Hypertron: Explicit Social-Temporal Hypergraph Framework for Multi-Agent Forecasting

Yu Tian[1,2], **Xingliang Huang**[1,2], **Ruigang Niu**[1,2], **Hongfeng Yu**[1], **Peijin Wang**[1], **Xian Sun**[1,2]
[1]Key Laboratory of Network Information System Technology, Aerospace Information Research Institute, Chinese Academy of Sciences
[2] School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences
{tianyu181, huangxingliang20,niuruigang18,wangpeijin17}@mails.ucas.ac.cn,
{hfyu,sunxian}@mail.ie.ac.cn

Abstract

Forecasting the future trajectories of multiple agents is a core technology for human-robot interaction systems. To predict multi-agent trajectories more accurately, it is inevitable that models need to improve interpretability and reduce redundancy. However, many methods adopt implicit weight calculation or black-box networks to learn the semantic interaction of agents, which obviously lack enough interpretation. In addition, most of the existing works model the relation among all agents in a one-to-one manner, which might lead to irrational trajectory predictions due to its redundancy and noise. To address the above issues, we present Hypertron, a human-understandable and lightweight hypergraph-based multi-agent forecasting framework, to explicitly estimate the motions of multiple agents and generate reasonable trajectories. The framework explicitly interacts among multiple agents and learns their latent intentions by our coarse-to-fine hypergraph convolution interaction module. Our experiments on several challenging real-world trajectory forecasting datasets show that Hypertron outperforms a wide array of state-of-the-art methods while saving over 60% parameters and reducing 30% inference time.

1 Introduction

Human intention prediction is an instinctive ability. Taking pedestrian trajectories as an example, humans can predict others' future trajectories based on a priori experience and explicit features of other agents (e.g., location, sociality, time, speed, etc.) to socially make a proper trajectory strategy. However, building predictive models with such capability is challenging. Such models are often parameter redundant and lack interpretability. Therefore, a good forecasting method should effectively predict future trajectories based on the explicit social and temporal interactions among agents and their latent intentions within an acceptable inference time.

There are many existing methods for multi-agent trajectories prediction within scenes, ranging from recurrent neural network (RNN) [Lee *et al.*, 2017; Ivanovic and Pavone, 2019] or attention mechanism [Giuliari *et al.*, 2021; Yuan *et al.*, 2021] to graph neural network (GNN) [Salzmann *et al.*, 2020; Wang *et al.*, 2021]. As shown in Fig 1, many of their interactions among agents are modeled implicitly, which introduce massive homogeneous learnable parameters. These methods have the following shortcomings in agents' interactions: 1) Homogeneous learning methods adopt a uniform representation to model interactions, which is poorly interpretable because it fails to consider rich semantic information. 2) One-to-one approachs calculate the relation between vertices, which can't simulate the relation among multiple vertices and introduce too much redundancy.

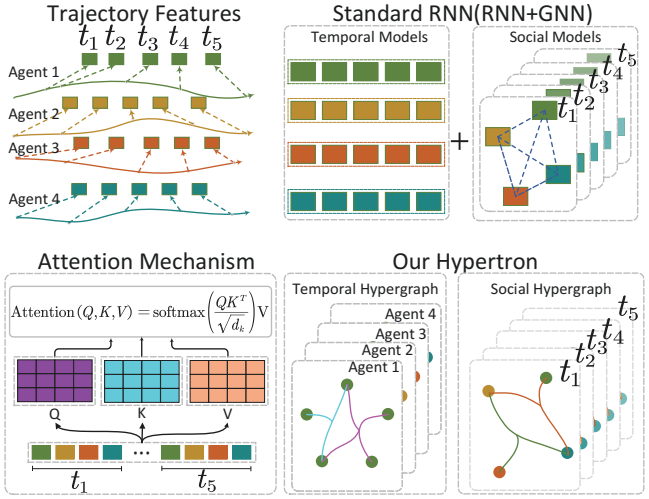

Figure 1: Examples of multi-agent interaction modeling. Different color lines represent different simulation methods. Unlike other standard approaches for implicit, homogeneous, and one-to-one modeling methods, Hypertron adopts an explicit hypergraph modeling approach, which allows for lightweight learning of the relations among vertices by heterogeneous hyperedges.

To tackle the aforementioned problems, we drive our research perspective to a hypergraph approach to model the relations among agents, which is explicit, heterogeneous, and lightweight as shown in Fig 1. We introduce a hypergraph-

based explicit agents interaction method following a coarse-to-fine paradigm, which learns relation among multiple agents simultaneously by heterogeneous hyperedges. Firstly, it constructs coarse temporal and social hypergraphs based on explicit relation among agents (e.g., location, sociality). Then, the coarse hypergraphs are optimized by the coarse-to-fine hypergraph convolution (CFHconv) to incorporate the latent intentions of the agents. Finally, the trajectory features are further updated by the CFconv Interaction Module (CIM), which enables the interaction of agents in different dimensions.

Furthermore, we construct a human-understandable and lightweight framework, dubbed Hypertron, an explicit social-temporal hypergraph framework for multi-agent forecasting. Hypertron stacks multiple layers of CIM to explicitly learn the relation among agents in the social and temporal dimensions and generate diverse and plausible trajectories.

To the best of our knowledge, we are the first to propose a hypergraph based framework for multi-agent forecasting, which is illustrated in Fig 2. Our main contributions are summarized as follows:

(1) We introduce an interpretable and lightweight hypergraph-based trajectory prediction framework that generates diverse hypotheses to reflect plausible future trajectories.

(2) We present a coarse-to-fine hypergraph construction strategy that can explicitly simulate the relation of multi-agent trajectories with a sequence representation.

(3) We design the CFHconv interaction module that stacks social- and temporal-CFHconv to model different dimensional interactions and incorporate latent intentions of agents.

(4) We demonstrate that our framework achieves highly competitive results while saving over 60% parameters and reducing 30% inference time.

2 Related Work

Trajectory/Motion Forecasting. The trajectory sequence can be directly represented by sequential models such as RNN [Lee *et al.*, 2017; Ivanovic and Pavone, 2019]. With the rise of powerful attention mechanism, these methods [Giuliari *et al.*, 2021; Yuan *et al.*, 2021] directly model the relation of all agents at any time, which introduce a lot of redundancy in the feature attending process. Graph-based models [Salzmann *et al.*, 2020; Wang *et al.*, 2021] construct relatively sparse edges, but it's hard to construct efficient and interpretable edges. Most of the above methods introduce too many redundant connections among all agents. Unlike the one-to-one modeling paradigm of prior works, our Hypertron models the high-order relation of agents efficiently with heterogeneous hyperedges.

Social Interaction Modeling. Methods for social interaction modeling mainly focus on the temporal and social dimensions. Sequential methods model the temporal features, and GNN are adopted as social models for agent's interaction[Wang *et al.*, 2021]. However, temporal models and relation models only model the interaction implicitly, since all the edges and attention connections in those methods are homogeneous. Our method explicitly defines hyperedges in temporal and social dimensions for agents, thus the model can easily understand agents' interaction at any dimension.

Hypergraph Learning. An emerging research topic in complex networks is to extend graph to hypergraph [Feng *et al.*, 2019; Ji *et al.*, 2020]. It models high-order constraints among data and learns their correlation by connecting two or more vertices simultaneously. In sequence representation, researchers use hypergraphs for tasks like molecular optimization [Kajino, 2019], video object segmentation [Jiang *et al.*, 2019], and traffic flow prediction [Yi and Park, 2020]. However, most methods adopt an implicit approach in modeling and updating the hypergraph. It is still a worthwhile question to consider how to explicitly construct the hypergraph and make it effective in learning the relation among vertices by giving explicit and heterogeneous definitions to the hyperedges.

3 Methods

3.1 Problem Formulation

We formulate the multi-agent trajectory prediction problem as estimating future trajectory distributions of N (variable) agents based on their motions and the tracking history of all surrounding agents. We denote the current time step as $t = 0$, our goal is to learn the posterior distribution $P(\mathrm{Y}|\mathrm{X}, \mathcal{I})$ of all N agents' future trajectories over T future time steps $\mathrm{Y} = (\mathrm{Y}^1, \mathrm{Y}^2, ..., \mathrm{Y}^T)$. $\mathrm{X} = (\mathrm{X}^{-H}, \mathrm{X}^{-H+1}, ..., \mathrm{X}^0)$ denote all agent states' history at all $H+1$ observed time steps. And depending on the data, the optional conditional items $\mathcal{I}$ may contain the future planning of agents, semantic maps, etc.

3.2 Overview of the Hypertron

To tackle the stochasticity and multi-modality of agents' future behavior, we adopt the conditional variational autoencoders (CVAEs) inside of Hypertron. Our model learns the distribution of future trajectory Y conditioned on past trajectory X and contextual information $\mathcal{I}$ by introducing a stochastic latent variable Z. We reformulate future trajectory distribution as:

$$p\left(\mathrm{Y}|\mathrm{X}, \mathrm{I}\right) = \int p\left(\mathrm{Y}|\mathrm{Z}, \mathrm{X}, \mathrm{I}\right) p\left(\mathrm{Z}|\mathrm{X}, \mathrm{I}\right) d\mathrm{Z} \tag{1}$$

where $p\left(\mathrm{Z}|\mathrm{X}, \mathrm{I}\right)$ is the Gaussian prior distribution and $p\left(\mathrm{Y}|\mathrm{X}, \mathrm{I}, \mathrm{Z}\right)$ is the conditional likelihood distribution. Since the integration over Z is intractable, we apply negative evidence lower bound (ELBO) L_{elbo} as one of our loss functions:

$$\begin{aligned} L_{elbo} = &- E_{q(\mathrm{Z}|\mathrm{Y},\mathrm{X},\mathrm{I})}\left[\log p\left(\mathrm{Y}|\mathrm{Z}, \mathrm{X}, \mathrm{I}\right)\right] \\ &+ D_{KL}\left(q\left(\mathrm{Z}|\mathrm{Y}, \mathrm{X}, \mathrm{I}\right)||p\left(\mathrm{Z}|\mathrm{X}, \mathrm{I}\right)\right) \end{aligned} \tag{2}$$

where $q\left(\mathrm{Z}|\mathrm{Y}, \mathrm{X}, \mathrm{I}\right)$ is the posterior net to approximate the true posterior distribution of Z, and $D_{KL}(||)$ denotes the KL-divergence.

Hypertron can explicitly learn the semantic relation of agents in the social and temporal dimensions and predict the distribution of their future trajectories by stacking multiple layers of CIM-based encoders and decoders. Its overall

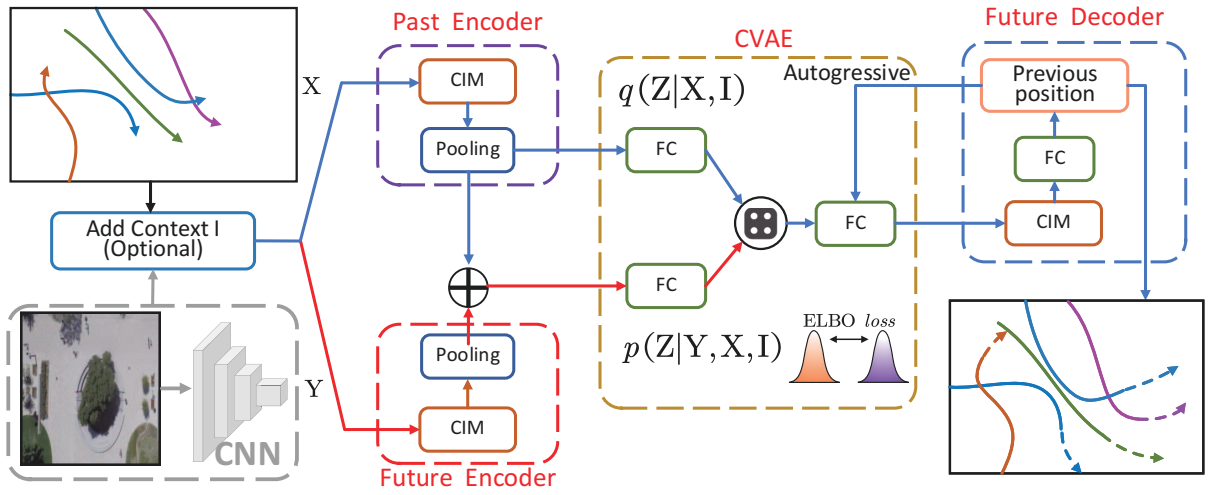

Figure 2: Overall architecture of the Hypertron. It comprises of three sub-networks: Past Encoder, Future Encoder, CVAEs, and Future Decoder. The flow via red paths are only available in the training.

framework is shown in Fig 2, and we will introduce its modules as follows.

Past Encoder. Past Encoder is used to encode the multi-agent past observed sequence X. By feeding X into CIM, we get the explicit interactive past features C_P. Then, we use a mean pooling layer across timesteps to get the mean past features $\tilde{\mathrm{C}}_P = mean(\mathrm{C}_P)$. And fully connected layer (FC) is to map $\tilde{\mathrm{C}}_P$ to the gaussian parameters (μ_p, σ_p) of $p\left(\mathrm{Z}|\mathrm{X},\mathrm{I}\right) = \mathcal{N}(\mu_p, Diag\left(\sigma_p\right)^2)$.

Future Encoder. Similar to Past Encoder, Future Encoder feeds future observed sequence Y into the CIM to learn the semantic relation among multiple agents. Then we compute the mean past features $\tilde{\mathrm{C}}_F = mean(\mathrm{C}_F)$ by the same methods as Past Encoder. And $\tilde{\mathrm{C}}_P$ and $\tilde{\mathrm{C}}_F$ are concatenated and fed into the FC to the gaussian parameters (μ_q, σ_q) of $q\left(\mathrm{Z}|\mathrm{Y},\mathrm{X},\mathrm{I}\right) = \mathcal{N}(\mu_q, Diag\left(\sigma_q\right)^2)$.

Future Decoder. Our future decoder is similar to other recursive network decoders in an autoregressive way. With one exception, we adopt an explicit regression approach and generate social and temporal hypergraphs of the current trajectories to predict the trajectories of the next timestep. The input sequence of decoder can be formed as $\mathrm{F}^i = \hat{\mathrm{Y}}^i + \mathrm{Z}$, where $\hat{\mathrm{Y}}^i$ is the output $\hat{\mathrm{Y}}$ of Future decoder at i-th timestep, $\hat{\mathrm{Y}}^0$ is initialized from the X^N and Z is the sample from past encoder (testing) or future encoder (training). And to approximate $p\left(\mathrm{Y}|\mathrm{Z},\mathrm{X},\mathrm{I}\right)$ according to $q\left(\mathrm{Z}|\mathrm{Y},\mathrm{X},\mathrm{I}\right)$, we minimize the mean squared error between the predicted trajectories and ground truth Y. And our loss is defined as :

$$L = min||\mathrm{Y} - \hat{\mathrm{Y}}||L_{elbo} \tag{3}$$

To generate more diverse and plausible trajectories after the training phase, we refer to [Yuan *et al.*, 2021] and adopt the diversity sampling technique in DLow [Yuan and Kitani, 2020] to Hypertron.

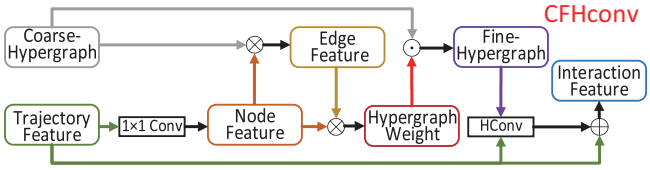

Figure 3: Overview of the coarse-to-fine hypergraph convolution.

3.3 CFHconv

To better model agents' social and temporal relation by explicit hypergraph, we propose coarse-to-fine hypergraph convolution (CFHconv), a coarse-to-fine paradigm (as shown in Fig 3). The mathematical formulation of the CFHconv to aggregate agents' information from input feature X and coarse hypergraph $\mathcal{H}_{coa}$ is given as:

$$\mathcal{X} = Hconv(\mathcal{A}(\mathcal{F}_e(\mathcal{H}_{coa}, \mathrm{X}_v))) \tag{4}$$

where X_v is vertex feature learned from trajectory feature by a nonlinear feature transformation (instantiated as 1×1 convolution). $Hconv$ is hypergraph convolution, $\mathcal{A}$ denotes the constructing function, and $\mathcal{F}_e$ is the hyperadge aggregation function. Different from the traditional hypergraph convolution, our module adopts a coarse-to-fine paradigm, and we further state the structure in three steps:

Step1 (Hyperedge feature calculation). Our aim is to optimize the coarse hypergraph by computing the fine-grained association between hyperedge features and vertex features. Therefore, the module constructs the hyperedge feature X_e from the vertex features with the coarse hypergraph. For simplicity, we implement $\mathcal{F}_e$ by a matrix multiplication with an embedding:

$$\mathrm{X}_e = W_e(\mathrm{X}_v \otimes \mathcal{H}) \tag{5}$$

where $\otimes$ denotes matrix multiplication, W_e is a hyperedge embedding to be learned, that can be implemented by a nonlinear function (instation 1×1 convolution and $ReLU(\cdot)$).

Step2 (Fined hypergraph construction). Then, we calculate the correlation matrix W_h by hyperedge features and node features, and thus further optimize the coarse hypergraph by W_h to obtain the fined hypergraph $\mathcal{H}_f$:

$$W_h = \mathrm{X}_e \otimes \mathrm{X}_v \tag{6}$$
$$\mathcal{H}_f = softmax(W_h \odot \mathcal{H}_c) \tag{7}$$

where $\odot$ is element-wise multiplication, and $softmax$ is used for normalization.

By this paradigm, our fined hypergraph can learn the explicit (distance calculation) and intrinsic (feature association) connections of agents in temporal and social dimensions.

Step3 (Hypergraph convolution module). Since fined hypergraph contains more relational information (distance and association weights) than the traditional hypergraph, we need to redesign the hypergraph convolution for it.

We define fined vertex $\tilde{\mathrm{D}}$ and fined hyperedge degree $\tilde{\mathrm{B}}$ as $\tilde{\mathrm{D}}_{ii} = \sum_{j=1}^{K} \mathcal{H}_f(i,j)$, $\tilde{\mathrm{B}}_{jj} = \sum_{i=1}^{N} \mathcal{H}_f(i,j) = 1$, respectively. $\mathcal{H}_f(i,j)$ denotes the corrlation between the j-th hyperedge to the i-th vertex. Fined vertex $\tilde{\mathrm{D}}$, fined hyperedge degree $\tilde{\mathrm{B}}$, and fined hypergraph $\mathcal{H}_f$ are used to calculate our hypergraph convolution.

By following the transfer formulation in [Feng *et al.*, 2019; Zhu *et al.*, 2021], we design our hypergraph convolution module, and it can be generalized as:

$$\tilde{\mathrm{X}}_v = \sigma(\tilde{\mathrm{D}}^{-\frac{1}{2}} \mathcal{H}_f \mathcal{H}_f^{\mathrm{T}} \tilde{\mathrm{D}}^{-\frac{1}{2}} \mathrm{X}_v \Theta) \tag{8}$$

where Θ is a trainable parameter and σ denotes an nonlinear activation function (implemented by $ReLU(\cdot)$), and $\tilde{\mathrm{X}}_e$ is the interaction feature. Since $\tilde{\mathrm{B}}$ is an identity matrix, it is omitted in Eq 8.

Further, we use the residual connection to update the vertex features and aggregate the interaction feature $\tilde{\mathrm{X}}_v$. The calculation of the output features $\hat{\mathrm{X}}_v$ can be formulated as:

$$\hat{\mathrm{X}}_v = \tilde{\mathrm{X}}_v + \mathrm{X}_v \tag{9}$$

By residual connection, our module can easily stack multiple layers and can be inserted into other model without breaking its initial behavior.

3.4 CFHconv Interaction Module

Our CFconv Interaction Module (CIM) is an interaction module based on CFHconv. As shown in Fig 4, it learns the interactions of different agents in temporal and social dimensions through fine-grained hypergraphs. For better applicability to the sequence representation task, CIM follows the encoder and decoder design of [Lee *et al.*, 2017; Yuan *et al.*, 2021]. Compared with the traditional hypergraph convolutional neural network, CFconv has two main types of differences: (1) It adopts an explicit modeling approach, thus the information interaction is more target-oriented, and the network is more lightweight. (2) It follows a coarse-to-fine modeling idea, which provides a more comprehensive interaction and updating process.

Trajectory Feature Extractor. Given N agents' trajectories over T times $\mathrm{S} = \{S_{pos}^t\}_{t=1}^{t=T}$ with their associated scene

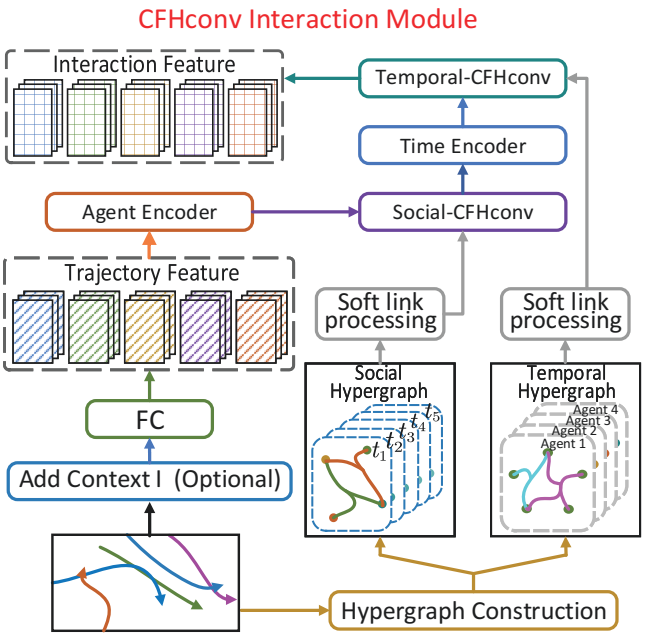

Figure 4: Overview of CFconv Interaction Module. It adopts a coarse-to-fine paradigm to learns relation among multiple agents simultaneously by heterogeneous hyperedges in social and temporal dimensions.

context I (e.g., semantic map information extracted by CNN networks), and S_{pos}^t is all agents' normalized positions at time t. Based on S_{pos}^t, the extractor estimates agents' velocities S_{vel}^t and accelerations S_{acc}^t to obtain variable trajectory features $\mathrm{X} = \mathrm{X}_1, \mathrm{X}_2, ..., \mathrm{X}_T$, and X_t is defined as:

$$\mathrm{X}_t = \mathrm{FC}([S_{pos}^t, S_{vel}^t, S_{acc}^t, S_I^t]) \tag{10}$$

where S_I^t is the current context, and FC is a fully connected layer that maps trajectory features to high-dimensional space.

Explicit Hypergraph Construction. Thanks to the property that hyperedge can contain multiple vertices, it can describe complex associations among multiple vertices more accurately by fewer hyperedges. We propose a human-understandable and machine-learnable hypergraph construction approach that can explicitly model the relation among agents in social and temporal dimensions: (1) **Social dimension,** we construct the social hypergraph $\mathrm{H}_{soc}^t = \{e_n^t\}_{n=1}^{n=N}$ at the t step based on the Euclidean distances of all agents at time t, where e_n^t denotes the relation between n-th agent and other agents. (2) **Temporal dimension,** the temporal hypergraph $\mathrm{H}_{tem}^n = \{\varepsilon_n^t\}_{n=1}^{n=N}$ of n-th agent is constructed by multiplying the Euclidean distance with a scaling factor based on the length of the time step, where ε_n^t represents the relation of an agent at different times. Meanwhile, to let the hypergraph focus on more relevant vertex information, we design a soft link, a module that uses a threshold θ to filter the too-small correlation noise and normalize it by softmax function. The soft link processing is computed as:

$$\mathcal{H} = softmax(\mathrm{H}), \begin{cases} \mathrm{H}_{i,j} = \mathrm{H}_{i,j}, \mathrm{H}_{i,j} > \theta \\ \mathrm{H}_{i,j} = 0, \mathrm{H}_{i,j} < \theta \end{cases} \tag{11}$$

Method	ADE_{20}/FDE_{20} ↓(m), $K = 20$Samples					
	ETH	Hotel	Univ	Zara1	Zara2	Average
S-GAN	0.81/1.52	0.72/1.61	0.60/1.26	0.34/0.69	0.42/0.84	0.58/1.18
PECNet	0.54/0.87	0.18/0.24	0.35/0.60	0.22/0.39	0.17/0.30	0.29/0.48
STAR	0.36/0.65	0.17/0.36	0.31/0.62	0.26/0.55	0.22/0.46	0.26/0.53
AgentFormer	0.45/0.75	0.14/0.22	0.25/0.45	0.18/**0.30**	0.14/0.24	0.23/0.39
Trajectron++	0.39/0.83	**0.12**/0.21	**0.20/0.44**	**0.15**/0.33	**0.11**/0.25	**0.19**/0.41
Hypertron	**0.35/0.51**	0.13/**0.17**	0.22/**0.44**	0.19/0.31	0.15/**0.23**	0.21/**0.34**

Table 1: Comparison to SOTA models on ETH/UCY dataset.

where H_j^i denotes the correlation between the j-th hyperedge to the i-th vertex.

Agent/Time Encoding. Unlike other sequential representation networks, Hypertron feeds into all agents' temporal data in a fixed time period simultaneously. Therefore, before performing Social- and Temporal-CFHconv, we adopt positional encoding to add identity information to the data in social and time dimensions to inform each element's explicit criteria for CFHconv. For trajectory sequence X, we use sine and cosine functions to locate embedding:

$$P_{obs}^i = \begin{cases} \sin\left(\frac{i}{10000^{d/D}}\right), d \text{ is even} \\ \cos\left(\frac{i}{10000^{d/D}}\right), d \text{ is odd} \end{cases} \tag{12}$$

where P_{obs}^i denote the i-th feature of P_{obs} and d is each dimension of embedding. And the positional encoding output $\hat{\mathrm{X}}$ is computed as $\hat{\mathrm{X}} = W_2(W_1\mathrm{X} + P_{obs})$, where W_1,W_2 is a learnable transformation function.

4 Experiments

Datasets. To evaluate our methods, we conduct experiments on three publicly examined datasets: The ETH/UCY datasets and the Stanford Drone Dataset. As in previous works, the experimental models predict the next 12 timesteps trajectories by observing trajectories of last 8 timesteps.

There are five subsets of real-world pedestrian trajectory prediction in ETH/UCY datasets, each of which contains complex pedestrian behavior, including multi-directional nonlinear trajectories, working together, standing and unpredictable movement to avoid collisions, etc. The multi-agent social scene trajectories with rich interactions are sampled by 0.4 second intervals.

The Stanford drone dataset (SSD) includes 20 top-down scenes of university scenes captured by drones. It is much larger than ETH/UCY and includes other objects like bicycles or cars in addition to pedestrians. And the frame rate is the same as for ETH/UCY.

Metrics. We employ minimum average displacement error ($\mathrm{ADE_K}$) and minimum final displacement error ($\mathrm{FDE_K}$) of K prediction samples $\hat{\mathrm{Y}}_k$ of each agent compared to the ground truth Y_k: $\mathrm{ADE_K} = \frac{1}{T}\min_{k=1}^{K}\sum_{t=1}^{T}\|\hat{y}_n^{t,(k)} - y_n^t\|^2$, $\mathrm{FDE_K} = \min_{k=1}^{K}\|\hat{y}_n^{t,(k)} - y_n^t\|^2$, where $\hat{y}_n^{t,(k)}$ denotes the future location of agent n at time t in the k-th sample and y_n^t is the corresponding ground truth.

Implementation Details. We train the Hypertron with Adam optimizer, and the initial learning rate is 0.001. The number of hyperedges in the social hypergraph is set to 32, and each hyperedge e_s^i indicates the social correlation of the i-th agent with others. Similarly, the counterpart of the temporal hypergraph is set to 20, and each e_t^j indicates the temporal correlation of the agent in the j-th timestep with others.

	S-GAN	Sophie	PECNet	PCCSNet	Hypertron
ADE_{20}	27.23	16.27	9.96	8.62	8.86
FDE_{20}	41.44	29.38	15.88	16.16	16.25

Table 2: Comparison to SOTA models on SSD dataset.

Methods	Parameters	Inference time	$mADE_{20}/mFDE_{20}$
S-GAN	**43.6K**	**0.3258s**	0.58/1.18
S-LSTM	252K	0.9356s	0.72/1.54
PECNet	2.08M	0.6070s	0.29/0.48
STAR	1.06M	0.6211s	0.26/0.53
AgentFormer	1.53M	0.7453s	0.23/0.39
Trajectron++	0.87M	0.8469s	**0.19**/0.41
Hypertron	0.61M	0.5138s	0.21/**0.34**

Table 3: Comparison to SOTA models in parameters and inference time. $mADE_{20}$ and $mFDE_{20}$ denote the average performance on ETH/UCY.

4.1 Quantitative Evaluation

Accuracy. For fair comparisons, we do not use any semantic/visual information for ETH/UCY to compare with prior work, including S-GAN [Gupta *et al.*, 2018], PECNet [Mangalam *et al.*, 2020], STAR [Yu *et al.*, 2020], Agentformer [Yuan *et al.*, 2021] and Trajectron++ [Salzmann *et al.*, 2020]. Tab 1 shows the results on all five subsets. We find that our method significantly improves state of the art (SOTA), where it outperforms other works on 1 and 4 out of 5 subsets in ADE_{20}/FDE_{20}, respectively. Especially, Hypertron achieves the lowest average FDE_{20} error of 0.34 on all subsets, outperforming the second best method (Trajectron++) by 17%. Notably, our method shows promising results on two higher crowd density subsets, ETH and University (Univ), illustrating that Hypertron can effectively interact among multiple agents in dense scenes by the social and temporal hypergraphs and generate accurate future trajectories.

We report the performance on SSD dataset in Tab 2 with Sophie [Sadeghian *et al.*, 2019], PCCSNet [Sun *et al.*, 2021], etc. Our model can obtain competitive results on SSD dataset, achieving only a lower accuracy than PCCSNet. We argue that our model does not extract enough contextual information and that a better extract approach should lead to improved performance. This result reflects that Hypertron can effectively achieve temporal and social interaction among agents and has strong robustness.

Efficiency. The number of parameters and the inference time are essential to model's applicability. We compare the parameters and inference time of Hypertron with other SOTA in Tab 3, including S-GAN, S-LSTM [Alahi *et al.*, 2016], etc. To measure the inference time, we use a V100 GPU. Hypertron is very compact compared to previous work and achieves a better balance of size, inference time, and performance.

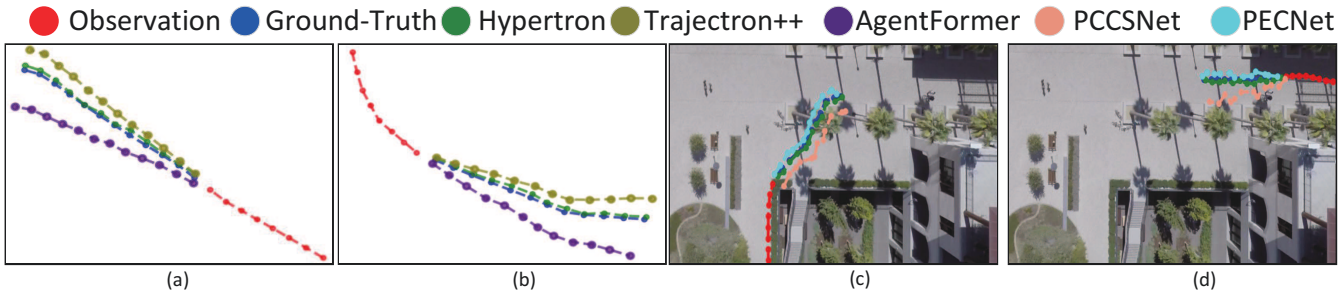

Figure 5: Trajectory visualization. (a), (b) are in the ETH/UCY, and (c), (d) are in the SSD.

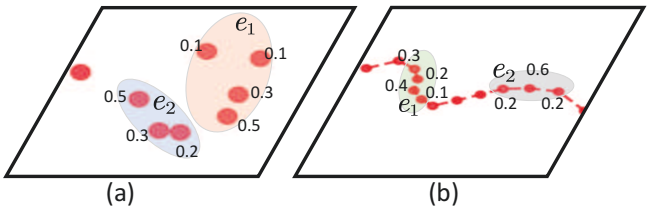

Figure 6: Visualization of coarse-to-fine hypergraph. (a) is a social hypergraph at one timestep and (b) is a temporal hypergraph for one agent's trajectory.

Method	ADE_{20}/FDE_{20} ↓(m), $K = 20$Samples					
	ETH	Hotel	Univ	Zara1	Zara2	Average
w/o Social	0.47/0.70	0.16/0.22	0.29/0.49	0.24/0.35	0.20/0.33	0.27/0.42
w/o Temporal	0.43/0.61	0.14/0.20	0.29/**0.44**	0.22/0.32	0.18/0.31	0.25/0.38
Social-TF	0.39/0.56	**0.13**/0.20	0.24/0.51	0.21/0.36	0.16/0.24	0.23/0.37
Temporal-TF	0.39/0.55	0.14/0.19	0.26/0.46	**0.19**/0.35	0.17/0.27	0.23/0.36
Hypertron	**0.35/0.51**	**0.13/0.17**	**0.22/0.44**	**0.19/0.31**	**0.15/0.23**	**0.21/0.33**

Table 4: Ablation studies on the ETH/UCY. w/o Social/Temporal means without Social/Temporal-CFHConv. Social/Temporal-TF denotes replacing Social/Temporal-CFHconv with transformer.

The results show that Hypertron can save over 60% parameters and reduce more than 30% inference time compared to AgentFormer. We analysis that our model can achieve better performance with acceptable inference time and parameters because our CIM uses hypergraphs to construct the relations among agents in social and temporal dimensions. It can achieve lightweight interactions while reducing the noise and useless computation from interaction redundancy.

Ablation Studies. We further conduct extensive ablation studies at ETH/UCY. The results are shown in Tab 4, and we find each component of our model plays a key role in improving performance. First, we use Social-CFHconv and Temporal-CFHconv separately instead of our final model to explore the effectiveness of our module. The results in a loss of at least 0.04/0.05 $mADE_{20}/mFDE_{20}$ score on the test set, showing that Social/Temporal-CFHConv can greatly improve our model by interacting in the social and temporal dimensions. Second, we replace Social/Temporal-CFHConv with Transformer [Vaswani *et al.*, 2017] to further enhance the validation of our model performance. The performance drops at least 0.02/0.03 $mADE_{20}/mFDE_{20}$ score, indicating that our CFHConv has better interaction capabilities than Transformer. Our performance gains are attributed to our explicit hypergraph construction, it allows Hypertron to learn the specific meaning of each hyperedge and achieve target-oriented optimization of the model.

4.2 Qualitative Evaluation

Trajectory visualization. Fig 5 shows Hypertron's most likely predictions for some examples of ETH/UCY and SSD datasets. We find that Hypertron's predictions are consistent with these groups, resulting in lower error than other models. By comparing results on both datasets, although our model shows good performance, it still has relative errors in prediction due to the lack of additional contextual information (e.g., map, environment information). However, by adding additional information, the prediction results are more stable.

Visualization of coarse-to-fine hypergraph. We visualize some coarse-to-fine hypergraph as examples in Fig 6. Fig 6(a) is a social hypergraph at one timestep and Fig 6(b) is a temporal hypergraph for one agent's trajectory. We can find our hypergraphs can model both social and temporal dimensions very well. In the social dimension, with the social hypergraph, weights can be assigned based on the explicit distance among agents and their latent intentions. And with the temporal hypergraph, it can focus on the relevance and importance between different timesteps. It is worth noting that we explicitly focus on the distance information between different timesteps in the temporal hypergraph, so we can promisingly find that Hypertron can kindly handle the case of sharp turns by assigning a relatively higher weight to the closer vertices.

5 Conclusion

In this work, we present Hypertron, an explicit social-temporal hypergraphs framework for multi-agent forecasting. It generates diverse and reasonable trajectories by stacking multiple explicit agent interaction modules to estimate the intentions of agents. The interaction module constructs coarse temporal and social hypergraphs based on explicit relation among agents (e.g., location, sociality). The coarse hypergraphs are optimized by the coarse-to-fine hypergraph convolution to incorporate the latent intentions of the agents. Extensive experiments on several challenging trajectory forecasting datasets with SOTA methods show that Hypertron achieves better performance with fewer parameters and inference time.

Acknowledgments

The work is supported by the National Natural Science Foundation of China under Grant 62171436 and 61725105.

References

[Alahi *et al.*, 2016] 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 CVPR*, pages 961–971, 2016.

[Feng *et al.*, 2019] Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. Hypergraph neural networks. In *Proceedings of AAAI*, volume 33, pages 3558–3565, 2019.

[Giuliari *et al.*, 2021] Francesco Giuliari, Irtiza Hasan, Marco Cristani, and Fabio Galasso. Transformer networks for trajectory forecasting. In *Proceedings of ICPR*, pages 10335–10342. IEEE, 2021.

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

[Ivanovic and Pavone, 2019] Boris Ivanovic and Marco Pavone. The trajectron: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs. In *Proceedings of ICCV*, pages 2375–2384, 2019.

[Ji *et al.*, 2020] Shuyi Ji, Yifan Feng, Rongrong Ji, Xibin Zhao, Wanwan Tang, and Yue Gao. Dual channel hypergraph collaborative filtering. In *Proceedings of SIGKDD*, pages 2020–2029, 2020.

[Jiang *et al.*, 2019] Zhengkai Jiang, Peng Gao, Chaoxu Guo, Qian Zhang, Shiming Xiang, and Chunhong Pan. Video object detection with locally-weighted deformable neighbors. In *Proceedings of AAAI*, volume 33, pages 8529–8536, 2019.

[Kajino, 2019] Hiroshi Kajino. Molecular hypergraph grammar with its application to molecular optimization. In *Proceedings of ICML*, pages 3183–3191. PMLR, 2019.

[Lee *et al.*, 2017] Namhoon Lee, Wongun Choi, Paul Vernaza, Christopher B Choy, Philip HS Torr, and Manmohan Chandraker. Desire: Distant future prediction in dynamic scenes with interacting agents. In *Proceedings of CVPR*, pages 336–345, 2017.

[Mangalam *et al.*, 2020] Karttikeya Mangalam, Harshayu Girase, Shreyas Agarwal, Kuan-Hui Lee, Ehsan Adeli, Jitendra Malik, and Adrien Gaidon. It is not the journey but the destination: Endpoint conditioned trajectory prediction. In *Proceedings of ECCV*, pages 759–776. Springer, 2020.

[Sadeghian *et al.*, 2019] Amir Sadeghian, Vineet Kosaraju, Ali Sadeghian, Noriaki Hirose, Hamid Rezatofighi, and Silvio Savarese. Sophie: An attentive gan for predicting paths compliant to social and physical constraints. In *Proceedings of CVPR*, pages 1349–1358, 2019.

[Salzmann *et al.*, 2020] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *Proceedings of ECCV*, pages 683–700. Springer, 2020.

[Sun *et al.*, 2021] Jianhua Sun, Yuxuan Li, Hao-Shu Fang, and Cewu Lu. Three steps to multimodal trajectory prediction: Modality clustering, classification and synthesis. In *Proceedings of ICCV*, pages 13250–13259, October 2021.

[Vaswani *et al.*, 2017] 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 NeurIPS*, pages 5998–6008, 2017.

[Wang *et al.*, 2021] Chengxin Wang, Shaofeng Cai, and Gary Tan. Graphtcn: Spatio-temporal interaction modeling for human trajectory prediction. In *Proceedings WCACV*, pages 3450–3459, 2021.

[Yi and Park, 2020] Jaehyuk Yi and Jinkyoo Park. Hypergraph convolutional recurrent neural network. In *Proceedings of SIGKDD*, pages 3366–3376, 2020.

[Yu *et al.*, 2020] Cunjun Yu, Xiao Ma, Jiawei Ren, Haiyu Zhao, and Shuai Yi. Spatio-temporal graph transformer networks for pedestrian trajectory prediction. In *Proceedings of ECCV*, pages 507–523. Springer, 2020.

[Yuan and Kitani, 2020] Ye Yuan and Kris Kitani. Dlow: Diversifying latent flows for diverse human motion prediction. In *Proceedings of ECCV*, pages 346–364. Springer, 2020.

[Yuan *et al.*, 2021] Ye Yuan, Xinshuo Weng, Yanglan Ou, and Kris Kitani. Agentformer: Agent-aware transformers for socio-temporal multi-agent forecasting. In *Proceedings of ICCV*, 2021.

[Zhu *et al.*, 2021] Yutao Zhu, Kun Zhou, Jian-Yun Nie, Shengchao Liu, and Zhicheng Dou. Neural sentence ordering based on constraint graphs. In *Proceedings of AAAI*, pages 14656–14664. AAAI Press, 2021.