STEMFold: Stochastic Temporal Manifold for Multi-Agent Interactions in the Presence of Hidden Agents

Hemant Kumawat HKUMAWAT6@GATECH.EDU
Biswadeep Chakraborty BISWADEEP@GATECH.EDU
Saibal Mukhopadhyay SAIBAL.MUKHOPADHYAY@ECE.GATECH.EDU
School of Electrical and Computer Engineering, Georgia Institute of Technology, GA, USA

Editors: A. Abate, K. Margellos, A. Papachristodoulou

Abstract

Learning accurate, data-driven predictive models for multiple interacting agents following unknown dynamics is crucial in many real-world physical and social systems. In many scenarios, dynamics prediction must be performed under incomplete observations, i.e., only a subset of agents are known and observable from a larger topological system while the behaviors of the unobserved agents and their interactions with the observed agents are not known. When only incomplete observations of a dynamical system are available, so that some states remain hidden, it is generally not possible to learn a closed-form model in these variables using either analytic or data-driven techniques. In this work, we propose STEMFold, a spatiotemporal attention-based generative model, to learn a stochastic manifold to predict the underlying unmeasured dynamics of the multi-agent system from observations of only visible agents. Our analytical results motivate STEMFold design using a spatiotemporal graph with time anchors to effectively map the observations of visible agents to a stochastic manifold with no prior information about interaction graph topology. We empirically evaluated our method on two simulations and two real-world datasets, where it outperformed existing networks in predicting complex multiagent interactions, even with many unobserved agents.

Supplementary: Supplementary with Analytical Proofs and Additional Results

Keywords: Unobservable Agents, Trajectory Prediction, and Incomplete Observations

1. Introduction

Understanding the unknown underlying dynamics governing a group of co-evolving agents and how they influence each other's behavior is a crucial task across various domains, including robotics (Mavrogiannis and Knepper (2020), Abbeel and Ng (2004)), social networks (Alahi et al. (2016a), Luber et al. (2010)), and transportation networks (Jahangiri and Rakha (2015), Wojtusiak et al. (2012)). It poses a challenge to uncover hidden relations and predict dynamics based on observed states, which is vital for downstream decision-making. An important task in discovering and understanding multi-agent dynamics is predicting the state of all agents over time (trajectory prediction). Deep learning techniques such as latent interaction graphs (Kipf et al. (2018), Alet et al. (2019)), attention-based methods for graphs (Vemula et al. (2017), Hoshen (2017), Kosaraju et al. (2019), Huang et al. (2021)), recurrent neural networks (Rubanova et al. (2019b), Zhan et al. (2019)), and neural message passing (Santoro et al. (2017a), Li et al. (2020)) have been developed to predict emergent behavioral patterns in multi-agent systems. All the prior works assume that the dynamical systems are fully observable, i.e. the number of agents in the system is known and the trajectories can be sparsely or continuously sampled as shown in Figure 1A. However, many applications deal with unobservable agents due to inherent restrictions on sensing and observation capabilities. Such ”Agent-Unobservable” systems will demonstrate a lower number of independent degrees of freedom compared to its true intrinsic dimension. Developing deep learning models that can predict the trajectory of multi-agent systems under the limited observability of agents continues to be a challenging task. Table 1 offers an in-depth analysis with previous studies in multiagent modeling.

Scenario	**Description of Problem**	**References**
Complete observability with *known interaction topology*	Multi-agent systems where all agents are observable at all times, with a known interaction topology	Watters et al. (2017)
Complete observability with *unknown interaction topology*	All agents are observable at all times; however, the interaction topology is not predefined and must be inferred from observational data.	Alahi et al. (2016b) Banijamali (2022) Graber and Schwing (2020) Kipf et al. (2018) Alet et al. (2019) van Steenkiste et al. (2018) Santoro et al. (2017b)
Complete observability with *Irregular sampling of observations*	All agents are observable but the observation events are sporadic or irregular, leading to temporal data sparsity.	Rubanova et al. (2019a) Zhu et al. (2021) Huang et al. (2020)Marisca et al. (2022) Sun et al. (2019)
Agent Unobervable: Only few agents observable with *sparse temporal sampling*	Not all agents are observable, with some never being observed, coupled with sparse temporal data collection.	(Ours)

Table 1: Systematic classification of observation scenarios in multi-agent systems.

In this paper, we present STEMFold, a multi-agent behavior modeling framework to learn a stochastic temporal manifold to predict the trajectory of multi-agent systems by utilizing a dynamic spatiotemporal graph attention mechanism specifically tailored for systems where only a *subset of agents is observable at any given time.* Our analytical findings demonstrate that constructing a spatiotemporal graph using visible nodes in a multi-agent system results in a superior manifold mapping of the observation space, leading to enhanced performance in predicting the trajectories of visible agents. Empirically, we demonstrate that our network is capable of learning meaningful representations for multi-agent systems, utilizing two simulated and two real-world datasets. Our model offers improved long-term prediction even when a substantial number of agents are unobservable in these diverse scenarios.

2. Spatial-Temporal Attention Model

2.1. Problem Description

We consider a multi-agent system with M homogeneous or heterogeneous agents, out of which only N agents could be observed (*Observable Agents*) at any time and the rest $(M - N)$ agents are unobserved (*Hidden Agents*). The number of agents could vary depending on the system and we assume that we do not know the total number of agents and hidden agents present in the system. We could only observe the spatial-temporal state sequences of the observable agents. We model the observable agents as a graph $G = \langle O, R \rangle$ where nodes $O = \{o_1, o_2, o_3, ...o_N\}$ represents the observed agents with $R = \{\langle i, j \rangle\}$ representing the interactions among them. We model the interactions among the agents as graph edges. These functional interactions among agents could be inferred from the physical proximity of the agents or the structure of the system they are placed in.

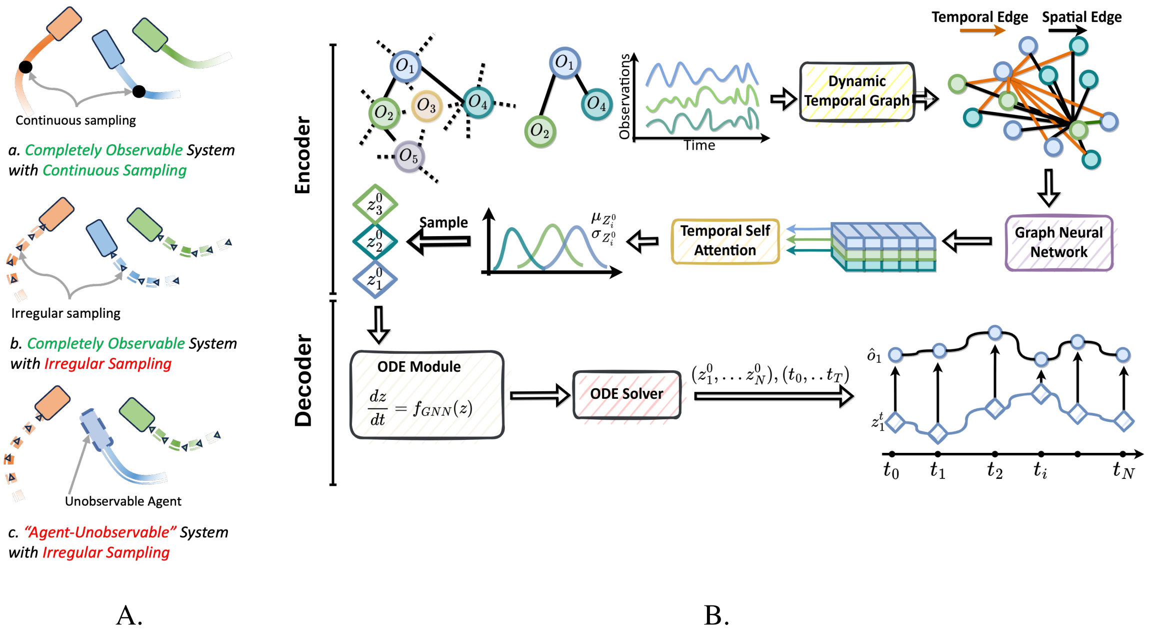

Figure 1: A) Problem landscape in prior works. 'a' & 'b' depict problems addressed in previous works, while 'c' illustrates the unique problem tackled in our work. B.) Model overview. Firstly, the encoder computes the initial latent states for edges and nodes based on the observed sequence of agent observations and adjacency matrix sequence. This computation occurs in two steps: Step 1 involves attention-based representation learning over the dynamic spatiotemporal graph. Step 2 focuses on sequence attention, to learn posterior over the initial latent state. Afterward, the neural ODE framework propagates the latent state through time, and subsequently, the decoder generates predicted observations for the agents.

We model the interactions $R = \{\langle i, j \rangle\}$ as a weighted adjacency matrix $A \in \mathbb{R}^{N \times N}$ with $a_{i,j} > 0$ representing an edge going from i^{th} node to the j^{th} with interaction strength given by the value of $a_{i,j}$. For each agent, we denote spatio-temporal sequences as $o_i = \{o_i^t\}$ where $t \in \{t_1, t_2, ...t_T\}$ and $o_i^t \ \in \mathbb{R}^D$ denotes the spatial feature of object i at time t. The observation sequences are only available for the observed agents and we have no contextual or state information about the hidden agents. We denote the the set of historical state sequence as $\mathcal{X}^H = o_i^{1:T_h}, i = 1, ..., N$ and we aim to estimate $p(\mathcal{X}^{T_{h+1}:T_{h+f}} | \mathcal{X}^{1:T_h}, R^{1:T_h})$ to forecast agent trajectories given historical observations up to $t = T_h$ where $T = T_h + T_f$ and T_f denotes the forecasting horizon.

2.2. Model Description

Our method *STEMFold* is designed to learn representations from spatiotemporal observations of multi-agent systems with interaction graphs sampled from a larger, unknown topological system. The model constructs a parameterized, stochastic latent manifold by aggregating temporal representations from multiple agent observations, each weighted according to node-specific attention coefficients. The overall framework is depicted in Figure 1 and it consists of three parts that are trained jointly. (1) An encoder module that maps the observations to the manifold and learns the initial latent point for all the nodes while taking into account the interactions among entities. (2)

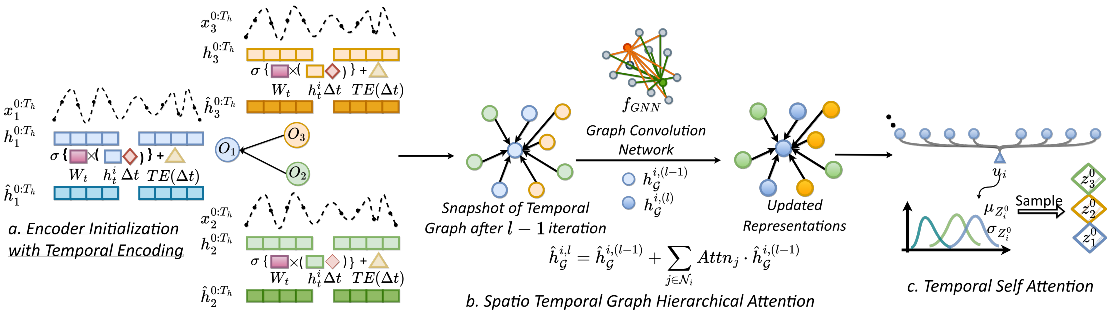
$x_3^{0:T_h}$
$h_3^{0:T_h}$
$\sigma\{\ \times(\ \ )\} +$
W_t h_t^i Δt $TE(\Delta t)$
$\hat{h}_3^{0:T_h}$
$x_1^{0:T_h}$
$h_1^{0:T_h}$
$\hat{h}_1^{0:T_h}$
O_1
O_3
O_2
$x_2^{0:T_h}$
$h_2^{0:T_h}$
$\hat{h}_2^{0:T_h}$
a. Encoder Initialization with Temporal Encoding
f_{GNN}
Graph Convolution Network
$h_{\mathcal{G}}^{i,(l-1)}$
$h_{\mathcal{G}}^{i,(l)}$
Snapshot of Temporal Graph after $l-1$ iteration
Updated Representations
$\hat{h}_{\mathcal{G}}^{i,l} = \hat{h}_{\mathcal{G}}^{i,(l-1)} + \sum_{j \in \mathcal{N}_i} Attn_j \cdot \hat{h}_{\mathcal{G}}^{i,(l-1)}$
b. Spatio Temporal Graph Hierarchical Attention
y_i
$\mu_{Z_i^0}$
$\sigma_{Z_i^0}$
Sample
z_3^0
z_2^0
z_1^0
c. Temporal Self Attention

Figure 2: Illustration of the spatiotemporal attention layer in action: On the left side, there's a spatiotemporal graph with each node having an associated time series. In the center(b), you can observe how this layer functions to update the target representation. Finally, the module is passed through the self-attention layer to get the initial latent distribution.

A generative neural-ode model characterized by ODE functions for latent states for nodes to learn the latent dynamics of the system. (3) A decoder that generates the node predictions for the visible agents conditioned on the latent state.

Dynamic spatiotemporal graph with Temporal Anchors The core component of STEMFold is the dynamic temporal graph that learns and propagates the structural temporal information from observed observations. Rather than developing an encoder to distill temporal features from the original subgraph (Watters et al. (2017)), our approach constructs a temporal graph derived directly from the agents' observations. A temporal node is instantiated for every i^{th} agent whenever an observation is made at time t, and we define a temporal relation, denoted by $r \in R\{\langle i, j \rangle\}$, between agents. Every i^{th} node in the graph is characterized by a unique feature vector, denoted as $o_{i,t} = [x_{i,t}, v_{i,t}]$, which is a concatenation of the agent's spatial location $(x_{i,t})$ and velocity $(v_{i,t})$. Each node is then assigned with time anchors $a_i = t_i - t_{0,i}$ where t_i represents the node's observation time. This calculated temporal position encapsulates the chronological information, allowing for the nuanced depiction of temporal relationships within the graph. The depiction of temporal relationships is further refined through the construction of edges, based on an edge matrix where each element represents the temporal disparity between two nodes, i and j, formalized as r$_{ij} = a_i - a_j$. The existence of an edge and its attributes are contingent upon this time difference, with an edge being formulated and assigned the value of the time difference if it is within a predefined threshold, the maximum allowable gap. Subsequently, we will denote this temporal graph as $\mathcal{G}$.

Stochastic Manifold with Temporal Graph Hierarchical Attention Given a certain set of trajectories of observable agents, there may be multiple different settings of hidden agents (e.g., different numbers, different states) that lead to the same observations of the observable agents leading to stochasticity in the prediction. This inherent stochasticity in prediction is tackled by employing a stochastic latent state model, designed to learn the distribution of possible agent configurations. The model, informed by observations and updated beliefs, generates a latent state that accurately encapsulates the specific system configuration at hand. Once the initial setup of these agents is determined, their trajectory progression becomes deterministic, characterized by a single modality. To effectively map the latent manifold within the spatiotemporal graph $\mathcal{G}$, we utilize a graph attention-based neural message passing technique. This method's core objective is to assimilate aggregated representations based on the observed data $\mathcal{X}^i_{1:T_h}$ of the i^{th} multi-agent and the observations of its neighboring agents $\mathcal{X}^{j}_{1:T_h}$, where $j \in \mathcal{N}(i)$. The learned representation for the i^{th} node at the l^{th} layer is denoted as $h^{i,(l)}_{\mathcal{G}}$. We initialize the representation encoding with temporal positional encoding $\mathbf{q^i}$ as: $h^{i,(0)}_{\mathcal{G}} = \sigma(W_{init}[o_{i,t} \| \Delta t_{start}]) + q^i(\Delta t_{start})$. Here, $\sigma(.)$ is a nonlinear activation function and $\|$ is a concatenation operation for tensors. This process is depicted in the left sketch of Figure 2 where this initialization process is shown for a sample graph with three visible nodes. We then update the initialized representations by spatial-temporal attention operations Huang et al. (2021) for each node using graph neural message passing. Similar to Vaswani et al. (2017), we define *query* as the token for which we need a new representation, a *key* as a feature for the source token, and the *value* as the representation or message of the token to be passed. The interaction representation message $\mathbf{Message}_{r\to s} \in \mathbb{R}^{d_h}$ from the s^{th} source node to the r^{th} receiver node is computed as:

$$\mathbf{Message}^{l-1}_{r\to s} = W_v \hat{h}^{s,(l-1)}_{\mathcal{G}}, \quad \hat{h}^{s,(l-1)}_{\mathcal{G}} = \sigma(W_t[h^{s,(l-1)}_{\mathcal{G}} \| \Delta t_{\text{start}}]) + q^i(\Delta t_{\text{start}}) \tag{1}$$

Here, W_v and W_t are linear transformation weight matrices. Next, we find the attention scores for the messages:

$$\mathbf{Attn}^{l-1}_{r\to s} = softmax\{(W_{key}\hat{h}^{s,(l-1)}_{\mathcal{G}})^T (W_{query} h^{r,(l-1)}_{\mathcal{G}}) \cdot \frac{1}{\sqrt{d}}\} \tag{2}$$

Then, all the temporal messages are aggregated to update the node-level context features:

$$h^{r,(l)}_{\mathcal{G}} = h^{r,(l-1)}_{\mathcal{G}} + \sum_{s\in\mathcal{N}_r} (\mathbf{Attn}^{l-1}_{r\to s} \cdot \mathbf{Message}^{l-1}_{r\to s}) \tag{3}$$

This is shown in Figure 2b, where the graph convolution network is used to update the $(l-1)^{th}$ layer's representations.

Loss function and Training The encoder, decoder, and generative model are trained together by maximizing the evidence lower bound (ELBO), as illustrated below where the first term is the prediction loss for visible nodes, and the second term is the KL divergence.

$$ELBO(\theta, \phi) = \mathbb{E}_{Z^0 \sim q_\phi(Z^0|\mathcal{X})}[\log p_\theta(\mathcal{X})] - \text{KL}[q_\theta(Z^0|\mathcal{X}) \| p(Z^0)] \tag{4}$$

2.3. Analytical Results

Let $G(V(t), E(t))$ be the graph with nodes $V(t)$ and edges $E(t)$ at time t. Let G' be a subgraph of G with observed nodes $x_1(t), x_2(t), \ldots, x_N(t)$. The temporal graph T' can be defined as a multiset of the states of graph G' at different time points, represented as: $T' = \{G'(t_1), G'(t_2), \ldots, G'(t_r)\}$ where each $G'(t_i)$ is a member of the multiset representing the state of graph G' at time t_i, and additional temporal edges are added between nodes in $G'(t_i)$ and $G'(t_{i+1})$ for all $i = 1, 2, \ldots, r-1$ to represent the temporal connections between the different states of graph G'. Here, a multiset is a generalized notion of a set that allows multiple instances of its elements. We first state the following two theorems:

Theorem 1: *The Fisher information of the embedding of the multiset X_i is greater than the Fisher information of the embedding of each individual element $x_i(t)$*

Intuitive Proof: (***For proof refer to Supp. Sec. 2.2 Theorem 1***): Fisher Information, denoted as $I(\theta)$ for a parameter θ, measures the expected amount of information that an observable random variable X carries about θ: $I(\theta) = \mathbb{E}\left[\left(\frac{\partial}{\partial\theta}\log f(X;\theta)\right)^2\right]$, where $f(X;\theta)$ is the probability density function of X. When computing Fisher Information for X_i, we account for the joint distribution of all $x_i(t)$ within X_i. This joint distribution inherently includes correlations among $x_i(t)$. Since Fisher Information is additive for independent samples, the information from a multiset is at least the sum of the information from individual elements, assuming independence. However, when elements are not independent, the correlations contribute additional information. This is because the joint variability and the relationships among elements provide extra 'insights' into θ.

Theorem 2: *Given the reduced temporal graph T' , the corresponding reduced spatial graph G', and the static spatial graph G, if the Fisher information of the embedding of T' exceeds the Fisher information of the embedding of G', i.e., $I(T') > I(G')$ then it follows that the covariance of the reduced temporal graph, $Cov(T')$, is less than the covariance of the reduced spatial graph, $Cov(G')$, represented as: $Cov(T') < Cov(G')$ (For proof refer to Supp. Theorem 2)*

Short Proof (*For full proof refer to Sec. 2.2 Supp. Theorem 2*): Given the reduced temporal graph T' and the corresponding reduced spatial graph G', derived from a complete graph G, we assert that higher Fisher information in T' (denoted as $I(T')$) compared to G' (denoted as $I(G')$) implies a lower covariance in T'. Utilizing the Cramér-Rao Lower Bound (Ben-Haim and Eldar (2009)), which suggests a tighter bound on the covariance of any unbiased estimator with higher Fisher information, and considering that T', encapsulating temporal dynamics, inherently contains more information than the spatial snapshot G', it follows that $I(T') > I(G')$. Hence, the inverse relationship between Fisher Information and covariance (CRLB) leads to $\text{Cov}(T') < \text{Cov}(G')$, demonstrating that T' is a more precise estimator for the complete graph G than G'.

Based on the above two theorems, we can deduce that if $\text{Cov}(T')$ and $\text{Cov}(G')$ are the estimators of parameters θ of the full spatial graph $\text{Cov}(G)$ then: $\text{Cov}(T') < \text{Cov}(G')$ i.e. the covariate of the temporal graph $\text{Cov}(T')$ is a better estimator of the complete graph $\text{Cov}(G)$ than Cov(G'). Hence, constructing a temporal graph from the spatial graph of visible nodes in a multi-agent system where some nodes are unobservable all the time yields a superior representation of the entire system compared to the reduced spatial graph, subsequently enhancing the performance of visible agent trajectory prediction.

3. Empirical Evaluation

Datasets We validate the effectiveness of our proposed approach by conducting experiments on four distinct datasets: datasets involving agents connected by springs and charged particles (Kipf et al. (2018)), the CMU motion capture dataset (cmu), and the basketball dataset (Yue et al. (2014)). The first two datasets are simulated, where each sample consists of N particles interacting within a 2D box without any external forces. To introduce hidden agents, we randomly conceal M agents out of the total N agents in the system after completing all the simulations. As for the motion dataset, we specifically select walking sequences from the CMU motion capture dataset. Each sample in this dataset comprises 31 trajectories, where each trajectory corresponds to a single joint of the subject. Similar to the simulated dataset, we randomly hide joints for the subject. On the other hand, the basketball dataset contains trajectories of 5 agents out of 10 agents with 50% observability preprocessed into 49 frame data. Figure 3 shows motion and basketball dataset setup.

Baselines Since we do not have any existing prior work on this work, we consider state-of-the-art models from Table 1 with complete observability and unknown interaction topology. We evaluate against two recurrent neural network (RNN) baselines, Single RNN and Joint RNN, which utilizes shared-weight LSTMs for each object and a concatenated LSTM for all objects' states prediction,

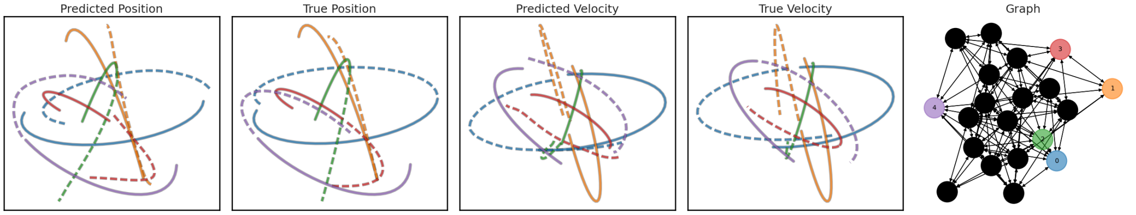

Figure 4: Visualizations depicting predictive trajectories for a system with 10 agents with 75% hidden agents. Dotted lines represent predicted trajectories, while solid lines represent observed trajectories.

Table 2: MSE Error ($\times 10^{-2}$) for 30^{th} step in predicting trajectories for spring interactions.

Total Agents	Springs 10					Springs 20		Springs 30
Unobserved Agents	20%	30%	40%	50%	60%	75%	80%	83.33%
Single RNN (Schmidt (2019))	3.20 ± 1.83	3.88 ± 2.33	3.85 ± 2.37	4.51 ± 2.71	4.33 ± 2.797	4.81 ± 3.49	3.61 ± 2.68	3.60 ± 2.68
FC Graph (Watters et al. (2017))	6.2 ± 2.00	5.91 ± 2.01	5.97 ± 2.12	5.01 ± 2.23	4.01 ± 2.06	2.75 ± 1.26	2.64 ± 1.41	2.55 ± 1.26
JointRNN (Schmidt (2019))	1.23 ± 0.96	1.62 ± 1.20	1.77 ± 1.28	2.10 ± 1.50	2.33 ± 1.73	2.38 ± 1.30	2.46 ± 1.67	2.31 ± 1.48
D-NRI (Graber and Schwing (2020))	1.49 ± 0.75	1.85 ± 0.91	2.34 ± 1.33	2.49 ± 1.85	2.30 ± 1.38	2.77 ± 1.64	1.97 ± 1.28	2.06 ± 1.36
STEMFold (ours)	**0.20 ± 0.16**	**0.62 ± 0.23**	**0.65 ± 0.32**	**0.78 ± 0.39**	**0.96 ± 0.58**	**0.91 ± 0.47**	**0.96 ± 0.59**	**0.97 ± 0.51**

respectively. We also implement Fully Convolutional Graph Messaging, using a message-passing network decoder similar to (Watters et al. (2017)) over a fully connected graph of visible agents. Furthermore, we consider DNRI (Graber and Schwing (2020)), which combines graph neural networks and variational inference, introducing a latent variable model that captures temporal evolution with irregular sampling through an RNN component.

Experimental Settings In our experiments, we studied particles with varying visibility and observed their trajectories within $[t_0, t_h]$. Our model was designed to learn and predict their trajectories for a future interval $[t_{h+1}, t_N]$. We used a 64-dimensional GNN with two layers in its temporal attention module and a 128-dimensional temporal context attention module. For solving differential equations, we applied a Runge-Kutta solver in a single-layer graph network with a 128-dimensional node representation. The time values t_h and t_N were set to 30 and 60 for simulated and motion datasets, and 49 for the basketball dataset. We evaluated trajectory accuracy using mean squared error (MSE).

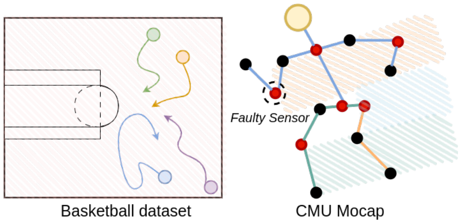

Figure 3: Basketball and CMU Mocap Dataset

Results Figure 4 displays the qualitative results predicting the spring system's behavior, portraying the model's efficacy with 75% hidden, unobservable agents. Within the graph, nodes colored in black symbolize hidden agents, and those in color represent observable ones. Notably, in the system with 75% unobservable agents, agent number 4 demonstrates a unique case—it maintains no connections with visible agents and is exclusively linked to seven hidden ones. Impressively, even in such a challenging scenario, our model proficiently exploits the spatiotemporal observations of visible agents to predict their trajectories with high accuracy. In Figure 5, a visual representation of

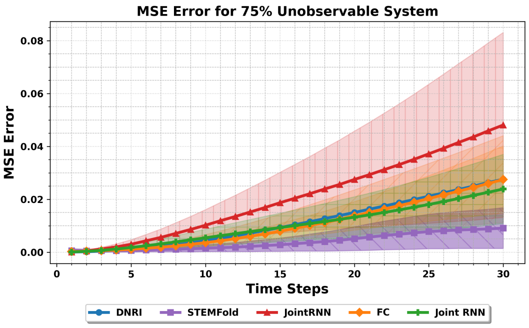

Figure 5: MSE Error values vs time for spring system with 75% unobservable agents.

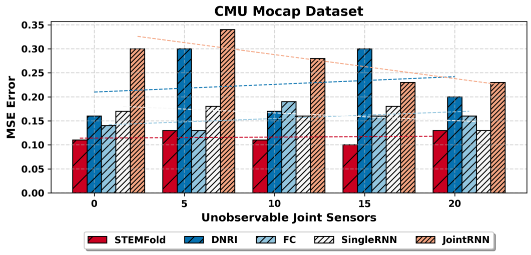

Figure 6: MSE Error ($\times\ 10^{-2}$) in predicting trajectories for Motion Dataset

Table 3: MSE Error ($\times 10^{-2}$) for 30^{th} step in predicting trajectories for charged interactions.

Total Agents	Charged 10					Charged 20		Charged 30
Unobserved Agents	20%	30%	40%	50%	60%	75%	80%	83.33%
Single RNN (Schmidt (2019))	0.54 ± 0.48	0.53 ± 0.49	0.77 ± 0.54	0.78 ± 0.63	0.83 ± 0.69	0.78 ± 0.54	0.88 ± 0.65	1.14 ± 0.73
FC Graph (Watters et al. (2017))	1.17 ± 0.52	1.01 ± 0.49	1.21 ± 0.60	0.91 ± 0.76	1.49 ± 0.76	1.65 ± 0.72	1.71 ± 0.85	2.33 ± 1.14
JointRNN (Schmidt (2019))	0.59 ± 0.59	0.60 ± 0.64	0.79 ± 0.69	0.78 ± 0.75	0.84 ± 0.82	0.88 ± 0.71	1.03 ± 0.82	1.28 ± 1.03
D-NRI (Graber and Schwing (2020))	0.78 ± 0.49	0.61 ± 0.49	0.82 ± 0.51	0.83 ± 0.60	0.75 ± 0.62	1.00 ± 0.66	1.11 ± 0.85	1.34 ± 0.93
STEMFold (ours)	**0.43 ± 0.42**	**0.47 ± 0.48**	**0.59 ± 0.69**	**0.58 ± 0.65**	**0.59 ± 0.7**	**0.72 ± 0.5**	**0.74 ± 0.72**	**0.94 ± 0.68**

the evolution error in dynamics is depicted for the above system, projecting 30 steps into the future. The STEMFold model outperforms all the baseline models in predicting future trajectories while maintaining both low error levels and minimal variance.

Table 2 and Table 3 present the 30^{th} step mean-squared error for trajectory prediction in both the spring and charged systems. We conducted experiments on four systems, specifically 5 agents, 10 agents, 20 agents, and 30 agents, respectively. For each system, we gradually hid agents and trained our framework accordingly. Our network consistently outperforms all the baselines for both systems, affirming the efficacy of our framework's design in learning representation. Even when a large portion of the interaction graph is unobserved, our model exhibits minimal prediction errors in experiments involving 20 or 30 agents with only 4 or 5 agents visible. Figure 6 shows the prediction results for motion datasets with a different set of joints randomly hidden to train the network. Similar to the spring and charged datasets, our network consistently outperforms the baseline models. It is noteworthy, however, that in this dataset, baseline models such as RNN and FC Graph exhibit markedly improved performance compared to their counterparts in the spring and charged datasets. This enhanced performance can be attributed to the inherent geometric constraints of joints moving in synchronization with the overall body's trajectory, facilitating more accurate predictions of each joint's trajectory. This contrast is evident when compared to the spring and charged datasets, where an agent's motion is predominantly influenced by its neighboring agents, with no overarching constraints guiding the entire system's movements.

Prediction of Highly Stochastic Systems Basketball is highly stochastic due to its dynamic nature and the interactions between players that are influenced by numerous unpredictable factors, such as their opponents' actions, their own team's strategies, and spontaneous in-game events. Figure 7 displays the outcomes of the basketball dataset, where only 50% of the agents are observable. To

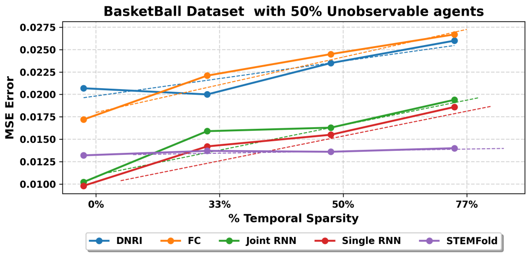

Figure 7: MSE Error for basketball data with 50% observable agents as the temporal sparsity is increased

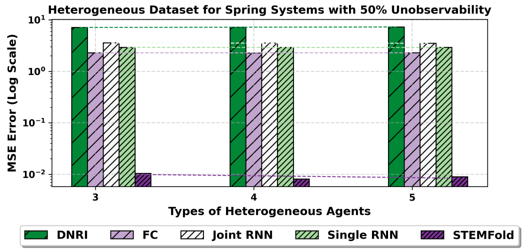

Figure 8: Performance Metrics for Different Models for Heterogeneous Agents

Table 4: Ablation study: MSE error for three STEMFold model variants for different configurations for spring dataset.

Total Agents	Spring 5				Spring 10						Spring 20		Spring 30	
Unobserved Agents	0%	20%	40%	60%	30%	40%	50%	60%	70%	80%	75%	80%	83.33%	87.33%
SF-all connected	1.2	0.6	0.45	0.49	0.67	0.76	0.93	0.63	0.58	0.67	0.58	0.60	0.81	0.59
SF w/o attention	0.22	0.48	1.04	0.60	0.60	1.04	0.70	0.84	0.72	0.73	0.73	0.75	1.08	1.37
SF w/o temporal Encoding	0.28	0.25	0.34	0.5	1.28	0.87	0.38	0.41	0.49	0.68	0.43	0.45	0.9	0.56
STEMFold original	**0.21**	**0.25**	**0.33**	**0.43**	**0.27**	**0.26**	**0.31**	**0.37**	**0.45**	**0.57**	**0.39**	**0.42**	**0.47**	**0.54**

SF-all connected: STEMFold with visible agents fully connected, SF w/o attention: SF without attention mechanism, SF w/o temporal encoding: network with temporal encoding removed, Orignal: network with attention mechanism, temporal encoding and visible graph linkings

further make the task challenging, we introduce temporal sparsity through random sparse sampling to encoder observations and utilize them for trajectory prediction, following the methodology outlined in Sun et al. (2019). Our observations reveal that in scenarios involving concealed agents and limited temporal observability in the basketball dataset, our model surpasses the baseline models in performance.

Importance of Temporal Encoding and Attention Our network comprises two core components: the dynamic spatio-temporal graph and the temporal graph attention. We conducted an ablation study to delve into each module's significance. In the first model variant, we trained the model without prior edge relationship knowledge, resulting in a fully connected temporal graph. The temporal graph attention module consists of two key elements: attention and temporal encoding. For the other two variants, we examined models that lacked either attention or temporal encoding. In these variations, we didn't incorporate attention to nodes over time, and we omitted node temporal importance through temporal encoding. We assessed these models' performance by measuring mean squared error (MSE) across various scenarios in spring simulations. Our original model consistently outperformed all alternative variations, as demonstrated in Table 4.

Analysing Systems with Heterogeneous Agent Characteristics In this section, we explore heterogeneous agents, with variability in agent dynamics with each agent, as a heterogeneous entity, possessing distinct and unknown agent parameters. In contrast to our earlier homogeneous agent experiments, here all the agents exhibit heterogeneity in the dynamics. For these experiments, we explore three types of heterogeneous agents with three dynamics parameter sets. During simulations, each spring heterogeneous agent's coupling parameter is randomly selected from these sets with uniform probability. Figure 8 presents the error metrics for baseline models across different heterogeneous agent configurations with 50% observability, particularly when all agents are consid-

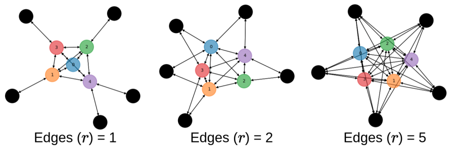

Figure 9: Illustration of the graph configurations to study the influence of hidden agents on visible agent predictions

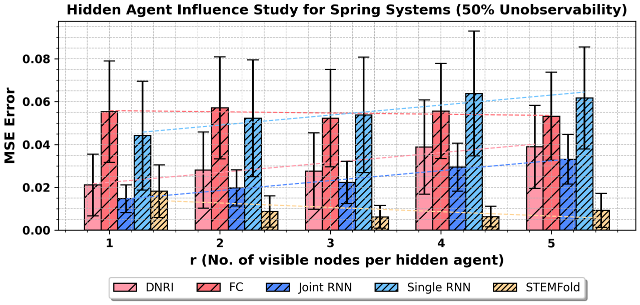

Figure 10: MSE error for models as the number of connections hidden to visible connections are increased.

ered heterogeneous. We observe that baseline models struggle to capture the intricate dynamics of this setup, resulting in significantly higher error rates compared to our proposed model.

Influence of Hidden Agent on Visible Agent Predictions In this study, we establish connections among all visible agents, thereby forming a fully connected subgraph comprised solely of visible agents for the spring system with 50% observability. Subsequently, we incrementally augment the number of edges between hidden and visible agents, ranging from $r = 1$ to $r = 5$. Here, r denotes the number of visible agents each hidden agent is connected to. Notably, there are no interconnections between any two hidden agents. This is illustrated in Figure 9.

Figure 10 illustrates the prediction error for the models on the spring system with 50% observability. It is evident that as the number of connections between hidden and visible agents increases from 2 to 5, STEMFold consistently outperforms, maintaining minimal prediction error and variance. In contrast, the baseline models exhibit a decline in predictive accuracy as the number of hidden-visible agent edges increases. Interestingly, when $r = 1$—signifying that each hidden agent is connected to only one visible agent, the observed error is higher compared to scenarios where each hidden agent is connected to two or more visible agents. This can be attributed to the absence of hidden agents between any two visible agents, resulting in a betweenness centrality of zero for all visible agent pairs with respect to a hidden agent. In contrast, for other configurations, at least one hidden agent exists between any pair of visible agents. This structural difference enables our network to adeptly uncover hidden influences through representation learning on spatiotemporal graphs. For additional insights and ablation studies, please refer to Supp. Section 3.

4. Conclusion

In this work, we have presented a framework for integrating spatiotemporal information from multi-agent observations with multiple co-evolving and interacting agents unobserved. In order to capture the underlying hidden representations of the evolution of dynamics, we propose a dynamic temporal graph to encode the observations to a latent manifold and use a neural ode to propagate the latent interaction dynamics forward. In the future, we would like to estimate the dynamics and intrinsic dimensions of the unobservable agents in the system. We would also like to consider large-scale interacting systems with heterogeneous agents where the interaction relations dynamically evolve over time. While this paper focuses on prediction tasks, an exciting future direction could involve controlling multi-agent systems with hidden agents.

Acknowledgments

This work is supported by the Army Research Office and was accomplished under Grant Number W911NF-19-1-0447. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office or the U.S. Government.

References

Cmu mocap dataset. http://mocap.cs.cmu.edu/. Accessed: 2023-06-05.

Pieter Abbeel and Andrew Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the Twenty-First International Conference on Machine Learning*, ICML '04, page 1, New York, NY, USA, 2004. Association for Computing Machinery. ISBN 1581138385. doi: 10.1145/1015330.1015430. URL https://doi.org/10.1145/1015330.1015430.

Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 961–971, 2016a. doi: 10.1109/CVPR.2016.110.

Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 961–971, 2016b. doi: 10.1109/CVPR.2016.110.

Ferran Alet, Erica Weng, Tomás Lozano-Pérez, and Leslie Pack Kaelbling. Neural relational inference with fast modular meta-learning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/b294504229c668e750dfcc4ea9617f0a-Paper.pdf.

Ershad Banijamali. Neural relational inference with node-specific information. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=HBsJNesj2S.

Zvika Ben-Haim and Yonina C. Eldar. The cramer-rao bound for sparse estimation, 2009.

Colin Graber and Alexander G. Schwing. Dynamic neural relational inference. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Yedid Hoshen. VAIN: attentional multi-agent predictive modeling. *CoRR*, abs/1706.06122, 2017. URL http://arxiv.org/abs/1706.06122.

Zijie Huang, Yizhou Sun, and Wei Wang. Learning continuous system dynamics from irregularly-sampled partial observations. *CoRR*, abs/2011.03880, 2020. URL https://arxiv.org/abs/2011.03880.

Zijie Huang, Yizhou Sun, and Wei Wang. Coupled graph ode for learning interacting system dynamics. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, page 705–715, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383325. doi: 10.1145/3447548.3467385. URL https://doi.org/10.1145/3447548.3467385.

Arash Jahangiri and Hesham A. Rakha. Applying machine learning techniques to transportation mode recognition using mobile phone sensor data. *IEEE Transactions on Intelligent Transportation Systems*, 16(5):2406–2417, 2015. doi: 10.1109/TITS.2015.2405759.

Thomas Kipf, Ethan Fetaya, Kuan-Chieh Wang, Max Welling, and Richard Zemel. Neural relational inference for interacting systems, 2018.

Vineet Kosaraju, Amir Sadeghian, Roberto Martín-Martín, Ian D. Reid, Seyed Hamid Rezatofighi, and Silvio Savarese. Social-bigat: Multimodal trajectory forecasting using bicycle-gan and graph attention networks. *CoRR*, abs/1907.03395, 2019. URL http://arxiv.org/abs/1907.03395.

Jiachen Li, Hengbo Ma, Zhihao Zhang, and Masayoshi Tomizuka. Social-wagdat: Interaction-aware trajectory prediction via wasserstein graph double-attention network. *CoRR*, abs/2002.06241, 2020. URL https://arxiv.org/abs/2002.06241.

Matthias Luber, Johannes A. Stork, Gian Diego Tipaldi, and Kai O. Arras. People tracking with human motion predictions from social forces. In *2010 IEEE International Conference on Robotics and Automation*, pages 464–469, 2010. doi: 10.1109/ROBOT.2010.5509779.

Ivan Marisca, Andrea Cini, and Cesare Alippi. Learning to reconstruct missing data from spatiotemporal graphs with sparse observations, 2022.

Christoforos I. Mavrogiannis and Ross A. Knepper. Multi-agent trajectory prediction and generation with topological invariants enforced by hamiltonian dynamics. In Marco Morales, Lydia Tapia, Gildardo Sánchez-Ante, and Seth Hutchinson, editors, *Algorithmic Foundations of Robotics XIII*, pages 744–761, Cham, 2020. Springer International Publishing. ISBN 978-3-030-44051-0.

Yulia Rubanova, Ricky T. Q. Chen, and David Duvenaud. Latent odes for irregularly-sampled time series. *CoRR*, abs/1907.03907, 2019a. URL http://arxiv.org/abs/1907.03907.

Yulia Rubanova, Ricky T. Q. Chen, and David Duvenaud. Latent odes for irregularly-sampled time series. *CoRR*, abs/1907.03907, 2019b. URL http://arxiv.org/abs/1907.03907.

Adam Santoro, David Raposo, David G. T. Barrett, Mateusz Malinowski, Razvan Pascanu, Peter W. Battaglia, and Timothy P. Lillicrap. A simple neural network module for relational reasoning. *CoRR*, abs/1706.01427, 2017a. URL http://arxiv.org/abs/1706.01427.

Adam Santoro, David Raposo, David G. T. Barrett, Mateusz Malinowski, Razvan Pascanu, Peter W. Battaglia, and Timothy P. Lillicrap. A simple neural network module for relational reasoning. *CoRR*, abs/1706.01427, 2017b. URL http://arxiv.org/abs/1706.01427.

Robin M. Schmidt. Recurrent neural networks (rnns): A gentle introduction and overview, 2019.

Chen Sun, Per Karlsson, Jiajun Wu, Joshua B. Tenenbaum, and Kevin Murphy. Stochastic prediction of multi-agent interactions from partial observations. *CoRR*, abs/1902.09641, 2019. URL http://arxiv.org/abs/1902.09641.

Sjoerd van Steenkiste, Michael Chang, Klaus Greff, and Jürgen Schmidhuber. Relational neural expectation maximization: Unsupervised discovery of objects and their interactions. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=ryH20GbRW.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

Anirudh Vemula, Katharina Mülling, and Jean Oh. Social attention: Modeling attention in human crowds. *CoRR*, abs/1710.04689, 2017. URL http://arxiv.org/abs/1710.04689.

Nicholas Watters, Daniel Zoran, Theophane Weber, Peter Battaglia, Razvan Pascanu, and Andrea Tacchetti. Visual interaction networks: Learning a physics simulator from video. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/8cbd005a556ccd4211ce43f309bc0eac-Paper.pdf.

Janusz Wojtusiak, Tobias Warden, and Otthein Herzog. Machine learning in agent-based stochastic simulation: Inferential theory and evaluation in transportation logistics. *Comput. Math. Appl.*, 64 (12):3658–3665, dec 2012. ISSN 0898-1221. doi: 10.1016/j.camwa.2012.01.079. URL https://doi.org/10.1016/j.camwa.2012.01.079.

Yisong Yue, Patrick Lucey, Peter Carr, Alina Bialkowski, and Iain A. Matthews. Learning fine-grained spatial models for dynamic sports play prediction. *2014 IEEE International Conference on Data Mining*, pages 670–679, 2014. URL https://api.semanticscholar.org/CorpusID:4649228.

Eric Zhan, Stephan Zheng, Yisong Yue, Long Sha, and Patrick Lucey. Generating multi-agent trajectories using programmatic weak supervision, 2019.

Yichen Zhu, Mengtian Zhang, Bo Jiang, Haiming Jin, Jianqiang Huang, and Xinbing Wang. Networked time series prediction with incomplete data. *CoRR*, abs/2110.02271, 2021. URL https://arxiv.org/abs/2110.02271.