation Letters*, 6(2):1574–1581, 2021. 2
- [72] Yi Xu, Lichen Wang, Yizhou Wang, and Yun Fu. Adaptive trajectory prediction via transferable gnn. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6520–6531, 2022. 2
- [73] Joonyoung Yi, Juhyuk Lee, Kwang Joon Kim, Sung Ju Hwang, and Eunho Yang. Why not to use zero imputation? Correcting sparsity bias in training neural networks. In *Proceedings of the International Conference on Learning Representations*, 2019. 2
- [74] Jinsung Yoon, James Jordon, and Mihaela Schaar. Gain: Missing data imputation using generative adversarial nets. In *The International Conference on Machine Learning*, pages 5689–5698. PMLR, 2018. 2, 3
- [75] Jinsung Yoon, William R Zame, and Mihaela van der Schaar. Estimating missing data in temporal data streams using multi-directional recurrent neural networks. *IEEE Transactions on Biomedical Engineering*, 66(5):1477–1490, 2018. 3

- [76] Cunjun Yu, Xiao Ma, Jiawei Ren, Haiyu Zhao, and Shuai Yi. Spatio-temporal graph transformer networks for pedestrian trajectory prediction. In *Proceedings of the European Conference on Computer Vision*, pages 507–523, 2020. [2](#)
- [77] 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 International Conference on Computer Vision*, pages 9813–9823, 2021. [2](#)
- [78] Eric Zhan, Stephan Zheng, Yisong Yue, Long Sha, and Patrick Lucey. Generating multi-agent trajectories using programmatic weak supervision. In *Proceedings of the International Conference on Learning Representations*, 2018. [2](#), [3](#), [6](#), [7](#)
- [79] Pu Zhang, Wanli Ouyang, Pengfei Zhang, Jianru Xue, and Nanning Zheng. SR-LSTM: State refinement for LSTM towards pedestrian trajectory prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12085–12094, 2019. [2](#)
- [80] Qingzhao Zhang, Shengtuo Hu, Jiachen Sun, Qi Alfred Chen, and Z Morley Mao. On adversarial robustness of trajectory prediction for autonomous vehicles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 15159–15168, 2022. [2](#)
- [81] He Zhao and Richard P Wildes. Where are you heading? Dynamic trajectory prediction with expert goal examples. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7629–7638, 2021. [2](#)
- [82] Yu Zheng, Licia Capra, Ouri Wolfson, and Hai Yang. Urban computing: concepts, methodologies, and applications. *ACM Transactions on Intelligent Systems and Technology*, 5(3):1–55, 2014. [2](#)