2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

DyHGDAT: Dynamic Hypergraph Dual Attention Network for multi-agent trajectory prediction

Weilong Lin[1] Xinhua Zeng[1,*] Chengxin Pang[2] Jing Teng[3] Jing Liu[4,5]

***Abstract*— Modeling the interactions among agents based on their historical trajectories is key to precise multi-agent trajectory prediction. Hypergraph Convolutional Networks (HGCN) have become a proper choice for capturing high-order interactions among agents in this field. However, most existing works only consider static hypergraphs, and ignore that in a hypergraph, the power of influence varies between vertices (or hyperedges). Therefore, we propose DyHGDAT, a dynamic hypergraph dual attention network to capture the high-order interactions among agents, which not only models the evolution of hypergraph over time but also highlights the vertices and hyperedges with larger impacts. We apply DyHGDAT to a CVAE-based prediction system for predicting plausible trajectories. To validate the effectiveness of prediction, we evaluate our proposed method on two well-established trajectory prediction datasets: the ETH/UCY datasets and the Stanford Drone Dataset (SDD). The experimental results show that with DyHGDAT, the CVAE-based prediction system outperforms state-of-the-art methods by 12.5%/5.3% in ADE/FDE on ETH/UCY, and the improvement on SDD is 6.4%/7.4%.**

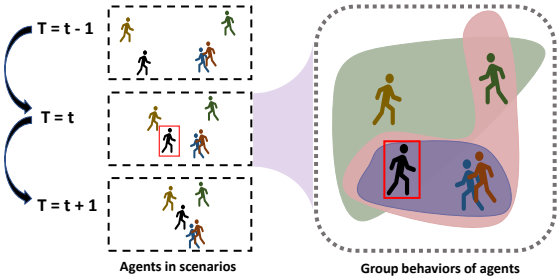

Fig. 1: An example of DyHGDAT captures high-order interactions among agents at each time step. The right side of Fig. 1 shows the group behaviors among agents at $T = t$, enclosed areas of different colors refer to the different group behaviors. In a hypergraph, the group behaviors will change over time, and the power of influence varies between vertices (or hyperedges).

I. Introduction

Multi-agent trajectory prediction aims to predict future trajectories of agents based on historical trajectories, which is crucial in many applications, such as multi-agent automatic driving, robotics, and surveillance systems. However, multi-agent trajectory prediction is challenging due to the complex social interactions among agents. Each agent affects other agents because of its latent intent and is also affected by other agents. To accurately predict the trajectories, we need to model the complex social interactions among agents.

There are many works attempt to model social interactions between agents, such as social pooling mechanism [1], [2], attention mechanism [3]–[6], and graph neural network [7]–[11]. Despite their success, there are several key challenges in modeling social interactions: (1) Most Previous works focus only on pair-wise interactions between agents, which are not always sufficient since there is a lot of group behavior in the real world, such as swarm behavior, cooperative hunting by wolves, and migration of migrating birds, etc. These group behaviors are seldom accurately modeled. Although Xu et al. [12] apply hypergraph to model the high-order interactions among agents and represent the group behavior with hyperedges, they ignore the evolution of hypergraph over time, i.e., the evolution of the group behavior is ignored. (2) while propagating and aggregating the vertex embedding within a hypergraph, some vertices may be dominant in a hyperedge while other vertices are not; Moreover, hyperedges may have different impacts on a vertex. To better represent learning, a key challenge is to highlight the dominant vertices and emphasize the hyperedges with larger impacts.

To tackle the aforementioned problems, we put efforts into two aspects: modeling the evolution of hypergraphs and enhancing hypergraph representation capability. To model the evolution of hypergraphs, we extend static hypergraphs to dynamic hypergraphs by constructing a hypergraph and inferring its topology at each time step. To enhance the representation capability of hypergraph, we propose a dual attention mechanism and design a hypergraph dual-attention paradigm, which enables the hypergraph to adaptively highlight the vertices and hyperedges with larger impacts.

Overall, we integrate these two designs and propose DyHGDAT, a Dynamic Hypergraph Dual Attention Network to capture more comprehensive high-order interactions. The main contributions of this paper are summarized as follows:

- We extend static hypergraphs to dynamic hypergraphs to model the evolution of the hypergraph structure over time, which enables our model to capture more comprehensive high-order interactions among agents.
- We propose DyHGDAT based on the dual attention mechanism to explicitly emphasize the hyperedges and vertices with larger impacts, which enhance the repre-

[1] Academy for Engineering&Technology, Fudan University, Shanghai, 200433, China
[2] School of Electronics and Information Engineering, Shanghai University of Electric Power, Shanghai, China
[3] School of Control and Computer Engineering, North China Electric Power University, Beijing, China
[4] School of Information Science and Technology, Fudan University, Shanghai, China
[5] Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, BC V6T 1Z4, Canada

sentation capability of hypergraph neural networks.
- We conduct full experiments on the ETH/UCY dataset and the SDD, the experimental result shows that our method significantly improves the SOTA on well-established multi-agent trajectory prediction datasets.

II. Related Work

A. Mutli-agent trajectory prediction

Most existing works formulate multi-agent trajectory prediction as a sequential prediction problem. Traditional approaches predict trajectories with handcrafted rules and energy functions [13]–[15], which tend to capture only simple interactions (e.g., repulsion and attractions) and might fail to generalize for complex settings. Thanks to the great advances in deep learning, sequential models such as GRU [16] and LSTM [17] are used to extract the temporal dependence of the trajectory sequence. Recent works focus on modeling social interactions between agents, early works [1], [2] merge agents' latent state by a pooling mechanism to consider the spatial interactions; and the attention mechanism [3]–[6] are also applied to capture spatial-temporal dependencies; Some recent works [7]–[11] explicitly model social interactions between agents base on graph convolutional network. All of these works focus only on modeling pair-wise interactions and ignoring the group behavior's influence on agents, while some studies [2], [18]–[20] begin to model the group behaviors among agents, most of these works rely on predefined annotations and fail to take into account the evolution of group behavior over time, which lead to the suboptimal modeling results. In this work, we extend static hypergraphs to dynamic hypergraphs to model the group behaviors among agents and learn the topology of hypergraphs in a data-driven way to capture the evolution of hypergraphs to capture more comprehensive high-order interactions.

B. Hypergraph learning

Hypergraph is a mathematical structure that extends the concept of simple graph to include the hyperedges that can connect more than two vertices. Zhou et al. [21] first introduce hypergraph learning to model the high-order relations among vertices; Li et al. [22] define the notion of p-Laplacians and derive corresponding nodal domain theorems and k-way Cheeger inequalities; Feng et al. [23] introduce the first hypergraph deep learning method, and Bai et al. [24] apply hypergraph attention to the hypergraph neural networks to enhance the representation capacity of hypergraph, but such attention mechanism focuses only on modeling the relationship between vertices and predefined hyperedges, thus failing to highlight the vertices which have a larger impact on hyperedges. Recently, hypergraphs have gained a lot of attention in the field of machine learning for capturing high-order relationships and dependencies. Xu et al. [12] relational reasoning about group behaviors of agents using multiscale hypergraphs; Jiang et al. [26] use hypergraph convolution to encode high-order data relations and dynamically update hypergraph structure on each layer. However, most previous works ignore that some vertices may be dominant within a hyperedge while other vertices are not, and hyperedges may have different impacts on a vertex. Consider the case where the hyperedges are homologous to the vertices, we propose the DyHGDAT based on the dual attention mechanism which can highlight the dominant vertices as well as hyperedges with larger impacts.

III. Problem Formulation

Given a set of N agents in a scene, we formulate the task of multi-agent trajectory prediction as generating a distribution of future trajectories for all agents based on their historical trajectory. Mathematically, we donate current timestep as$t = 0$, for observed timesteps, we represent the historical trajectories of all N agents as $X \in R^{T_{obs} \times N \times 2}$, and the 2D coordinates of all agents at past time t are defined as $X^t = [x_1^t, x_2^t, \ldots, x_N^t] \in R^{N \times 2}$. Similarly, the future trajectories of agents are donated as $Y \in R^{T_{pred} \times N \times 2}$ and the 2D coordinates of all agents at future time t are represented as $Y^t = [y_1^t, y_2^t, \ldots, y_N^t] \in R^{N \times 2}$. In addition, for hypergraphs at each time step t, we define it as $\mathcal{G}^t = (\mathcal{V}, \mathcal{E}, \mathcal{H})$, where $\mathcal{V}$ and $\mathcal{E}$ refer to the set of vertices (agents) and hyperedges (group behaviors) respectively, $\mathcal{H}$ is the topology of hypergraph which store the connectivity between hyperedges and vertices. Our goal is to learn a deep generative model $P_\theta(Y|X)$ where θ are trainable parameters.

IV. Methodology

The goal of DyHGDAT is to learn a hypergraph at each time step, where hyperedges represent group behaviors and vertices refer to agents, thus modeling the evolution of hypergraph to capture comprehensive interactions and apply the dual attention mechanism to adaptively emphasize the vertices and hyperedges with larger impacts to learn more accurate embedding of vertices.

A. Dynamic hypergraph topology inference

To capture more comprehensive high-order interactions, we consider constructing a hypergraph and inferring its topology at each time step to model the evolution of the hypergraph over time, as is shown in Fig. 2. Specifically, based on the historical trajectory embedding of agents, we calculate the degree of connectivity between vertices (agents) with the self-attention mechanism [27] as well as asymmetric convolution, and take it as a metric to group agents thus obtaining the topology of the hypergraph.

1) Feature extraction: We first extract the feature embedding of the agent's raw trajectory X and model the temporal dependency, which is formulated as follows:

$$E = MLP(X, \sigma, W_E) \tag{1}$$

where $MLPs(\cdot, \cdot, \cdot)$ donates the Multi-Layer Perceptron, σ and W_E refer to the activation function and trainable parameter respectively, then we adopt Gate Recurrent Unit (GRU) to model the temporal dependency as follow:

$$E_{\mathcal{V}}^{t_0} = GRU\left(E^{1:t_0}\right) \tag{2}$$

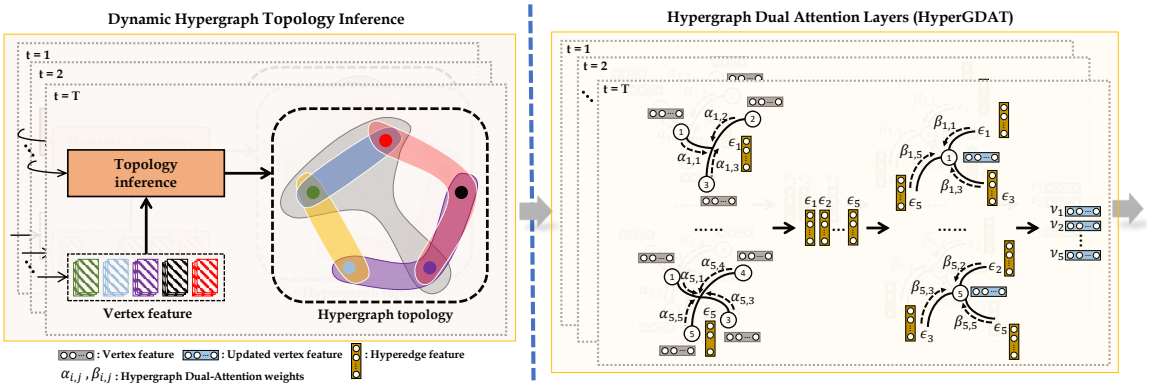

Fig. 2: Overall architecture of the DyHGDAT, which has two stages: dynamic hypergraph topology inference and the hypergraph dual attention layers, where infer the hypergraph topologies at each time step to model the evolution of hypergraph, and adaptively highlight the vertices and hyperedges with larger impacts to capture the high-order interactions, respectively.

where $E_{\mathcal{V}}^{t_0} \in R^{N\times d}$ is temporal feature embedding of agents at timestep t_0 $(1 \leq t_0 \leq T)$, and d is the dimension of embedding. And we denote the temporal dependency embedding over all time steps as $E_{\mathcal{V}} = \left[E_{\mathcal{V}}^1, E_{\mathcal{V}}^2, ..., E_{\mathcal{V}}^T\right]$.

2) Connectivity modeling: Given the temporal feature embedding $E_{\mathcal{V}}$, we donate the self-attention mechanism to calculate the attention score matrix as follows:

$$E_k = \emptyset_k\left(E_v, W_k\right) \tag{3}$$

$$E_q = \emptyset_q\left(E_v, W_q\right) \tag{4}$$

$$A = \frac{E_k \otimes E_q^T}{\sqrt{d_k}} \tag{5}$$

where $\emptyset_k\left(\cdot,\cdot\right)$ and $\emptyset_q\left(\cdot,\cdot\right)$ denote linear transformation, W_k and W_q are weights of the linear transformations. d_k is the scaled factor, and $A \in R^{N\times N}$ is the asymmetric attention matrix to reflect the directed interactions between vertices, in which $A_{i,j}$ refers to the influence of vertex i to vertex j. The rows and columns in A represent initiative and passive relations, the initiative relations refer to the influence a vertex exerts on other vertices, while the passive relations refer to the influence exerted by other vertices that a vertex endures. We consider a combination of these two relations and apply asymmetric convolution on A to get high-order interaction score to reflect the degree of connectivity among vertices, which can be formulated as:

$$M_{row}^l = Conv\left(M^{l-1}, \mathcal{K}_{(1\times S)}\right) \tag{6}$$

$$M_{col}^l = Conv\left(M^{l-1}, \mathcal{K}_{(S\times 1)}\right) \tag{7}$$

$$M^l = \sigma\left(M_{col}^l + M_{row}^l\right) + M^{(l-1)} \tag{8}$$

where $\mathcal{K}_{(1\times S)}$ and $\mathcal{K}_{(S\times 1)}$ are the asymmetric convolution kernels, σ refers to the activation function, and we set $M^0 = A$. It is worth noting that all convolution operations are padded with zeros to keep the matrix size constant. Finally, we get the connectivity degree score matrix $M_{Conn} = M^k$, k is the number of the layers of asymmetric convolution. The elements in M_{Conn} represent the degree of connection between vertices and groups, and the vertices will next be grouped based on M_{Conn}.

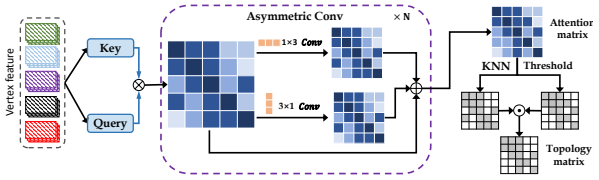

Fig. 3: Architecture of hypergraph topology inference module, which outputs topology of hypergraph.

3) Topology inference: we take M_{Conn} as metric to group agents to infer the topology of a hypergraph, highly correlated groups will be considered as hyperedges. As is shown in Fig. 3, we use the k-nearest neighborhood selection (KNN) strategy and the element-wise threshold on the rows of M_{Conn} to obtain two grouping results as $\mathcal{H}_{knn}$ and $\mathcal{H}_{Thd}$, respectively, both of which are 0-1 matrices. The final grouping result is calculated as follows:

$$\mathcal{H} = \mathcal{H}_{knn} \odot \mathcal{H}_{Thd} \tag{9}$$

and we take $\mathcal{H}$ as the topology of hypergraph. Compared to [12], [19], [20], our approach is novel from two aspects. First, we adopt the attention mechanism and asymmetric convolution to explicitly compute the degree of connectivity to achieve more accurate topological inference, while previous work used a fixed fully connected topology or obtained the topology through the similarity of agent embedding. Second, we model the evolution of the hypergraph by inferring the hypergraph topology at each time step, while previous approaches come with static hypergraph assumptions and ignore the evolution of the hypergraph over time.

B. Hypergraph dual attention network

With the hypergraph topology, we want to learn the embedding of vertices (agents) considering the correlations among vertices defined by different hyperedges (group behaviors). As mentioned earlier, our goal is to emphasize the hyperedges and vertices with larger impacts. We argue that in our task, the vertex set and the hyperedge set are from the same homogeneous domain, which enables us to

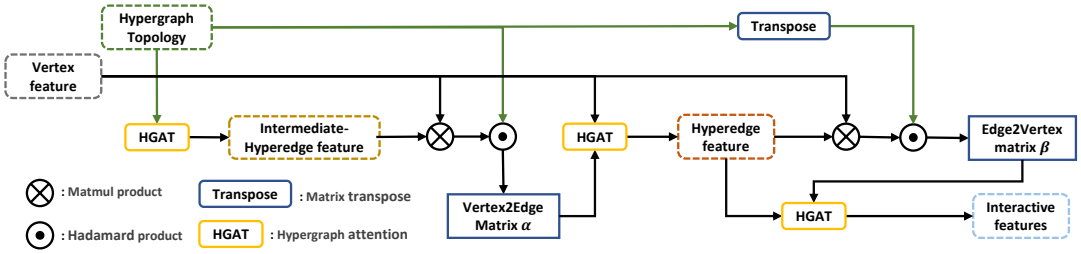

Fig. 4: The message passing paradigm for the hypergraph dual attention.

exert an attention learning module on topology matrix $\mathcal{H}$ to highlight the more impactful vertices and hyperedges. In the following, we introduce a novel hypergraph dual attention mechanism that applies the attention mechanism to learn the attention score matrix in the Vertex-to-Hyperedge phase and Hyperedge-to-Vertex phase, respectively. In addition, we design a message-passing paradigm for the hypergraph dual attention mechanism, as is shown in Fig. 4.

1) Vertex-to-Hyperedge phase: In hypergraph, information can propagate among vertices via hyperedge, which is the key factor for hypergraph representation learning. Since no predefined hyperedges are embedding, we obtain the intermediate hyperedges embedding by aggregating the vertices embedding using the topology matrix $\mathcal{H}$ as follows:

$$E_{ie} = \emptyset_{Re}\left(softmax\left(\mathcal{H}\right) \otimes E_v, W_{ie}\right) \tag{10}$$

where $softmax$ is used for normalization and all zero-inputs of $softmax$ are masked to let the model attend to the vertices connected to hyperedges. Then we apply the attention mechanism to enable the network to adaptively focus on those dominant vertices. We first calculate the attention score by intermediate hyperedges embedding and vertices embedding, and subsequently aggregate vertices embedding to corresponding hyperedges by $\mathcal{H}_{V\to E}$ which can be formulated as:

$$\mathcal{H}_{v\to e} = Softmax\left[\left(\frac{E_{ie} \otimes E_v^T}{\sqrt{d_k}}\right) \odot \mathcal{H}\right] \tag{11}$$

$$E_e = \emptyset_e\left(\mathcal{H}_{v\to e} \otimes E_v, W_e\right) \tag{12}$$

where $\odot$ denotes the element-wise multiplication.

2) Hyperedge-to-Vertex phase: To update the embedding for vertices, we aggregate the embedding from all its connected hyperedges. Similarly, we perform attention learning to model the difference in the degree of influence that hyperedges have on a vertex to emphasize the hyperedges that have a larger impact, which can be formulated as:

$$\mathcal{H}_{e\to v} = Softmax\left[\left(\frac{E_v \otimes E_e^T}{\sqrt{d_k}}\right) \odot \mathcal{H}^T\right] \tag{13}$$

$$E'_v = \emptyset_v\left(\mathcal{H}_{e\to v} \otimes E_e, W_v\right) \tag{14}$$

where $\mathcal{H}_{e\to v}$ is the attention score matrix for the hyperedge-to-vertex phase, and we update the vertices embedding using the residual connection:

$$E_v = E'_v + E_v \tag{15}$$

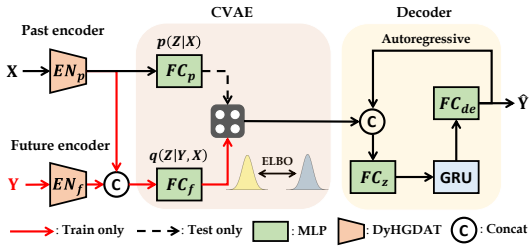

Fig. 5: Overall architecture of prediction system with dynamic hypergraph dual attention network (DyHGDAT), DyHGDAT are used as encoder in CVAEs.

Compared to [12], [23], our method is novel. We note that some vertices and hyperedges may have greater influence while others don't, and propose a novel hypergraph dual attention mechanism to explicitly emphasize the hyperedges and vertices with larger impacts, while all previous works have neglected to model the discrepancy.

C. Prediction System with DyHGDAT

We apply DyHGDAT into a simple CVAEs [33], as is shown in Fig. 5, to generate stochasticity and multimodality trajectory. CVAEs can model multiple modes in the conditional distribution of output variables Y given input X, making CVAEs suitable for modeling one-to-many mapping. The future trajectory distribution of agents can be reformulated by introducing the latent variable Z, as follows:

$$p\left(Y|X\right) = \int p\left(Y|Z,X\right) p\left(Z|X\right) dZ \tag{16}$$

where $p\left(Y|X\right)$ is posterior distribution, $p\left(Z|X\right)$ is prior distribution of Z and $p\left(Y|Z,X\right)$ refer to conditional likelihood distribution. Similar to previous work, we use the negative evidence lower bound (ELBO) as our loss function to train the network:

$$\begin{aligned}\mathcal{L}_{elbo} = &- E_{q(Z|Y,X)}\left[\log p\left(Y|Z,X\right)\right] \\ &+ KL\left(p\left(Z|X\right)||q\left(Z|Y,X\right)\right)\end{aligned} \tag{17}$$

where $q\left(Z|Y,X\right)$ refer to approximate posterior distribution, and $KL\left(\cdot \parallel \cdot\right)$ denotes the KL divergence. To generate more diverse and plausible trajectories, we adopt DLow [25] to CVAEs. CVAEs consist of an encoder and decoder, and we will introduce its module as follows.

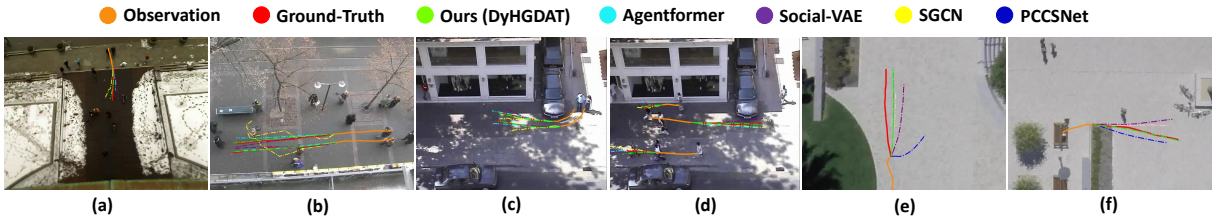

Fig. 6: Visualization of trajectory, (a),(b),(c),(d) are in the ETH/UCY, and (e),(f) are in the SDD.

Methods	Social-LSTM [1] CVPR 2016	Sophie [3] CVPR 2019	SGCN [10] CVPR 2021	PECNet [28] ECCV 2020	S-STAGE [29] ICRA 2021	GroupNet [12] CVPR 2022	Social-VAE [30] ECCV 2022	Agentformer [6] ICCV 2021	Graph-TERN [11] AAAI 2023	CVAE	DyHGDAT Our
ETH	1.09/2.41	0.70/1.43	0.63/1.03	0.65/1.13	0.44/0.77	0.46/0.73	0.47/0.76	0.45/0.75	0.42/**0.58**	0.47/0.80	**0.41**/0.61
HOTEL	0.86/1.91	0.76/1.67	0.32/0.55	0.18/0.24	0.28/0.50	0.15/0.25	0.14/0.22	0.14/0.22	0.14/0.23	0.17/0.31	**0.13/0.20**
UNIV	0.61/1.31	0.54/1.24	0.37/0.70	0.35/0.60	0.40/0.77	0.26/0.49	0.25/0.47	0.25/0.45	0.26/0.45	0.28/0.52	**0.22/0.42**
ZARA01	0.41/0.88	0.30/0.63	0.29/0.53	0.22/0.39	0.30/0.56	0.21/0.39	0.20/0.37	0.18/**0.30**	0.21/0.37	0.25/0.49	**0.17**/0.32
ZARA02	0.52/1.11	0.38/0.78	0.25/0.45	0.17/0.30	0.20/0.37	0.17/0.33	0.14/0.28	0.14/0.24	0.17/0.29	0.21/0.40	**0.13/0.24**
AVG	0.70/1.52	0.54/1.15	0.37/0.65	0.29/0.48	0.32/0.59	0.25/0.44	0.24/0.42	0.23/0.40	0.24/0.38	0.28/0.50	**0.21/0.36**

TABLE I: Comparison to SOTA models on ETH/UCY datasets, the 'AVG' means the average result over 5 subsets.

Methods	Social-GAN [2] CVPR 2018	PECNet [3] ECCV 2020	GroupNet [12] CVPR 2022	MANTRA [31] CVPR 2020	PCCSNet [32] ICCV 2021	Social-VAE [30] ECCV 2022	Graph-TERN [11] AAAI 2023	CVAE	DyHGDAT Our
ADE/FDE	27.23/41.44	9.96/15.88	9.31/16.11	8.96/17.76	8.62/16.16	8.88/14.81	8.42/14.26	10.22/18.18	**7.88/13.21**

TABLE II: Comparison to SOTA models on SDD dataset.

DyHGN	HTI	HGDAT	ETH/UCY	SDD
-	-	-	0.24/0.42	8.89/15.09
✓	-	-	0.23/0.39	8.41/14.37
✓	✓	-	0.22/0.38	8.02/13.56
✓	✓	✓	**0.21/0.36**	**7.88/13.21**

TABLE III: Ablation Studies on ETH/UCY and SDD. The ETH/UCY column reports the 'AVG' result.

1) Encoder: CVAE has two encoders, past encoder EN_p and future encoder EN_f, each of which consists of a DyHGDAT. Past encoder EN_p is used to encode multi-agent historical trajectory X to generate the prior distribution $p(Z|X)$, while future encoder EN_f is used to encode multi-agent future trajectories Y to generate the approximate posterior distribution $q(Z|Y,X)$. It is worth noting that the future encoder is only activated in the training phase.

2) Decoder: As is shown in Fig. 5, our decoder is autoregressive, outputting only one prediction at a time, then the current output will concatenate with latent variable Z and feed back into the model to produce the trajectories of the next timestep, we can obtain the predicted trajectory by:

$$\hat{y}_i^t = y_i^0 + \sum_{\tau=0}^{t} d_i^\tau \tag{18}$$

where $d_i^\tau \in R^2$ is the output of future decoder which is the relative displacement of ith agent at $t=\tau$, and y_i^0 is the last spatial coordinate of ith agent in the past trajectory X_i^-.

V. Experiment

A. Experimental setup

1) Datasets: To validate the efficacy of our proposed method, we conduct experiments on two well-established trajectory prediction datasets: the ETH/UCY datasets [35], [36] and the Stanford Drone Dataset [37]. Following the experimental setup of Social-GAN [2], we take the first 8 steps as a historical trajectory to predict the next 12 steps (future trajectory), and we use the "leave-one-out" strategy to train and test our model.

2) Metrics: We use average displacement error (ADE) and final displacement error (FDE) as evaluation metrics, and we calculate the ADE and FDE using best-of-20 predictions. ADE measures the average L-2 distance between the predicted trajectories and the ground-truth in terms of the whole trajectories, while FDE the measures L-2 distance between predicted endpoints to the ground-truth endpoints.

3) Implementation details: We train our model with Adam optimizer, the initial learning rate is 0.0005. The dimension of latent code z is 32, and the number of asymmetric convolution layers is set to 3 with kernel size $S=3$. The number of agents differs in different frames, so we set the number of hyperedges in the hypergraph equal to the agents' number, and the size of the hyperedge and k in KNN strategy is equal to the half of agents' number. The threshold in hypergraph topology inference is a trainable parameter with an initial value of 0.2. Our proposed model is implemented using PyTorch on a server running Ubuntu 18.04, with a single RTX 3080 GPU.

B. Quantitative Evaluation

1) Accuracy: We compare our method in ETH/UCY and SDD with several state-of-the-art methods, as shown in Table I and Table II. DyHGDAT with CVAE framework outperforms all the previous models. In ETH/UCY, the improvement of DyHGDAT over Grapg-TERN is 12.5% and 5.3% on average ADE and FDE, respectively. In SDD, DyHGDAT reaches a new state-of-the-art and outperforming the Graph-TERN by 6.4% in ADE and 7.4% in FDE. It is worth noting that our method performs well in the scenes with higher crowd density (UNIV), we argue that it is because DyHGDAT can capture comprehensive and accurate high-order interactions among agents in dense scenarios thus generating accurate future trajectories.

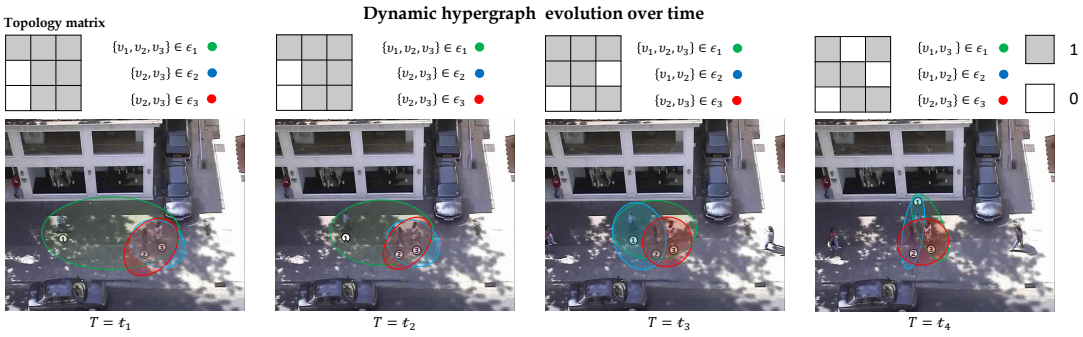

Fig. 7: Visualization of the dynamic evolution of hypergraph from $T = t_1$ to $T = t_4$.

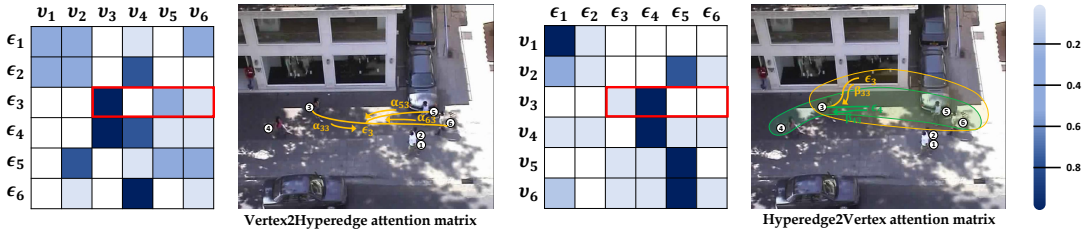

Fig. 8: Visualization of the dual attention mechanism. The left side of the figure shows the attention matrix for the Vertex-to-Hyperedge phase, while the right side of the figure shows the attention matrix for the Hyperedge-to-Vertex phase.

2) Ablation studies: Our DyHGDAT has three key components: dynamic hypergraph network (DyHGN), hypergraph topology inference (HTI), and the hypergraph dual attention mechanism. We conduct extensive ablation studies at ETH/UCY and SDD, and the results are shown in Tab III. For reference, the first row of the table shows the performance of CVAEs without any components (To be fair, Dlow is applied). Each component's contribution to our model, With all three components, DyHGDAT outperforms the baseline model in the ETH/UCY datasets by 12.5% and 14.3% on ADE and FDE respectively, while in SDD the improvement is 11.4% on ADE and 12.5% on FDE.

C. Quantitative Evaluation

1) Visualization of trajectory: Fig. 6 shows our predictions in the ETH/UCY and SDD, the observed trajectories appear in orange, while the ground-truth trajectories appear in red, and our method predictions appear in bright green. Our method can produce the most accurate predictions.

2) Visualization of hypergraph evolution: Fig. 7 shows an example of hypergraph evolution that DyHGDAT captures. The matrix at the top of the picture is the topology matrix DyHGDAT obtained by topology inference, which reveals the connections between vertices and hyperedges. It can be seen that as the relative positions among agents change, the vertices connected by the hyperedge are also changing. For example, when $T = t_3$ our model establishes a hyperedge that connects agent 1 and agent 2 as the distance between them gradually decreases. Compared with the previous works, our approaches can model the dynamic evolution of hypergraph over time, thus allowing our model to capture more comprehensive high-order interactions among agents.

3) Visualization of hypergraph attention: Fig. 8 shows the attention matrix calculated by our DyHGDAT. It can be seen that different vertices connected by a hyperedge are explicitly assigned different attention weights on the left side of Fig. 8, which enables the model to be able to highlight the dominant agents within a hyperedge. Similarly, the left side of Fig. 8 shows that different hyperedges have different degrees of impact on a vertex, and our model can emphasize the hyperedges with larger impacts. The dual attention mechanism enhances the ability of hypergraph for representation learning, which contributes significantly to the improvement of our prediction accuracy.

VI. Conclusion

This paper proposes DyHGDAT, a Dynamic Hypergraph Dual Attention Network for multi-agent trajectory prediction, which can capture comprehensive and accurate high-order interactions from two aspects: i) it extends static hypergraphs to dynamic hypergraphs to capture more comprehensive high-order interactions among agents. ii) it proposes a hypergraph dual attention mechanism to highlight the vertices and hyperedges with larger impacts to enhance the hypergraph for representation capability. We conduct extensive experiments on two well-established datasets, and the experimental results show that our approach outperforms a range of previous methods, which demonstrates the effectiveness of DyHGDAT in extracting high-order interactions among agents.

Acknowledgment

This work is partially supported by the National Natural Science Foundation of China (Grant No. 62373148) and the Science and Technology Commission of Shanghai Municipality Research (Grant No. 21JC1405300).

REFERENCES

[1] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social lstm: Human trajectory prediction in crowded spaces," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 961–971.

[2] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social gan: Socially acceptable trajectories with generative adversarial networks," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 2255–2264.

[3] A. Sadeghian, V. Kosaraju, A. Sadeghian, N. Hirose, H. Rezatofighi, and S. Savarese, "Sophie: An attentive gan for predicting paths compliant to social and physical constraints," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 1349–1358.

[4] J. Duan, L. Wang, C. Long, S. Zhou, F. Zheng, L. Shi, and G. Hua, "Complementary attention gated network for pedestrian trajectory prediction," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, no. 1, 2022, pp. 542–550.

[5] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, "Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data," in Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16. Springer, 2020, pp. 683–700.

[6] Y. Yuan, X. Weng, Y. Ou, and K. M. Kitani, "Agentformer: Agentaware transformers for socio-temporal multi-agent forecasting," in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 9813–9823.

[7] A. Mohamed, K. Qian, M. Elhoseiny, and C. Claudel, "Socialstgcnn: A social spatio temporal graph convolutional neural network for human trajectory prediction," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 14 424–14 432.

[8] C. Yu, X. Ma, J. Ren, H. Zhao, and S. Yi, "Spatio-temporal graph transformer networks for pedestrian trajectory prediction," in Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XII 16. Springer, 2020, pp. 507–523.

[9] I. Bae and H.-G. Jeon, "Disentangled multi-relational graph convolutional network for pedestrian trajectory prediction," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, no. 2, 2021, pp. 911–919.

[10] L. Shi, L. Wang, C. Long, S. Zhou, M. Zhou, Z. Niu, and G. Hua, "Sgcn: Sparse graph convolution network for pedestrian trajectory prediction," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 8994–9003.

[11] I. Bae and H.-G. Jeon, "A set of control points conditioned pedestrian trajectory prediction," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, no. 5, 2023, pp. 6155–6165.

[12] C. Xu, M. Li, Z. Ni, Y. Zhang, and S. Chen, "Groupnet: Multiscale hypergraph neural networks for trajectory prediction with relational reasoning," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 6498–6507.

[13] D. Helbing and P. Molnar, "Social force model for pedestrian dynamics," Physical review E, vol. 51, no. 5, p. 4282, 1995.

[14] J. Van Den Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in Robotics Research: The 14th International Symposium ISRR. Springer, 2011, pp. 3–19.

[15] D. Helbing, L. Buzna, A. Johansson, and T. Werner, "Self-organized pedestrian crowd dynamics: Experiments, simulations, and design solutions," Transportation science, vol. 39, no. 1, pp. 1–24, 2005.

[16] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," arXiv preprint arXiv:1412.3555, 2014.

[17] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[18] N. Bisagno, B. Zhang, and N. Conci, "Group lstm: Group trajectory prediction in crowded scenarios," in Proceedings of the European conference on computer vision (ECCV) workshops, 2018, pp. 0–0.

[19] L. Zhou, Y. Zhao, D. Yang, and J. Liu, "Gchgat: Pedestrian trajectory prediction using group constrained hierarchical graph attention networks," Applied Intelligence, vol. 52, no. 10, pp. 11 434–11 447, 2022.

[20] J. Sun, Q. Jiang, and C. Lu, "Recursive social behavior graph for trajectory prediction," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 660–669.

[21] D. Zhou, J. Huang, and B. Scholkopf, "Learning with hypergraphs: Clustering, classification, and embedding," Advances in neural information processing systems, vol. 19, 2006.

[22] P. Li and O. Milenkovic, "Submodular hypergraphs: p-laplacians, cheeger inequalities and spectral clustering," in International Conference on Machine Learning. PMLR, 2018, pp. 3014–3023.

[23] Y. Feng, H. You, Z. Zhang, R. Ji, and Y. Gao, "Hypergraph neural networks," in Proceedings of the AAAI conference on artificial intelligence, vol. 33, no. 01, 2019, pp. 3558–3565.

[24] S. Bai, F. Zhang, and P. H. Torr, "Hypergraph convolution and hypergraph attention," Pattern Recognition, vol. 110, p. 107637, 2021.

[25] Y. Yuan and K. Kitani, "Dlow: Diversifying latent flows for diverse human motion prediction," in Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IX 16. Springer, 2020, pp. 346–364.

[26] J. Jiang, Y. Wei, Y. Feng, J. Cao, and Y. Gao, "Dynamic hypergraph neural networks." in IJCAI, 2019, pp. 2635–2641.

[27] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," Advances in neural information processing systems, vol. 30, 2017.

[28] K. Mangalam, H. Girase, S. Agarwal, K.-H. Lee, E. Adeli, J. Malik, and A. Gaidon, "It is not the journey but the destination: Endpoint conditioned trajectory prediction," in Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16. Springer, 2020, pp. 759–776.

[29] S. Malla, C. Choi, and B. Dariush, "Social-stage: Spatio-temporal multi-modal future trajectory forecast," in 2021 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2021, pp. 13 938–13 944.

[30] P. Xu, J.-B. Hayet, and I. Karamouzas, "Socialvae: Human trajectory prediction using timewise latents," in European Conference on Computer Vision. Springer, 2022, pp. 511–528.

[31] F. Marchetti, F. Becattini, L. Seidenari, and A. D. Bimbo, "Mantra: Memory augmented networks for multiple trajectory prediction," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 7143–7152.

[32] J. Sun, Y. Li, H.-S. Fang, and C. Lu, "Three steps to multimodal trajectory prediction: Modality clustering, classification and synthesis," in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 13 250–13 259.

[33] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," arXiv preprint arXiv:1312.6114, 2013.

[34] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," Advances in neural information processing systems, vol. 28, 2015.

[35] A. Lerner, Y. Chrysanthou, and D. Lischinski, "Crowds by example," in Computer graphics forum, vol. 26, no. 3. Wiley Online Library, 2007, pp. 655–664.

[36] S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool, "You'll never walk alone: Modeling social behavior for multi-target tracking," in 2009 IEEE 12th

[37] A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese, "Learning social etiquette: Human trajectory understanding in crowded scenes," in Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14. Springer, 2016, pp. 549–565