Trajectory Prediction With Heterogeneous Graph Neural Network

Guanlue Li[1,2], Guiyang Luo[1], Quan Yuan[1], and Jinglin Li[1]

[1] Beijing University of Posts and Telecommunications, Beijing, China.
{liguanlue, luoguiyang, yuanquan, jlli }@bupt.edu.cn
[2] Science and Technology on Communication Networks Laboratory, Shijiazhuang, China.

Abstract. Trajectory prediction with dense traffic is a challenging task. The heterogeneity caused by multi-type of road agents complicates the mutual and dynamic relationship between agents. Besides, scene context will affect the trajectory of agents. To address the aforementioned challenges, we present a novel model named HTFNet. Specifically, we use a heterogeneous graph network to model multi-type of agents in traffic. In order to handle varying influence between nodes, interactions between nodes are modelled by a heterogeneous transformer neural network, which uses mate-relation-dependent parameters to distinguish heterogeneous attention over each edge. In addition, scene contexts are considered in multi-model destinations prediction. Through extensive experiments on Stanford Drone Dataset, the results show that our model achieves superior performance on the heterogeneous traffic dataset and produces more reasonable trajectories for different types of road agents.

Keywords: Trajectory prediction · Heterogeneous graph transformer · Multi-type agents.

1 Introduction

Predicting trajectory is an essential component for many applications. For example, autonomous driving needs accurate trajectory prediction to avoid collisions and ensure safety. When robots deliver goods in a complex environment, trajectory forecasting can help them take appropriate strategies to improve efficiency.

In a traffic scenario, there are multi-type road agents, such as cars, buses, pedestrians and bicycles. Different types of road agents increase the uncertainty of interaction effects between them. Most existing trajectory prediction works focus on one type of road agents such as pedestrians [3, 26, 6, 1] or vehicles [10, 25]. However, these methods ignore the difference in social interaction and dynamic patterns between multi-type road agents. For example, people will pay more attention to motor vehicles with greater speed and inertia than pedestrians. Hence, learning different interaction patterns is required to predict trajectories in dense traffic.

Many models divide the prediction into two steps [2, 28, 8], which predict destinations firstly, then predict the final trajectory based on a generated destination. Generative Adversarial Networks (GANs) and Variational Autoencoders are used to predict the destination distribution [9, 21]. Static scene feature need to be added in destination prediction, which prevents unrealistic predictions.

This work aims to develop a trajectory prediction model suitable for dense traffic with multi-type road agents. We follow the target-driven trajectory prediction framework and use conditional variational autoencoders (CVAEs) to predict the destination distribution. With the success of graph neural networks in processing graph-structured data, road agents can be modelled as a graph with rich relation information. We propose HTFNet, which uses a heterogeneous graph transformer network to model interactions between road agents. Meta-relation-based parameters are used to get adaptive scaling attention. We add scene information in the process of destinations and trajectories prediction, which increases the accuracy and reality of trajectories. We empirically validate our model on Stanford Drone Datasets. Experimental results show that our model significantly improves trajectory prediction tasks compared to baselines.

The contributions of this paper are summarized as follows:

- We model multi-type of agents as a dynamic heterogeneous graph and propose HFTNet to learn heterogeneous message transmission between nodes.
- In the destination and trajectory prediction process, we consider scene information and the dynamic pattern of the agents.
- Our model is evaluated in the short-term and long-term trajectory prediction tasks. The result shows that our model can produce more reasonable and accurate trajectories in complex traffic.

2 Related Work

There are considerable works on trajectory prediction for moving agents. Many approaches rely on recurrent neural networks (LSTMs or GRUs) to exploit temporal dependencies of time series. Many models take into account the interaction between road agents. Alahi et al. [1] introduces social pooling to pool nearby pedestrians' hidden features. Deo et al. [10] use convolutional social pooling to improve the pooling process. Another relevant work is STGAT by Huang et al. [15]. STGAT treats each agent as a graph node and exploits the graph attention network to share information across different pedestrians. Some algorithms consider heterogeneous data: Trajectron++ by Salzmann et al. [24] accounts for multiple interacting agents from heterogeneous input data and produces dynamically-feasible trajectory forecasts. TraPHic by Chandra et al. [7] uses a hybrid network LSTM-CNN to predict trajectories and take into account heterogeneous interactions. The dynamics of a bus-pedestrian interaction differ significantly from a pedestrian-pedestrian or a car-pedestrian. In order to process dense, heterogeneous traffic scenarios in-depth, we model the interactions among different types of agents by using a dynamic heterogeneous graph. Some methods also use reinforcement learning to model the interaction and communication between agents [17, 20].

Graph Neural Networks (GNNs) are aimed to process graph structured data and use message passing between the nodes to capture information from its neighbourhood with arbitrary depth. GNN has been used in many fields such as biology [12], traffic forecasting [19] and mobile networks [18]. The graph convolutional neural network can be divided into two categories: spectral domain and non-spectral domain. Spectral approaches represent the graph as spectral embedding based on adjacency matrices, while non-spectral approaches use convolutions directly on the graph based on groups of spatial neighbours. A heterogeneous graph is defined as a graph have several kinds of nodes and edges. The heterogeneous graph embedding mainly focuses on meta-relation, which utilizes the nodes' paths to model the context of a node. Meta-relation can group the neighbours according to their node types and distances. HAN by Wang et al. [27] proposed the heterogeneous graph attention network, which utilizes meta-relation to model node level and semantic attention and learns the weights of different neighbours. HGT by Hu et al. [14] uses transformer-like self-attention architecture for learning node representation.

3 Model Design

In this section, we introduce our HTFNet for trajectory prediction with multiple types of agents. Our approach is visualized in Fig. 1. There are three key components 1) Feature Encoder, 2) Destination Generator, and 3) Interaction and Prediction Module. We begin the section by describing the problem definition. Then we present details on how the proposed components are adapted to the task.

3.1 Problem Formulation

Problem Setup Multi-agent trajectory prediction is a task of forecasting the future states of agents. The inputs of model are the historical state of N agents $\boldsymbol{X} = [\boldsymbol{x}^{-t_h}, ..., \boldsymbol{x}^{-1}, \boldsymbol{x}^0]$ and scene context S in the time period $[-t_h, 0]$, where $\boldsymbol{x}^t = (x_1^t, x_2^t, ...x_N^t)$ is the joint state of N agents at time t. The goal of model is predicting the position of N agents $\boldsymbol{Y} = (\boldsymbol{Y}^1, \boldsymbol{Y}^2, ...\boldsymbol{Y}^{t_p})$ in the future time period $[1, t_p]$.

This paper is focused on a complex traffic scenario involving multi-type road agents, such as cars, pedestrians and bicycles. The scenario can be modeled by dynamic heterogeneous graph denoted as a series of snapshots $\{G\}_{-t_h}^{t_p}$, where nodes represent road agents and edges represent their interactions. The graph G at each time t represented as $G_t = (\mathcal{V}_t, \mathcal{E}_t, \mathcal{A}, \mathcal{R})$, where the nodes $\mathcal{V}_t$ and edges $\mathcal{E}_t$ change when the dynamic graph evolving. $\mathcal{A}$ and $\mathcal{R}$ are node and edge type sets respectively. Each node $v \in \mathcal{V}_t$ and each edge $e \in \mathcal{E}_t$ are associated with their type by mapping functions $\gamma(v) : \mathcal{V}_t \to \mathcal{A}$ and $\lambda(e) : \mathcal{E}_t \to \mathcal{R}$.

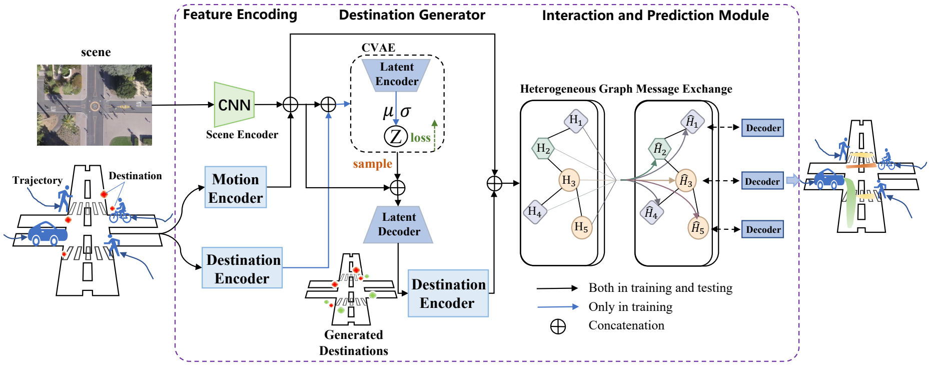


Fig. 1. Overview of the model architecture. Our model consists of three components: 1) Feature Encoder, 2) Destination Generator, and 3) Interaction and Prediction Module. The Destination Generator Module combines the motion feature of the historical trajectories and scene context to infer the destination distribution. The Interaction and Prediction Module uses the HGT network to exchange features between nodes.

3.2 Feature Extraction Module

The input of the model includes 2D location series of agents. We use a fully connected layer (FCL) with Relu activation as the motion encoder to extract temporal information of history state F_x:

$$\boldsymbol{F}_x = \mathrm{MLP}_x(\{\boldsymbol{x}_k\}_{-t_h}^{0}), \tag{1}$$

where $\{x_k\}_{-t_h}^{0}$ is the 2D location series of agent k. We follow target-driven framework that predicts the destination distribution firstly. Therefore we need to extract the trajectory endpoint for destination generative module. We also use a FCL and Relu activation as the destination encoder:

$$\boldsymbol{F}_d = \mathrm{MLP}_d(\boldsymbol{D}_k). \tag{2}$$

where $D_k = x_k^{t_p}$ is the ground-truth destination of agent k. Trajectories are significantly oriented by scene context. The objective of the scene encoder is to detect the scene edge information (e.g., sidewalk, boundaries and buildings). We use the hand-designed convolutional neural network (CNN) to extract visual features F_s:

$$\boldsymbol{F}_s = \mathrm{CNN}(\boldsymbol{S}_t). \tag{3}$$

where $\boldsymbol{S}_t$ is the image of traffic environment at time t.

3.3 Scene Context Aware Destinations Prediction

In this model, CAVE is used to learn the destination distribution. The goal of destination generator module is to model the destination distribution $p_\theta(\boldsymbol{D}|\boldsymbol{X}, \boldsymbol{S})$

conditioned on history motion $\boldsymbol{X}$ and contextual information $\boldsymbol{S}$. To consider the stochasticity of destination in complex traffic scenarios, latent variables $\boldsymbol{Z}$ are introduced. The future destination distribution of agents can be represented as:

$$p(\boldsymbol{D}_k|\boldsymbol{X}_k,\boldsymbol{S}_k)=\int p_\theta(\boldsymbol{D}_k|\boldsymbol{Z}_k,\boldsymbol{X}_k,\boldsymbol{S}_k)p_\nu(\boldsymbol{Z}_k|\boldsymbol{X}_k,\boldsymbol{S})d\boldsymbol{Z}_k \tag{4}$$

where $\boldsymbol{D}_k$ is the ground-truth destination of agent k. $\boldsymbol{Z}_k,\boldsymbol{X}_k$ is the latent intent and historical trajectory of agent k, respectively. $\boldsymbol{S}_k$ is the scene context in this period of time. Deep neural network are used to approximate prior network $p_\nu(\boldsymbol{Z}_k|\boldsymbol{X}_k,\boldsymbol{S})$ and decoder network $p_\theta(\boldsymbol{D}_k|\boldsymbol{Z}_k,\boldsymbol{X}_k,\boldsymbol{S}_k)$, where ν and θ denote the parameters of corresponding networks. The generative process of $\boldsymbol{D}_k$ is:
1. Sample a latent variable z from the prior network $p_\nu(\boldsymbol{Z}_k|\boldsymbol{X}_k,\boldsymbol{S}_k)$.
2. Generate destinations $\widehat{\boldsymbol{D}}_k$ through the response decoder $p_\theta(\boldsymbol{D}_k|\boldsymbol{Z}_k,\boldsymbol{X}_k,\boldsymbol{S}_k)$.

The goal of CVAE is maximizing the conditional log likelihood $\log p(\boldsymbol{D}_k|\boldsymbol{X}_k,\boldsymbol{S}_k)$, which can be trained by maximizing the variational lower bound of the conditional log likelihood. Finally, the loss function of the destination generator can be represented as:

$$\begin{aligned}L_d(\theta,\nu;\boldsymbol{D},\boldsymbol{X},\boldsymbol{S})&=\mathrm{KL}\left(q_\nu(\boldsymbol{Z}|\boldsymbol{D},\boldsymbol{X},\boldsymbol{S})||p_\theta(\boldsymbol{S}|\boldsymbol{X},\boldsymbol{S})\right)\\&\quad-E_{q_\nu(\boldsymbol{Z}|\boldsymbol{D},\boldsymbol{X},\boldsymbol{S})}[\log p_\theta(\boldsymbol{D}|\boldsymbol{Z},\boldsymbol{X},\boldsymbol{S})].\end{aligned} \tag{5}$$

3.4 Heterogeneous Graph Message Exchange

In a dense and complex traffic scene, different types of road agents have various types of interactions. In order to transmit information between them, we consider each road agent as a node of the graph and use a heterogeneous graph transformer network (HGTNet) to exchange messages. Our HGTNet is based on Heterogeneour Graph Transformer architecture [14]. HGTNet allows for aggregating information from neighbours by assigning different attention to different types of nodes. It also puts different attention within the same type. For example, vehicles and bicycles extract different information from the road agent in front and sides of them to avoid collisions.

There are two stages in the HGTNet. Firstly, the heterogeneous multi-head attention mechanism is used to calculate attention and messages. Then we use the heterogeneous message passing framework to exchange information between nodes. HGTNet is constructed by stacking HGT layers. The structure of a single HGT layer is shown in Fig. 2. We concatenate motion feature F_x, scene feature F_s and generated destination feature F_{gd} as the input of the Heterogeneous Message Exchange Module:

$$\boldsymbol{H}=\oplus(\boldsymbol{F}_x,\boldsymbol{F}_{gd},\boldsymbol{F}_s), \tag{6}$$

We map trajectory hidden feature of target node k and source node s into Query vector and Key vector, respectively. Then calculate their dot product as attention. For a meta relation $<\gamma(s),\lambda(e),\gamma(k)>$, the influence from source

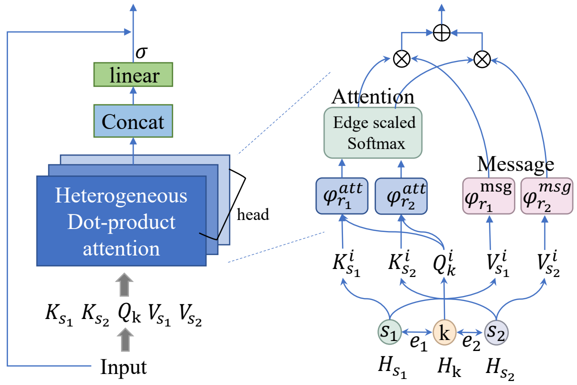


Fig. 2. Overview of architecture of Heterogeneous Graph Transformer

node s to the target node k are calculated by a meta-relation-based attention network:

$$\boldsymbol{K}^i(s) = K^i_{\gamma(s)}(\boldsymbol{H}^{l-1}[s]), \tag{7}$$

$$\boldsymbol{Q}^i(k) = Q^i_{\gamma(k)}(\boldsymbol{H}^{l-1}[k]), \tag{8}$$

$$Att^{h_i}(s,e,k) = \boldsymbol{K}^i(s)\boldsymbol{Q}^i(k)^T \boldsymbol{\varphi}^{att}_{<\gamma(s),\lambda(e),\gamma(k)>}, \tag{9}$$

where $Att^{h_i}(s,e,k)$ means one of multi-head attention. $\boldsymbol{H}^l$ means the output of l-th HGT layer, which is also the input of the $(l+1)$-th layer. In order to distinguish different meta relation between attention, a matrix $\varphi^{att}_{<\gamma(s),\lambda(e),\gamma(k)>}$ for meta relation is used to donate the difference. Then we concatenate different representations of attention and make them through the softmax procedure:

$$ATT_{HGT}(s,e,k) = \underset{\forall s \in N(k)}{Softmax}\Big(\underset{i\in[1,h]}{\Vert} Att^{h_i}(s,e,k)\Big). \tag{10}$$

After we get the multi-head attention, the message passing process can be computed in a similar way. We use a matrix $\boldsymbol{\varphi}^{msg}_{<\gamma(s),\lambda(e),\gamma(k)>}$ to distinguish different mate relation. Then all heads of message are aggregated:

$$Msg^{hi}(s,e,k) = M^i_{\gamma(s)}(\boldsymbol{H}^{l-1}[s])\boldsymbol{\varphi}^{msg}_{<\gamma(s),\lambda(e),\gamma(k)>}, \tag{11}$$

$$MSG_{HGT}(s,e,k) = \underset{i\in[1,h]}{\Vert} Msg^{h_i}(s,e,k). \tag{12}$$

As show in Fig. 2, $\boldsymbol{\varphi}^{att}_r$ and $\boldsymbol{\varphi}^{msg}_r$ denote the meta-relation-based attention and message. After getting heterogeneous multi-head attention and message, we need to aggregate them to get the final target node feature representation:

$$\widetilde{H}^l[k] = \sum_{\forall s \in N(k)} (ATT_{HGT}(s,e,k) \cdot MSG_{HGT}(s,e,k)), \tag{13}$$

following the residual connection, the output of the l-th layer is

$$H^{l+1}_{[k]} = \sigma(\boldsymbol{A}_{\gamma(k)} \widetilde{H}^l_{[k]}) + H^l_{[k]}, \tag{14}$$

$\boldsymbol{A}_{\gamma(k)}$ is a linear function mapping target note's vector back to its node type-specific distribution.

3.5 Trajectory Prediction

The final trajectory feature $\boldsymbol{H}$ of each node is passed through the prediction decoder to get the future trajectory. We use a FCL as the trajectory generator:

$$\{\boldsymbol{Y}_k\}_1^{t_f} = MLP_y(\boldsymbol{H}_k). \tag{15}$$

To train the full model HTFNet, we use the following losses:

$$L = L_d(\theta, \nu; \boldsymbol{D}, \boldsymbol{X}, \boldsymbol{S}) + \|\widehat{\boldsymbol{Y}} - \boldsymbol{Y}\|^2, \tag{16}$$

where $L_d(\theta, \nu; \boldsymbol{D}, \boldsymbol{X}, \boldsymbol{S})$ measures destination error and $\|\widehat{\boldsymbol{Y}} - \boldsymbol{Y}\|^2$ measures how far the generated trajectories from the ground truth.

4 Experiments

4.1 Dataset

We conduct experiments on Stanford Drone Datasets (SDD) [22]. SDD is a heterogeneous dataset which consists of the following road agents categories: pedestrians, skateboarders, bikers, cars, carts and buses. To make the category more general, we combine car, cart and bus into one type and denote as the vehicle. In total, there are eight unique top view scenes recorded by drone, 60 videos and 10300 unique trajectories. We follow the dataset split defined in TrajNet benchmark [4] which used in prior work [23, 5].

4.2 Experimental Settings

We use the Average Displacement Error (ADE) and Final Displacement Error (FDE) as performance metrics. The error is measured in the pixel space. We conduct the trajectory prediction task in three different time steps. In detail, We sample the data at 25fps. The length of input sequence is $t_p = 8$ (3.2 seconds) while the length of output sequence is $t_f = 12, 24, 36$ (4.8, 9.6, 14.4 seconds). For the adjacent matrix, we build a binary adjacency matrix, which based on spatial and temporal correlation. $\mathcal{E}_{i,j} = 1$ if the spatial and temporal correlations are satisfied between agent i and agent j:

$$\begin{gathered}\sqrt{(\boldsymbol{x}_i - \boldsymbol{x}_j)^2} \leq \delta_d, \\ \min(\min|t^i_{begin} - t^j_k|, \min|t^i_{end} - t^j_k|) \leq \delta_t,\end{gathered} \tag{17}$$

where δ_t and δ_d is the spatial and temporal threshold values, respectively.

The scene images are downsampled and resized to the same size. The entire network are trained end to end by ADAM optimizer and we use 2 layers HGT network to handle interaction between road agents.

4.3 Baselines

We compare the performance of our proposed model with the following baselines.
Social GAN [13]: In this approach SeqtoSeq model are used to encode motion histories and predict future trajectories. The outputs of LSTM are the generator of GAN. The diverse predictions are evaluated against with ground truth by the discriminator.
DESIRE [16]: This model use CAVE to generate diverse set of hypothetical future trajectories. Then use a scoring-regression module rank every prediction. A feedback mechanism further increases the prediction accuracy.
SoPhie [23]: This model predicts future trajectories based on GAN and consider two sources of information, which are history trajectories and scene context information. This model proposes physical attention and social attention to model interaction between agents.
CF-VAE [5]: This model use Conditional Flow Variational Autoencoder(CF-VAE) to learning multi-modal trajectories distributions. And it also proposed posterior regularization and condition regularization to stabilize training.
P2TIRL [11]: This model use MaxEnt IRL to infer goals and paths by learning rewards and it also defined coarse 2-D grid over the scene to predict trajectories.
PECNET [21]: This model first use CVAE to predict multi-modality destination of road agents. Then predict trajectories based on these destinations.

4.4 Quantitative Evaluation

We conduct trajectory prediction with different future time steps form 12 to 36. Table 1 shows the ADE and FDE values of our proposed method against the baselines on SDD. Our proposed HTFNet shows consistent lowest errors for different future time periods compared to the prior approaches. HTFNet outperforms PECNet by 1.2%, 3.4% and 3.8% on ADE in three time steps, respectively. This can be reasonably expected, since we add heterogeneous interaction between different types of road agents and static scene context, which help to find the reasonable destination.

We show the results of different types of agents in Table 2. Our method outperforms Social-GAN in all types of agents. Furthermore, our model has

Table 1. Different time steps comparison of recent methods on SDD. We report two metrics: minADE and minFDE of the trajectory with least error among K=20 predicted trajectories. The units of ADE/FDE are pixels.

Methods	12-step		24-step		36-step	
	min ADE	min FDE	min ADE	min FDE	min ADE	min FDE
Social-GAN(k=20)	27.25	41.44	56.28	123.39	114.87	267.40
DESIRE(k=5)	19.25	34.05	49.25	111.78	99.82	233.45
SoPhie(k=20)	16.27	29.38	36.05	81.14	79.09	172.47
CF-VAE(k=20)	12.60	26.30	33.89	68.66	73.09	138.47
P2TIRL(k=20)	12.58	22.07	24.07	42.40	53.47	93.05
PECNET(k=20)	9.96	15.88	24.88	43.51	53.71	94.71
HTFNet(ours)(k=20)	**9.84**	**15.76**	**24.01**	**42.46**	**52.80**	**89.46**

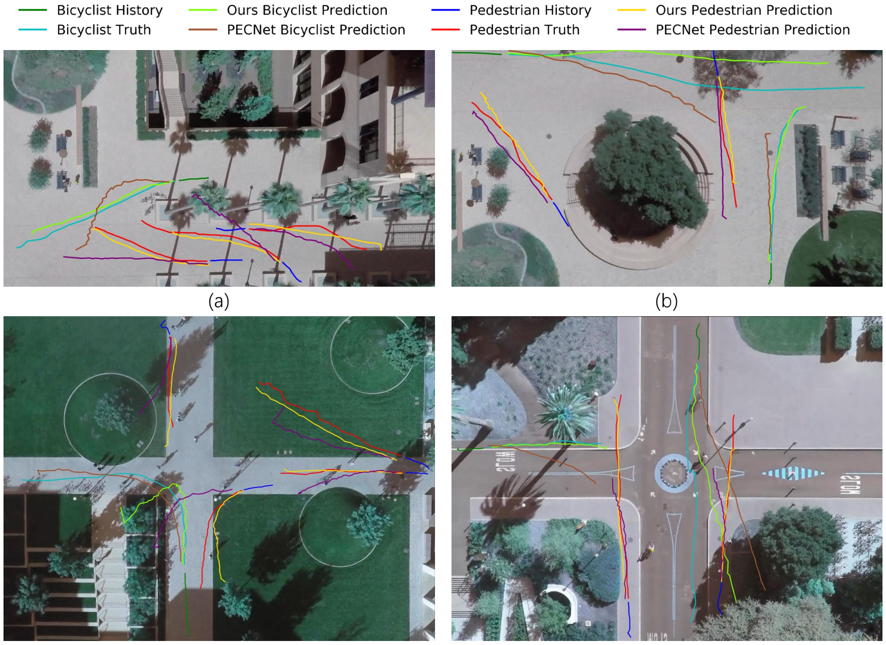
better performance in pedestrian, bicycle and vehicle categories compared to PECNet. This is because these categories have more complex interaction with scene context and other road agents. Our model can predict more reasonable trajectories through the heterogeneous transformer network.

Table 2. Performance of different types of road agents. minADE and minFDE are the least error among K=20 predicted trajectories.

Metric	type	Social-GAN	PECNet	HTFNet(ours)
minADE	pedestrian	24.66	9.57	**9.03**
	bicycle	312.33	311.34	**303**
	vehicle	204.26	177.72	**160.89**
	Skater	69.75	**39.25**	51.34
	average	27.25	9.96	**9.84**
minFDE	pedestrian	36.69	15.02	**14.72**
	bicycle	589.77	578.82	**569.53**
	vehicle	254.51	264.36	**167.67**
	Skater	92.55	**70.04**	83
	average	41.44	15.88	**15.76**

4.5 Ablation Study

We further conduct ablation studies to investigate the contribution of key technical components in our method. The ablation results are summarized in Table 3. We investigate the role of (1) heterogeneous interaction, (2) scene context and (3) destination prediction, and we denote the corresponding variants as ”w/o HI”, ”w/o SC” and ”w/o DP”. We can see that three variants lead to worse performance compared to our proposed method. After removing this heterogeneous interaction learning, the performance of the model has dropped since the different interactions between road agents are not considered. After removing the scene context feature in destination and trajectories prediction, the model


Bicyclist History
Bicyclist Truth
Ours Bicyclist Prediction
PECNet Bicyclist Prediction
Pedestrian History
Pedestrian Truth
Ours Pedestrian Prediction
PECNet Pedestrian Prediction

(a) (b)

(c) (d)

Fig. 3. Trajectory visualization of our HTFNet and PECNet

achieves 14.72 on ADE metric and 25.22 on FDE metric. This shows that extracting scene context has a certain effect on the accuracy of the model. The multi-model destination prediction also improves the performance of ADE and FDE metrics.

Table 3. Ablation study on effectiveness of heterogeneous interaction (HI), scene context (SC) and destination prediction (DP).

	w/o HT	w/o SC	w/o DP	Ours (HTFNet)
ADE	16.22	14.72	13.04	9.84
FDE	30.90	25.22	26.84	15.76

4.6 Qualitative Evaluation

Trajectory Visualization In Fig. 3, we visualize predicted trajectories of our method and PECNet. We can see that our model provides significant improvement especially in long-range trajectory prediction. In Fig. 3(a)(b), The distribution of trajectories and destination points of our model will avoid obstacles

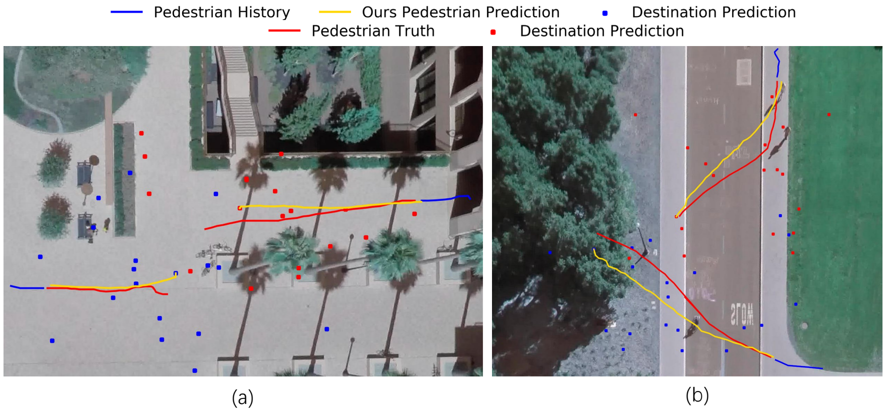


Fig. 4. Destination visualization of HTFNet

and buildings. In Fig. 3 (c)(d), our model can extract the contour of the scene, and then follow the interaction with the scene and other road agents.

Destination Visualization Fig. 4 show the predicted destination results obtained by our model. We use blue and red points to denote the sampled destination by CVAE. We can see that our model captures the multi-modality of the trajectory by destination distribution. In Fig. 4 (a), we can see that the distribution of destination depends on the scene context and history trajectory. In Fig. 4 (b), the prediction shows the pedestrians choose to cross the road.

5 Conclusion

In this work, we propose a model for motion prediction. This model first generates possible future destinations and then predicts trajectories based on multi-model destinations. We add scene information and dynamic pattern in the forecasting process. In order to model different interactions between different types of road agents, we use HGT to model their interactions with each other. We evaluate our model on a heterogeneous traffic dataset and prove that our model can predict a reasonable and accurate future trajectory.

6 Acknowledgments.

This work was supported in part by the Natural Science Foundation of China under Grant 61876023 and in part by the Foundation of Science and Technology on Communication Networks Laboratory.

References

1. Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 961–971, 2016.
2. Stefano V Albrecht, Cillian Brewitt, John Wilhelm, Balint Gyevnar, Francisco Eiras, Mihai Dobre, and Subramanian Ramamoorthy. Interpretable goal-based prediction and planning for autonomous driving. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1043–1049. IEEE, 2021.
3. Javad Amirian, Jean-Bernard Hayet, and Julien Pettré. Social ways: Learning multi-modal distributions of pedestrian trajectories with gans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.
4. Stefan Becker, Ronny Hug, Wolfgang Hübner, and Michael Arens. An evaluation of trajectory prediction approaches and notes on the trajnet benchmark. *CoRR*, abs/1805.07663, 2018.
5. Apratim Bhattacharyya, Michael Hanselmann, Mario Fritz, Bernt Schiele, and Christoph-Nikolas Straehle. Conditional flow variational autoencoders for structured sequence prediction. *arXiv preprint arXiv:1908.09008*, 2019.
6. Bruno Brito, Hai Zhu, Wei Pan, and Javier Alonso-Mora. Social-vrnn: one-shot multi-modal trajectory prediction for interacting pedestrians. *arXiv preprint arXiv:2010.09056*, 2020.
7. Rohan Chandra, Uttaran Bhattacharya, Aniket Bera, and Dinesh Manocha. Traphic: Trajectory prediction in dense and heterogeneous traffic using weighted interactions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8483–8492, 2019.
8. Luigi Filippo Chiara, Pasquale Coscia, Sourav Das, Simone Calderara, Rita Cucchiara, and Lamberto Ballan. Goal-driven self-attentive recurrent networks for trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2518–2527, 2022.
9. Patrick Dendorfer, Aljosa Osep, and Laura Leal-Taixé. Goal-gan: Multimodal trajectory prediction based on goal position estimation. In *Proceedings of the Asian Conference on Computer Vision*, 2020.
10. Nachiket Deo and Mohan M Trivedi. Convolutional social pooling for vehicle trajectory prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1468–1476, 2018.
11. Nachiket Deo and Mohan M Trivedi. Trajectory forecasts in unknown environments conditioned on grid-based plans. *arXiv preprint arXiv:2001.00735*, 2020.
12. Octavian-Eugen Ganea, Xinyuan Huang, Charlotte Bunne, Yatao Bian, Regina Barzilay, Tommi S. Jaakkola, and Andreas Krause. Independent se(3)-equivariant models for end-to-end rigid protein docking. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.
13. Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2255–2264, 2018.
14. Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. Heterogeneous graph transformer. In *Proceedings of The Web Conference 2020*, pages 2704–2710, 2020.

15. Yingfan Huang, Huikun Bi, Zhaoxin Li, Tianlu Mao, and Zhaoqi Wang. Stgat: Modeling spatial-temporal interactions for human trajectory prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6272–6281, 2019.
16. 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 the IEEE Conference on Computer Vision and Pattern Recognition*, pages 336–345, 2017.
17. G. Luo, H. Zhang, H. He, J. Li, and F. Y. Wang. Multiagent adversarial collaborative learning via mean-field theory. *IEEE Transactions on Cybernetics*, pages 1–14, 2020.
18. Guiyang Luo, Quan Yuan, Jinglin Li, Shangguang Wang, and Fangchun Yang. Artificial intelligence powered mobile networks: From cognition to decision. *IEEE Network*, pages 1–8, 2021.
19. Guiyang Luo, Hui Zhang, Quan Yuan, Jinglin Li, and Fei-Yue Wang. Estnet: Embedded spatial-temporal network for modeling traffic flow dynamics. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–12, 2022.
20. Guiyang Luo, Haibo Zhou, Nan Cheng, Quan Yuan, Jinglin Li, Fangchun Yang, and Xuemin Shen. Software-defined cooperative data sharing in edge computing assisted 5g-vanet. *IEEE Transactions on Mobile Computing*, 20(3):1212–1229, 2021.
21. 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 *European Conference on Computer Vision*, pages 759–776. Springer, 2020.
22. Alexandre Robicquet, Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Learning social etiquette: Human trajectory understanding in crowded scenes. In *European conference on computer vision*, pages 549–565. Springer, 2016.
23. Amir Sadeghian, Vineet Kosaraju, Ali Sadeghian, Noriaki Hirose, Hamid Rezatofighi, and Silvio Savarese. Sophie: An attentive gan for predicting paths compliant to social and physical constraints. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1349–1358, 2019.
24. Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. *arXiv preprint arXiv:2001.03093*, 2020.
25. Zihao Sheng, Yunwen Xu, Shibei Xue, and Dewei Li. Graph-based spatial-temporal convolutional network for vehicle trajectory prediction in autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 2022.
26. Liushuai Shi, Le Wang, Chengjiang Long, Sanping Zhou, Mo Zhou, Zhenxing Niu, and Gang Hua. Sgcn: Sparse graph convolution network for pedestrian trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8994–9003, 2021.
27. Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. Heterogeneous graph attention network. In *The World Wide Web Conference*, pages 2022–2032, 2019.
28. Hang Zhao, Jiyang Gao, Tian Lan, Chen Sun, Benjamin Sapp, Balakrishnan Varadarajan, Yue Shen, Yi Shen, Yuning Chai, Cordelia Schmid, et al. Tnt: Target-driven trajectory prediction. *arXiv preprint arXiv:2008.08294*, 2020.