MRGTraj: A Novel Non-Autoregressive Approach for Human Trajectory Prediction

Yusheng Peng, *Graduate Student Member, IEEE*, Gaofeng Zhang, Jun Shi, Xiangyu Li, Liping Zheng

***Abstract*—Forecasting human trajectory is an essential technology in intelligent surveillance systems, robot navigation systems, autonomous driving systems, etc. Most of the trajectory prediction models based on RNN and Transformers use autoregressive methods to generate future trajectories, which may accumulate displacement errors and are inefficient for training and testing. To address these problems, we propose a novel decoder named MRG decoder, which introduces a Mapping-Refinement-Generation structure to generate trajectory in a non-autoregressive manner. Furthermore, we design the MRGTraj trajectory prediction model based on the proposed MRG decoder. Firstly, we employ a Transformer as an encoder to extract encoded features from the past trajectory. Secondly, we introduce an interaction-aware latent code generator to learn a Gaussian distribution from the social context among pedestrians for latent code sampling. Finally, we feed the encoded features to the MRG decoder and sample the latent code multiple times from the learned Gaussian distribution, providing additional inputs to the MRG decoder to generate multiple socially acceptable future trajectories. Experimental results on two public datasets, ETH and UCY, validate the effectiveness of the MRGTraj model. Besides, the MRGTraj model achieves superior prediction performance, with improvements of 13.21% on FDE metrics and a 71.29% speed-up compared to state-of-the-art models. The code is available at https://github.com/wisionpeng/MRGTraj.**

***Index Terms*—Trajectory prediction, Non-autoregressive, Temporal mapper, Social refiner, Social interaction.**

I. Introduction

Accurately forecasting the future paths of pedestrians is of utmost importance for autonomous moving platforms, such as social robots [1]–[4] and self-driving cars [5]–[7]. For instance, social robots rely on observing the movements of pedestrians in their vicinity and predicting their future trajectories to plan their own paths safely. The task of pedestrian trajectory prediction involves predicting the future trajectory based on the given historical trajectory. However, this task is highly challenging due to the complex and subtle social interactions among pedestrians, as well as the inherent uncertainty in pedestrian movement.

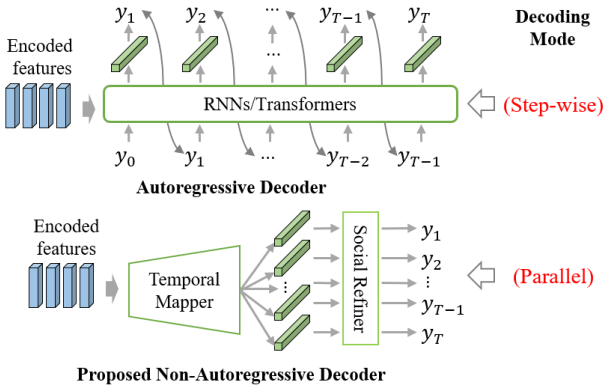


Fig. 1. Comparison illustration between the classical autoregressive decoder and the proposed non-autoregressive decoder. The autoregressive decoder generates predictions through a step-wise decoding operation from the encoded features. It predicts the future positions relying on the previously generated positions at each time step. In contrast, the proposed non-autoregressive decoder obtains decoding features for all future time steps in a single step using a temporal mapper from the encoded features. It generates predictions of all time steps simultaneously in a parallel manner.

To achieve this goal, researchers commonly employ models based on the sequence-to-sequence (Seq2Seq) architecture, which utilizes an encoder to extract encoded features from the past trajectory and a decoder to generate the future trajectory. In early years, recurrent neural networks such as Long Short-Term Memory (LSTM) were frequently used as encoders and decoders in trajectory prediction models [8]–[12]. However, in recent years, the Transformer model with a Seq2Seq structure has gained popularity for designing such models [13]–[17]. The decoder of these models, including recurrent neural networks or Transformers, belongs to the category of autoregressive decoding. As illustrated in Fig. 1, the autoregressive decoder performs predictions in a recursive manner. At each time step, the decoder cell takes the output from the previous time step as the current input to capture the hidden state and map it to a specific position. The design of these autoregressive models is independent of the time steps for observation and prediction. Therefore, a trained model can generate prediction trajectories of specified time steps according to the requirements without the need for retraining the model. This advantage facilitates the deployment and application of these models in online prediction systems. However, autoregressive decoders generally have two limitations: (i) the step-wise mode leads to the propagation and

This work was supported in part by the National Natural Science Foundation of China under Grant 61972128 and Grant 61906058 and also in part by the Fundamental Research Funds for the Central Universities of China under Grant PA2021KCPY0050 and Grant JZ2022HGTB0285. *(Corresponding author: Liping Zheng.)*

Yusheng Peng, Xiangyu Li are with the School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230601, China (e-mail: wisionpeng@mail.hfut.edu.cn).

Gaofeng Zhang, Jun Shi are with the School of Software, Hefei University of Technology, Hefei 230601, China (e-mail: g.zhang@hfut.edu.cn; juns@hfut.edu.cn).

Liping Zheng is with the School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230601, China, and also with Anhui Province Key Laboratory of Industry Safety and Emergency Technology, Hefei University of Technology, Hefei 230601, China. (e-mail: zhenglp@hfut.edu.cn).

accumulation of prediction errors from earlier time steps to later time steps, and (ii) the process cannot be parallelized, requiring sequential generation of predictions. To address these issues, we propose a novel non-autoregressive decoder enables parallel generation of future trajectories, as illustrated in Fig. 1. To achieve parallelism, we introduce a temporal mapper that directly captures the decoding features of all future time steps in a single step from the encoded features. This approach allows for later refinement and generation operations of all time steps to be performed in parallel, resulting in improved inference efficiency. Moreover, by obtaining decoding features for all time steps simultaneously and independently, we mitigate the error accumulation issues that arise from the step-wise decoding process used in autoregressive decoders.

In the existing works, there are several non-autoregressive prediction models, predominantly based on Convolutional Neural Networks (CNNs), such as Social-STGCNN [18] and its variants AST-GCN [19], Att-GCNN [20], SGCN [21], DMRGCN [22], etc. As illustrated in the top row of Fig. 2, these models expand input graph embedding in time-space through the Time-Extrapolator Convolutional Neural Network (TXP-CNN). They generate a binary Gaussian distribution and sample a single future trajectory from it. However, the prediction performance of these methods is generally unsatisfactory. This is mainly because the convolution kernels of the TXP-CNN can only capture local time consistency, and the binary Gaussian distribution fails to capture the intricate distribution of future trajectories. To address these limitations, we propose the MRGTraj model, which achieves improved prediction performance, as shown in the bottom row of Fig. 2. To effectively model the global time consistency between past and future time steps, we introduce temporal mapper using a Multilayer Perceptron (MLP) that captures the decoding features for all future time steps from the encoded features. By utilizing the temporal mapper, the decoding features of all future time steps are obtained in a single step, and the fully connected operations between the MLP neurons successfully capture global time consistency information. Additionally, we introduce a latent code generation module to learn an multivariate Gaussian distribution and sample latent codes from it. These latent codes are then concatenated with the decoding features to generate the future trajectory. This approach effectively captures the intricate distribution of future trajectories. Moreover, the latent codes sampled from this distribution serve as better guidance for the model to generate predictions that are closer to the ground-truth.

In this work, we propose a novel non-autoregressive decoder called the MRG decoder, which consists of three components: a temporal mapper, a social refiner, and a generator. Firstly, the temporal mapper takes the encoded features of the past trajectory as input and generates the decoding features for all future time steps in a single step. Next, the social refiner refines these decoding features by modeling the interaction among pedestrians. Finally, the generator maps the refined decoding features to a future trajectory. To incorporate the MRG decoder into a comprehensive model, we design the MRGTraj model, utilizing a Transformer as the encoder and the MRG decoder as the decoder. In particular, we introduce an interaction-aware

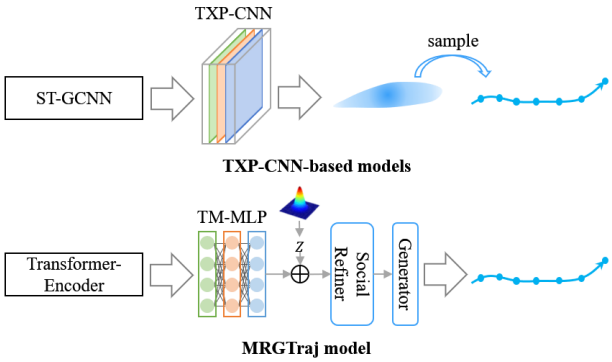


Fig. 2. Comparison illustration between TXP-CNN-based models and the proposed MRGTraj model. Both methods generate future trajectories using a non-autoregressive approach. TXP-CNN-based models typically generate a bivariate normal distribution and sample multiple future trajectories from it, while the MRGTraj model directly generates multiple future trajectories by repeatedly incorporating latent code during the decoding process.

latent code generator to learn a Gaussian distribution from the context of pedestrians' future social interaction. This allows us to sample latent codes multiple times from the learned distribution and concatenate them with the decoding features in the MRG decoder to generate multiple socially acceptable future trajectories. The main contributions of this paper are summarized as follows:

- We introduce a novel MRG decoder that utilizes the Mapping-Refinement-Generation structure to enable non-autoregressive decoding. To the best of our knowledge, this is the first application of the MRG decoder in the field of trajectory prediction.
- We propose an interaction-aware latent code generator that captures latent codes based on the context of future social interactions. As far as we know, this is the first approach to learn a latent distribution specifically from the context of future social interactions.
- We introduce the MRGTraj model, incorporating the MRG decoder and the interaction-aware latent code generator. The experimental results on two public datasets highlight the significant advancements achieved by the MRGTraj model, with a notable improvement of 13.21% on the Final Displacement Error (FDE) metric compared to state-of-the-art methods. Additionally, the MRGTraj model achieves a remarkable 71.29% speed-up.

The rest of the paper is organized as follows. Section II reviews related works on human trajectory prediction. Section III details the proposed MRG decoder. The proposed MRGTraj trajectory prediction model is described in Section IV. Section V presents the experimental results containing quantitative results, ablation studies, and qualitative results. Finally, the conclusions and future work are provided in Section VI.

II. Related Work

A. Autoregressive Methods for Trajectory Prediction

Recurrent Neural Networks (RNN), including variants such as Long Short-Term Memory (LSTM) [23] and Gated Recurrent Units (GRU) [24], have been successfully applied in

various sequence prediction tasks [25]–[27]. In the field of trajectory prediction, these models have been widely adopted to design high-performance models. For instance, the S-LSTM [28] model utilizes LSTM to model individual pedestrian, with the hidden state representing motion features and predicting future positions. It incorporates a social pooling module to capture the social tensor among neighboring pedestrians, which serves as an additional input to the LSTM at each time step. The Collision-free-LSTM [29] and Group LSTM [30] models introduce repulsion pooling and group-obstacle pooling, respectively, effectively integrating motion information from neighboring pedestrians to improve prediction performance. These models follow a step-by-step learning and reasoning approach, where the same LSTM is utilized for past and future time steps. In addition, LSTMs can be employed in a Seq2Seq structure, serving as both the encoder and decoder for trajectory prediction. Gupta et al. [31] propose the Social-GAN model, which uses a Seq2Seq structure in the generator and uses an LSTM as an encoder to extract encoded features from the past trajectory. The encoded feature will be used as the initial hidden state of the decoder LSTM. They also introduce a global pooling module to extract the social tensor and serve it as input of decoder LSTM to generate the future trajectory. Huang et al. [32] use two LSTMs as encoders to encode movement and interaction, respectively, and concatenate the two encoded features to initialize the hidden state of decoder LSTM for prediction. It is worth noting that all the above-mentioned methods rely on the output of the previous time step as the input for generating predictions step by step during the inference stage.

B. Non-autoregressive Methods for Trajectory Prediction

In recent years, there has been a growing interest in non-autoregressive structure-based trajectory prediction models, which enable parallel prediction and offer faster inference efficiency. For instance, Mohanmed et al. [18] treat pedestrians as nodes of a graph to create a spatio-temporal embedding through a Spatio-Temporal Graph Convolution Neural Network (ST-GCNN). They introduce a Time-Extrapolator Convolution Neural Network (TXP-CNN) that operates directly on the temporal dimension of the graph embedding and expands it to generate a binary Gaussian distribution of future trajectory. Zhou et al. [19] propose a variant model of Social-STGCNN by replacing Graph Convolution Networks (GCN) of STGCNN module with Graph Attention Network (GAT) for feature extraction on the spatial graph and effectively improved the prediction performance. Similarly, Shi et al. [21] present a Sparse Graph Convolution Network (SGCN), which explicitly models the sparse directed interaction with a sparse directed spatial graph to capture adaptive interaction among pedestrians, and uses a sparse directed temporal graph to model the motion tendency. Bae et al. [22] construct the multi-relational weighted graphs based on distances and relative displacements among pedestrians. They use a novel disentangled multi-scale aggregation to model social interactions among pedestrians on the weighted graph. In the prediction step, they also propose a global temporal aggregation to alleviate accumulated errors for pedestrians changing their directions. This is also the first of such models to improve the TXP-CNN module.

In this work, we propose a novel non-autoregressive decoder based on a Mapping-Refinement-Generation structure, which can directly generate the future trajectory without step-by-step decoding. This autoregressive decoder allows parallel operations across all future time steps, resulting in high efficiency for generating future trajectories.

C. Refinement in Trajectory Prediction

The refinement in the existing trajectory prediction models generally refines the hidden state of the current time step and then used for forecasting. For instance, Zhang et al. [33] use an LSTM with state refinement toward trajectory prediction (SR-LSTM). The refinement module collects effective information from the neighboring pedestrians through a message-passing mechanism to refine the hidden state of LSTM. The refined hidden state not only contains the movement information of pedestrians but also integrates the social interaction information among pedestrians. Thus, it can predict the future trajectory more effectively. They also propose an improved version [34] of their method, which introduces a spatial edge LSTM to handle relative spatial locations and incorporates a node-spatial-edge-based state refinement module. As another improvement, Su et al. [35] propose a collision-prior guided refinement, which can lead to understanding the collision situations of a crowd through a message-passing mechanism. Similarly, Bertugli et al. [36] propose an attentive hidden state refinement module that utilizes a GAT model adaptively calculate attention weight from hidden states among neighboring pedestrians and weighted aggregate the hidden states to refine its own hidden state. All the above methods use the state refinement module to refine the hidden state of LSTM in the autoregressive prediction model.

Inspired by the above methods, we introduce a social refiner in the MRG decoder to refine decoding features output by the temporal mapper. The social refiner operates concurrently across all future time steps and effectively incorporates the social interaction information among pedestrians into the refined decoding features. Thus, it the decoder to generate the socially acceptable future trajectory.

D. Latent Code Generation in Trajectory Prediction

Given a past trajectory, its future trajectory remains uncertain which depends on goals, self-decision-making, and the behavior of surrounding pedestrians. To address uncertainty, some GAN-based models [31], [37]–[39] commonly add Gaussian noise to the decoder multiple times to generate multiple socially acceptable future trajectories. To ensure the predicted future trajectory is as close to the ground-truth as possible, researchers introduce Conditional Variational Autoencoders (CVAEs) to learn a Gaussian distribution that approximates the distribution of ground-truth. For instance, Ivanovic et al. [40] adopt LSTMs as the encoder and decoder, and add an extra LSTM as the future trajectory encoder. In the training stage, a Gaussian distribution is learned from the combination of the past encoded feature and future encoded

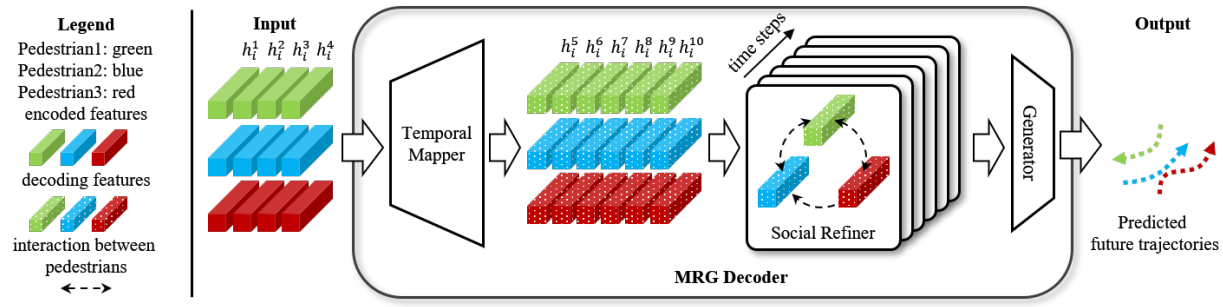


Fig. 3. Overview of the proposed MRG decoder. The MRG decoder includes three main modules: the temporal mapper, the social refiner, and the generator. The temporal mapper takes encoded features as input and produces decoding features for all future time steps. These decoding features are then refined by the social refiner, which models the interactions among pedestrians. Finally, the refined features are used by the generator to generate future trajectories. Innovatively, the introduction of the temporal mapper enables the model to make predictions in a non-autoregressive manner, enhancing its predictive capabilities.

feature, and then sample a latent code from it as an extra input to the decoder. In the testing stage, the latent code is sampled from the distribution learned only from the past encoded feature. It is also trained through the Kullback-Leibler divergence of the two distributions to ensure the similarity of the two distributions. Later works [14], [41], [42] also adopt such a way to learn the Gaussian distribution for latent code sampling to generate multiple future trajectories. Mangalam et al. [43] propose a two-stage method that predicts the endpoint of the future trajectory first and then generates the intermediate path. In this method, an endpoint CVAE is used to learn a Gaussian distribution from the endpoint feature to sample a latent code, and the latent code is sampled from the standard Gaussian distribution when testing.

Given the significant impact of social interaction on the uncertainty of pedestrian future trajectories, we propose to learn a Gaussian distribution specifically from the context of future social interactions among pedestrians. As a result, the model can sample latent codes from this distribution to guide the generation of multiple socially acceptable future trajectories.

III. MRG Non-Autoregressive Decoder

In trajectory prediction research, recurrent neural networks (RNNs), such as LSTMs, GRUs, and Transformer models, are commonly employed as encoders and decoders to capture the movement patterns of pedestrians. These methods typically generate future trajectories gradually in an autoregressive manner. In this section, we introduce a non-autoregressive decoder called MRG decoder, which is based on a Mapping-Refinement-Generation framework. Fig. 3 illustrates the components of the MRG decoder, including the temporal mapper, social refiner, and generator. We will provide detailed explanations of these components in the subsequent sections.

A. Temporal Mapper

Existing trajectory prediction methods generally use LSTM or Transformer as the decoder, which follows an autoregressive approach where the future trajectory is generated gradually by using the output of the previous time step as the input for the current time step. However, this sequential decoding process hampers parallel processing and limits efficiency. Moreover, errors in previous time steps can accumulate and impact subsequent predictions. To address these limitations, we propose a more direct approach to extract decoding features for all future time steps simultaneously from the encoded features. This non-autoregressive decoding method effectively mitigates the drawbacks associated with autoregressive decoders.

In the proposed MRG decoder, we introduce a temporal mapper module that processes the past encoded features to generate decoding features for all future time steps in a single step. Assume that N pedestrians in the scene and their past trajectories contain T_o time steps, thus the encoded feature tensor $H \in \mathbb{R}^{N \times T_o \times D}$ can be obtained by the encoder part. D is the dimension of the encoded feature. The temporal mapper starts with the encoded features tensors H, and the operation of the temporal mapper can be formulated as follows:

$$F = \Psi(H) \tag{1}$$

where $\Psi(\cdot)$ represents the operation function of the temporal mapper, and $F \in \mathbb{R}^{N \times T_p \times D}$ represents the decoding features of T_p future time steps of these N pedestrians. As shown in Fig. 3, the temporal mapper module allows for the simultaneous generation of decoding features for all pedestrians across all future time steps in a single step. This approach ensures that the decoding operations for each time step are independent of one another. By avoiding the step-wise decoding process used by autoregressive decoders, the temporal mapper module helps mitigate the potential for error accumulation.

B. Social Refiner

The temporal mapper module extends the encoded feature in the time-space domain to obtain decoding features for all pedestrians across all future time steps. To further enhance the effectiveness of these decoding features, we design a social refiner module. The purpose of the social refiner is to refine the decoding features by modeling the interactions among pedestrians and integrating latent social interaction information into the refined decoding features. Formally, assume that the decoding feature of pedestrian i is denoted as $F_i \in \mathbb{R}^{T_p \times D}$. The operation of the social refiner is formulated by:

$$\mathcal{F} = \Phi(F_1, F_2, \cdots, F_N) \tag{2}$$

where $\Phi(\cdot)$ represents the operation function of the social refiner, $\mathcal{F} \in \mathbb{R}^{N \times T_p \times D}$ represents the refined decoding features.

In detail, the social refiner models the different influences among pedestrians based on the decoding features of all pedestrians. It then facilitates the exchange and collection of information among pedestrians by leveraging these distinct influences. The collected information is subsequently utilized to update the decoding features of each pedestrian, thereby achieving the objective of refinement. By incorporating social interaction information through these operations, the refined decoding features enable the generation of future trajectories that are more socially plausible and acceptable.

C. Generator

Unlike the TXP-CNN-based models that produce a binary Gaussian distribution, the MRG decoder directly generates a specific future trajectory. By obtaining the decoding features for all future time steps, it can directly generate the complete future trajectory using these features. This non-autoregressive approach is advantageous as it helps mitigate the accumulation of errors that often occur in step-by-step forecasting. Furthermore, the MRG decoder's parallel processing capability allows for the handling of all time steps simultaneously. This means that the positions for all future time steps can be generated synchronously, enhancing efficiency and coherence in the trajectory prediction process. By avoiding sequential generation, the MRG decoder enables faster and more synchronized trajectory predictions for multiple time steps.

IV. MRGTraj Model for Trajectory Prediction

The objective of pedestrian trajectory prediction is to forecast the future trajectory based on the given past trajectory. Following the standard formulation of trajectory forecasting problem in the literature [31], [44], we assume that N pedestrians are involved in a scenario, and the model aims to predict the future trajectories for T_p time steps based on the past T_o time steps' trajectories. For each pedestrian i, its past trajectory is denoted as $X_i = (X_i^1, X_i^2, ..., X_i^{T_o})$, and its future trajectory is denoted as $Y_i = (Y_i^1, Y_i^2, ..., Y_i^{T_p})$. The predicted future trajectory is denoted as $\hat{Y}_i = (\hat{Y}_i^1, \hat{Y}_i^2, ..., \hat{Y}_i^{T_p})$.

A. Overview

In this section, we introduce MRGTraj, a novel human trajectory prediction model that leverages the MRG decoder. As illustrated in Fig. 4, the MRGTraj model comprises three key components: an encoder, an interaction-aware latent code generator, and an MRG decoder. To begin, the encoder utilizes Transformers to extract encoded features from the past trajectories of pedestrians. This encoding process captures essential information from the historical trajectories, serving as the foundation for accurate future trajectory prediction. Subsequently, the interaction-aware latent code generator produces latent codes based on the social interaction context. These latent codes encompasse the collective social dynamics within the scene, contributing to more informed trajectory predictions. Finally, the MRG decoder module takes both the latent code and encoded features as input to generate multiple plausible future trajectories. By considering the latent code and encoded features simultaneously, the MRG decoder effectively models the dependencies among pedestrians and generates socially plausible future trajectories. In the following sections, we will provide a comprehensive explanation of each module, outlining their specific functionalities and contributions to the MRGTraj model.

B. Encoder

In classical trajectory prediction models, LSTM models have been widely used as encoders in many trajectory prediction models [32], [45], [46] to extract motion features from past pedestrian trajectories. However, in recent years, the Transformer network [47] and its variants [48]–[51] have shown remarkable performance in various natural language processing and computer vision tasks. Given the successful application of the Transformer network in sequence modeling, we choose to employ it as the encoder in our MRGTraj model for motion feature extraction.

In this paper, we adopt the encoder component of the Transformer network to serve as the encoder in the MRGTraj model. The encoder consists of several essential layers: an input embedding layer, a position encoding layer, a multi-head self-attention layer, a feed-forward layer, and two residual connection layers. The input embedding layer plays a vital role in converting the input trajectory data into a fixed-length vector representation. The position encoding layer introduces positional information to capture the temporal order of the trajectory data. This positional encoding helps the model understand the sequence of pedestrian movements. The multi-head self-attention layer is a key component that applies the self-attention mechanism to capture dependencies between different time steps of the input trajectories. By attending to relevant parts of the input sequence, the model effectively captures the temporal dependencies and maintains the consistency of the pedestrian's movement patterns. The feed-forward layer applies nonlinear transformations to the attended features, further enhancing the model's ability to extract meaningful representations. Lastly, the residual connection layers enable the direct flow of information from earlier layers to subsequent layers, facilitating gradient flow and improving the overall training process. By incorporating these layers and the self-attention mechanism, the encoder effectively models the temporal consistency of pedestrian movement, allowing the MRGTraj model to capture intricate patterns and dependencies in the input trajectories. Formally, the encoder operation is represented by:

$$H_i^X = \text{Transformer}(X_i) \tag{3}$$

where $H_i^X = (h_i^1, h_i^2, ..., h_i^{T_o})$ is the encoded feature sequence of pedestrian i. The encoded features of the N pedestrians can be denoted as a sequence $H^X = (H_1^X, H_2^X, ..., H_N^X)$ of length $N \times T_o$.

C. Interaction-aware Latent Code Generator

In the context of predicting future pedestrian trajectories, the inherent randomness in individual movement patterns

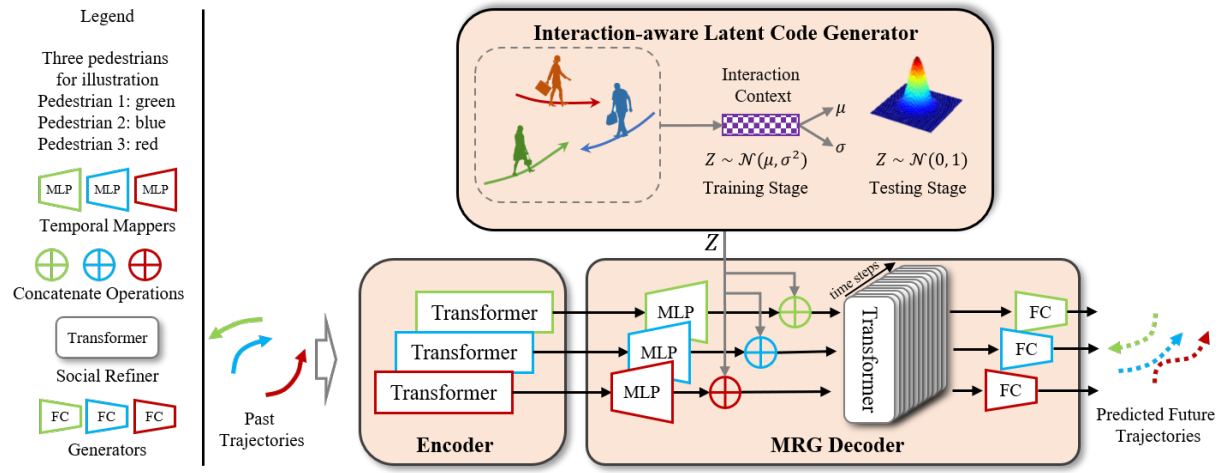


Fig. 4. Overview of the proposed MRGTraj model. The MRGTraj model consists of three main modules: encoder, interaction-aware latent code generator, and the MRG decoder.

generally leads to the existence of multiple socially acceptable future trajectories. Existing models address this by incorporating either Gaussian noise or latent codes to guide the decoder in generating multiple plausible future trajectories [14], [37], [39], [43], [52], [53]. However, these approaches often overlook the crucial role of social interactions in the presence of multimodal trajectories. To address this limitation, we propose an interaction-aware latent code generator using a CVAE structure. This novel generator takes into account the social interaction context among pedestrians and learns a Gaussian distribution. By sampling latent codes from this learned distribution, the model is guided to generate multiple socially acceptable trajectories.

The interaction-aware latent code generator mainly includes two steps: interaction context extraction and latent code generation. The process is illustrated in Fig. 5. In the first step, the model calculate the social interaction context among pedestrians in the scene at a specific time step t. To do this, it first obtain an embedding vector for each pedestrian by processing their location and velocity information through a Multilayer Perceptron (MLP). These embedding vectors capture individual pedestrian characteristics. Then, the model aggregate the embedding vectors of all pedestrians present in the scene to obtain a global social interaction context denoted as C^t. In the second step, the model compute the future social interaction context by considering the global social interaction contexts across future time steps: $C = \text{mean}(C^{T_o+1}, C^{T_o+2}, \cdots, C^{T_o+T_p})$. This aggregation step allows the model to capture the collective influence of social interactions on future trajectories. Next, the model uses another MLP to map the aggregated social interaction context C to the parameters (μ, δ) of a Gaussian distribution $\phi(\mathrm{z}|Y_s) = \mathcal{N}(\mu, \delta)$. The latent code z is sampled from this distribution and treated as an additional input to the MRG decoder, enabling it to generate multimodal future trajectories. During the inference stage, where the future trajectory is unknown, the model samples the latent code z from a standard normal distribution $\mathcal{N}(0,1)$ to capture the uncertainty and diversity of future predictions. By incorporating the interaction-aware latent code generator, the MRGTraj model considers the social context and generates multiple socially acceptable trajectories, providing a more comprehensive and realistic prediction of future pedestrian movement.

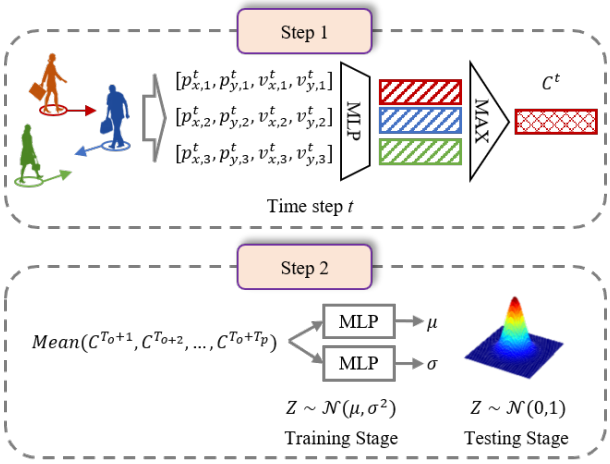


Fig. 5. The illustration of the interaction-aware latent code generator. The generator consists of two steps: interaction context extraction and latent code generation.

D. MRG decoder

1) Temporal Mapper: In the MRGTraj model, we adopt an MLP with three hidden layers as the temporal mapper. The input of the temporal mapper MLP is a feature sequence $H = (H_1, H_2, ..., H_N)$ of length $N \times T_o$, which is output from the encoder. These features are processed through multiple layers of neurons, allowing the model to capture complex dependencies between the input and output features. The temporal mapper MLP leverages this deep architecture to effectively model the relationships between each input feature and its corresponding output feature. The output of the temporal mapper is a decoding feature sequence $F = (F_1, F_2, ..., F_N)$ of length $N \times T_p$, which serves as the basis for generating

future trajectories. For each pedestrian i, their decoding feature sequence is denoted as $F_i = (f_i^1, f_i^2, ..., f_i^{T_p})$, representing the decoding features for the T_p future time steps. One key advantage of the temporal mapper is its ability to capture the decoding features for all future time steps in a single operation. This non-autoregressive approach allows the model to generate future trajectories more efficiently, as it does not rely on the sequential generation of each time step. Instead, the temporal mapper simultaneously predicts the decoding features for all future time steps, enabling a more parallelized and streamlined trajectory generation process.

2) Social Refiner: We adopt a Transformer model as the social refiner in the MRGTraj model. Different from the Transformer model as an encoder, the Transformer model as the social refiner applies the attention mechanisms across pedestrians. This enables it to capture and leverage the social interactions among pedestrians for feature refinement. For multimodal trajectory prediction, the social refiner takes the concatenation of the encoded features and the latent code as input and produces refined decoding features. These decoding features not only integrate the social interaction information of pedestrians but also the multimodal information of the future trajectory. To illustrate the process, at each time step t, the input to the social refiner is a sequence denoted as $\mathbb{F}^t = \{\mathrm{f}_1^t, \mathrm{f}_2^t, ..., \mathrm{f}_N^t\}$, where $\mathrm{f}_i^t = f_i^t + \mathrm{z}$. The social refiner then generates a refined feature sequence denoted as $\hat{\mathbb{F}}^t = \hat{\mathrm{f}}_1^t, \hat{\mathrm{f}}_2^t, ..., \hat{\mathrm{f}}_N^t$. Through this refinement process, the social interaction information is effectively incorporated into the decoding features, enhancing the model's ability to infer future trajectories that adhere to social norms and interactions among pedestrians.

3) Generator: To generate the future trajectory for each pedestrian, we utilize a fully connected layer as the generator in MRG decoder. This generator takes the refined decoding features as input and maps them to the future locations. The predicted future trajectory for pedestrian i is denoted as $\hat{Y}_i = \hat{Y}_i^1, \hat{Y}_i^2, ..., \hat{Y}_i^{T_p}$, where $\hat{Y}_i^t$ represents the predicted location of pedestrian i at time step t in the future. By applying the generator to the refined decoding features, we obtain the anticipated trajectory for each pedestrian, enabling us to forecast their future movements accurately.

E. Loss Function

We train the entire network end to end by using the following loss functions:

$$\begin{aligned} Loss &= L_{mse} + L_{kl} \\ L_{mse} &= \frac{1}{N}\sum_{i=1}^{N} \parallel Y_i - \hat{Y}_i \parallel_2 \\ L_{kl} &= \mathrm{KL}(\phi(z|Y_s) \parallel \mathcal{N}(0, I)) \end{aligned} \tag{4}$$

where the first term encourages the predicted future trajectory $\hat{Y}_i$ to approach the ground-truth Y_i. The Kullback-Leibler divergence $\mathrm{KL}(\cdot)$ acts as a regularizer and encourages the prior distribution of the latent code z close to standard Gaussian distribution. The Y_S represents the future social interaction context among pedestrians. At test time, we sample latent codes $\{z^{(1)}, z^{(2)}, ..., z^{(K)}\}$ from $\mathcal{N}(0, 1)$ instead of sampling from the learned distribution and decode the decoding features into trajectories $\{\hat{Y}_i^1, \hat{Y}_i^2, ..., \hat{Y}_i^K\}$.

V. Experiments

To evaluate the effectiveness of the proposed model, we perform qualitative and quantitative experiments. In this section, we provide the implementation details of the model and give a brief description of the datasets and evaluation metrics. Then, we list the state-of-the-art baseline methods and demonstrate the qualitative and quantitative comparative results.

A. Implementation and Evaluation Details

We implement the proposed model on top of the Pytorch. The dimensions of embedding vectors of past and future trajectory encoders are set to 256. The dimension of key, query, and value vectors in self-attention is set to 64. All MLPs have only 3 middle layers and the neurons number are 512, 1024, and 512 respectively. The dimension of the latent code is set to 64. The dimensions of feature vectors in the social refiner Transformer are set to 320. We train the model by the SGD optimizer with an initial learning rate of 0.0001 for 300 epochs, and the batch size is around 256 pedestrians.

The model is trained and tested in ETH [54] and UCY [55] datasets, which consist of five scenarios (ETH, HOTEL, UNIV, ZARA1, and ZARA2). Following previous work [18], [31], [32], we consider the previous 8 keyframes (3.2s) positions as the past trajectory and the subsequent 12 keyframes (4.8s) positions as the future trajectory. The evaluation of all models is conducted using the Average Displacement Error (ADE) and Final Displacement Error (FDE) metrics. The ADE measures the average L2 distance between the predicted trajectories and the ground-truth trajectories, while the FDE measures the L2 distance between the predicted final destination and the ground-truth final destination after the prediction horizon of T_p.

B. Baseline Methods

We compare the proposed MRGTraj model with the following state-of-the-art models:

- *S-LSTM* [28]: A classical LSTM-based method that introduces a social pooling mechanism for social interaction modeling.
- *S-GAN* [31]: A classical GAN-based method that introduces a global pooling module for social interaction modeling.
- *SCAN* [56]: An improved model of S-GAN which adopts a spatial context attentive network for social interaction modeling.
- *RAMP* [44]: A GAN-base model which develops a reciprocal network learning approach for human trajectory prediction.
- *SSALVM* [53]: A RNN-base model to achieve prediction via latent variational model by coupling modeling social interaction and human-scene interaction.

- *STGAT* [32]: An LSTM-based Seq2Seq model which uses LSTM as encoder and decoder and models social interaction by GATs.
- *NMMP* [46]: An improved version of S-GAN with a novel neural motion message passing for social interaction modeling.
- *IGGAN* [57]: A GAN-based model to jointly model the self intentions and social interactions by a novel intention-interaction graph.
- *STIRNet* [58]: An LSTM-based model which uses GAT for social interaction modeling.
- *MG-GAN* [59]: A GAN-based model for multimodal trajectory prediction which prevents out-of-distribution samples through the multi-generator model.
- *E-SR-LSTM* [34]: An LSTM-based model which performs state refinement for node and spatial edge for social-aware trajectory prediction.
- *SRAI-LSTM* [52]: An LSTM-based model which introduces a novel social relation encoder and model social interaction by social relation attention.
- *STAR* [60]: A Transformer-based model which consists of spatial Transformers and temporal Transformers while performing prediction in an autoregressive manner.

As the MRGTraj model achieves trajectory prediction in a novel non-autoregressive manner, we also compare it with the following non-autoregressive baseline methods:

- *S-STGCNN* [18]: A classical CNN-based model which introduces a time extrapolator CNN for non-autoregressive prediction.
- *AST-GNN* [19]: An improved version of S-STGCNN which develops the spatial-temporal modeling by stacking spatial-GNN and temporal-GNN.
- *Att-GCNN* [20]: An improved version of S-STGCNN which utilizes the near pedestrian attention function to compute the weight adjacency matrix to improve prediction accuracy.
- *SGCN* [21]: An improved version of S-STGCNN which performs spatial-temporal graph convolution on sparse directed spatial graph and temporal graph.
- *DMRGCN* [22]: An improved version of S-STGCNN which performs spatial-temporal graph convolution through disentangled multiscale aggregation and multi-relational graph aggregation.

C. Quantitative Evaluation

1) Stability Analysis: To validate the effectiveness and stability of MRGTraj, we conduct 100 repeated testing on the trained model across five different scenarios, respectively. As shown in Fig. 6, the mean and standard deviation of ADEs and FDEs from 100 testing experiments were calculated, which are presented using histograms and error bars, respectively. It is important to note that during each test, 20 predicted trajectories are generated, and the trajectory that is closest to the ground-truth is selected to calculate the ADE and FDE. The ADE and FDE values presented in the subsequent tables and figures are also calculated using this approach. After 100 repeated testing experiments, the average test results for

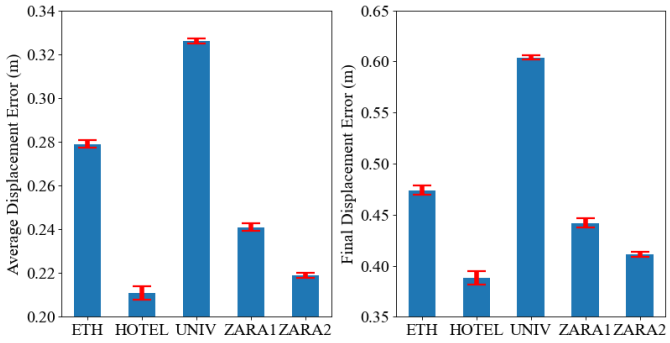


Fig. 6. Experiment results of reliability validation. The left figure displays the statistical results on the ADE metric from 100 repeated testing experiments, while the right figure shows the statistical results on the FDE metric. The histograms represent the average values of the 100 tests, and the error bars indicate the standard deviation of the test results.

the five scenarios are as follows: ETH: 0.28/0.47, HOTEL: 0.21/0.39, UNIV: 0.33/0.60, ZARA1: 0.24/0.44, ZARA2: 0.22/0.41. These results indicate that the prediction performance of MRGTraj is satisfactory. Furthermore, the standard deviations of the test results are also within an acceptable range. For example, for the ADE metric, the standard deviation is highest in the HOTEL scenarios, approximately 0.0031, while it is lowest in the UNIV scenario, approximately 0.0010. Similarly, for the FDE metric, the standard deviation is highest in the HOTEL scenario, approximately 0.0067, while it is lowest in the UNIV scenario, approximately 0.0024. These results indicate that the prediction capability of the MRGTraj model is sufficiently stable, and the generated predictions are reliable and satisfactory.

2) Comparison with Existing Work: We compare our MRGTraj model with the state-of-the-art baselines mentioned above, and the comparisons are listed in Table I, and all values are rounded to two decimal places. The ADEs and FDEs of all approaches are the best predictions among 20 prediction samples except the S-LSTM model. The MG-GAN [59] outperforms the other models on the HOTEL dataset. The STAR [60] model outperforms the other methods on the UNIV dataset. The proposed MRGTraj model outperforms the other methods on the remaining three datasets, which achieve the minimum average ADE and FDE on five datasets. In autoregressive-based baselines, the SRAI-LSTM [52] and STAR [60] models outperform the other baselines, which have an average ADE/FDE of 0.26/0.53 on the five datasets. Compared to these two models, the average ADE of MRGTraj is 0.26, which is equivalent to theirs. However, MRGTraj achieves an average FDE value of 0.46, representing an improvement of approximately 13.21%. On the other hand, the DMRGCN [22] model outperforms the other non-autoregressive methods with the average ADE and FDE of 0.34 and 0.58. Compared to the DMRGCN model, MRGTraj achieves the average ADE and FDE with improvements of about 26.47% and 20.69%, respectively.

3) Comparisons on Inference Speed: To evaluate the inference speed, we list out comparisons of parameter size and inference time between our model and publicly available models which we could benchmark against. We treat each

TABLE I
COMPARISONS OF THE MRGTRAJ MODEL WITH BASELINE APPROACHES ON ETH AND UCY DATASETS FOR ADE AND FDE METRICS. NA DENOTES THAT THE MODEL ACHIEVES PREDICTION IN A NON-AUTOREGRESSIVE MANNER.

Model	Year	NA	ETH		HOTEL		UNIV		ZARA1		ZARA2		AVG	
			ADE	FDE	ADE	FDE	ADE	FDE	ADE	FDE	ADE	FDE	ADE	FDE
S-LSTM [28]	2016	–	1.09	2.35	0.79	1.76	0.67	1.40	0.47	1.00	0.56	1.17	0.72	1.54
S-GAN [31]	2018	–	0.81	1.52	0.72	1.61	0.60	1.26	0.34	0.69	0.42	0.84	0.58	1.18
SCAN [56]	2021	–	0.79	1.79	0.37	0.74	0.58	1.23	0.37	0.78	0.31	0.66	0.48	0.98
RAMP [44]	2022	–	0.69	1.24	0.43	0.87	0.53	1.17	0.28	0.61	0.28	0.59	0.44	0.90
SSALVM [53]	2021	–	0.61	1.09	0.28	0.51	0.59	1.24	0.37	0.78	0.30	0.64	0.43	0.85
STGAT [32]	2019	–	0.65	1.12	0.35	0.66	0.52	1.10	0.34	0.69	0.29	0.60	0.43	0.83
NMMP [46]	2020	–	0.61	1.08	0.33	0.63	0.52	1.11	0.32	0.66	0.29	0.61	0.41	0.82
IGGAN [57]	2022	–	0.66	1.11	0.24	0.38	0.47	1.00	0.34	0.69	0.27	0.57	0.40	0.75
STIRNet [58]	2021	–	0.48	0.95	0.22	0.41	0.54	1.15	0.37	0.80	0.31	0.70	0.38	0.80
MG-GAN [59]	2021	–	0.47	0.91	**0.14**	**0.24**	0.54	1.07	0.36	0.73	0.19	0.60	0.36	0.71
E-SR-LSTM [34]	2022	–	0.44	0.79	0.19	0.31	0.50	1.05	0.32	0.64	0.27	0.54	0.34	0.67
SRAI-LSTM [52]	2022	–	0.32	0.59	0.18	0.34	0.35	0.72	**0.24**	0.51	0.23	0.50	**0.26**	0.53
STAR [60]	2020	–	0.36	0.65	0.17	0.36	**0.31**	0.62	0.26	0.55	**0.22**	0.46	**0.26**	0.53
S-STGCNN [18]	2020	✓	0.64	1.11	0.49	0.85	0.44	0.79	0.30	0.48	0.34	0.53	0.44	0.75
AST-GNN [19]	2021	✓	0.69	1.27	0.36	0.62	0.46	0.83	0.32	0.53	0.28	0.44	0.40	0.66
Att-GCNN [20]	2021	✓	0.68	1.22	0.31	0.41	0.39	0.69	0.34	0.65	0.28	0.44	0.40	0.66
SGCN [21]	2021	✓	0.63	1.03	0.32	0.55	0.37	0.70	0.29	0.53	0.25	0.45	0.37	0.65
DMRGCN [22]	2021	✓	0.60	1.09	0.21	0.30	0.35	0.63	0.29	0.47	0.25	**0.41**	0.34	0.58
Ours	2022	✓	**0.28**	**0.47**	0.21	0.39	0.33	**0.60**	**0.24**	**0.44**	**0.22**	**0.41**	**0.26**	**0.46**

TABLE II
COMPARISONS OF PARAMETER AMOUNT AND INFERENCE SPEED ON ETH & UCY DATASETS. ALL MODELS EVALUATED ON NVIDIA GTX2080TI GPU.

Model	NA	Parameters(k)	Speed(ms)
S-GAN [31]	–	46.4	11.24 (2.34x)
STGAT [32]	–	44.6	9.81 (2.04x)
NMMP [46]	–	115.8	15.26 (3.18x)
STAR [60]	–	964.9	27.12 (5.65x)
SRAI-LSTM [52]	–	67.1	19.00 (3.96x)
S-STGCNN [18]	✓	7.6	7.50 (1.56x)
SGCN [21]	✓	25.4	7.74 (1.61x)
MRGTraj	✓	4,358.9	4.80

fragment with 20 time steps as a sample and calculate the inference time of a single future trajectory generation. The inference speeds listed in Table II are the average inference time of all samples among five datasets. Different from the autoregressive-based models S-GAN[1] [31], STGAT[2] [32], NMMP[3] [46] and SRAI-LSTM [52], our model obtains decoding features through a temporal mapper and generate future trajectory in a parallel manner. Although the parameter amount of the MRGTraj model is about thousands of times that of these models, the inference speed is the fastest. The inference speed of these autoregressive models is commonly more than two times slower than the MRGTraj model. Furthermore, compared with non-autoregressive-based models S-STGCNN[4] [18] and SGCN[5] [21], the inference speed of MRGTraj still exceeds them by about 36% and 37.98%. Although MRGTraj has a large number of parameters, its encoder and decoder are processed in parallel across different time steps, so it has a faster reasoning ability. In brief, MRGTraj not only outperforms in prediction but also achieves superior inference speed.

[1]https://github.com/agrimgupta92/sgan
[2]https://github.com/huang-xx/STGAT
[3]https://github.com/PhyllisH/NMMP
[4]https://github.com/abduallahmohamed/Social-STGCNN
[5]https://github.com/shaishiliu/SGCN

D. Ablation Study

1) Contribution of MRG decoder: MRG decoder is a novel non-autoregressive decoder that can decode in parallel at all future time steps. Thus, it has a faster inference speed than those autoregressive decoders. We verify these two advantages of the MRG decoder through multiple variant models with different encoders and decoders, and the results are listed in Table III. We use RNN, GRU, LSTM, and TF as encoders and decoders, respectively, and compare them with the MRG decoder as its decoder. For fair comparison, all autoregressive decoders are equipped with TF-based social refiner to refine the decoding features. Additionally, the feature dimensions for all variant models are set to 256 to highlight the differences brought by the decoding methods as much as possible. The comparison results show that for the same encoder, the average ADE and FDE of the MRG decoder are commonly lower than that of autoregressive decoders (that of the two models with LSTM as the encoder is the same). Our proposed MRGTraj model (TF as encoder and MRG decoder as decoder) achieves the best prediction performance. During the inference process, the MRG decoder obtains decoding features for all future time steps through a single operation of the temporal mapper. Subsequent refinement and generation operations can be parallelized across all time steps. In contrast, autoregressive decoding requires more computation time as it performs step-by-step inference for all future time steps. As shown in Table III, variant models using the MRG decoder, despite having more parameters, generally exhibit faster inference speed. The

TABLE III
ABLATION EXPERIMENTS ON ENCODER AND DECODER OF MRGTRAJ MODEL. ALL AUTOREGRESSIVE DECODERS ARE EQUIPPED WITH TF-BASED SOCIAL REFINERS FOR FAIR COMPARISON. ALL ADE AND FDE VALUES ARE CALCULATED FROM THE BEST ONE OF THE 20 PREDICTIONS.

Encoder	Decoder	Parameters (k)	ETH	HOTEL	UNIV	ZARA1	ZARA2	AVG	Speed (ms/batch)
RNN	RNN + (TF)Refiner	**2,709.3**	0.34/0.64	0.29/0.62	0.47/0.97	**0.26**/0.48	**0.23/0.41**	0.32/0.63	17.75
RNN	MRG decoder	3,670.9	**0.30/0.54**	**0.25/0.46**	**0.36/0.67**	**0.26/0.47**	0.24/0.43	**0.28/0.51**	**4.12**
GRU	GRU + (TF)Refiner	**3,104.5**	**0.32/0.58**	**0.26**/0.56	0.39/0.75	**0.25/0.45**	**0.23/0.40**	**0.29**/0.55	17.66
GRU	MRG decoder	3,868.6	**0.32**/0.59	0.27/**0.50**	**0.36/0.64**	0.27/0.48	0.24/0.43	**0.29/0.53**	**4.04**
LSTM	LSTM + (TF)Refiner	**3,302.1**	**0.33/0.60**	**0.25/0.51**	0.41/0.80	**0.26/0.48**	**0.23/0.42**	**0.30/0.56**	17.66
LSTM	MRG decoder	3,967.4	**0.33/0.60**	0.31/0.57	**0.37/0.67**	0.27/0.50	0.24/0.44	**0.30/0.56**	**4.03**
TF	RNN + (TF)Refiner	**3,397.3**	0.43/0.82	0.27/0.54	0.40/0.86	0.29/0.59	0.24/0.50	0.32/0.66	18.35
TF	GRU + (TF)Refiner	3,594.9	0.37/0.62	0.29/0.57	0.37/0.74	0.25/0.49	0.22/0.42	0.30/0.57	18.53
TF	LSTM + (TF)Refiner	3,693.7	0.36/0.62	0.26/0.54	**0.32**/0.63	0.25/0.48	**0.21**/0.41	0.28/0.53	18.41
TF	TF + (TF)Refiner	4,085.5	0.34/0.66	0.30/0.56	0.43/0.73	0.27/0.54	0.24/0.46	0.32/0.59	38.53
TF	MRG decoder	4,359.9	**0.28/0.47**	**0.21/0.39**	0.33/**0.60**	**0.24/0.44**	0.22/**0.41**	**0.26/0.46**	**4.82**

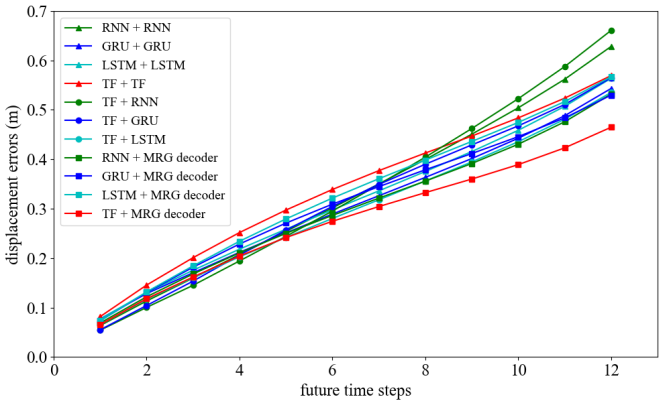


Fig. 7. Comparison of the displacement errors at each future time step of the MRGTraj model and its variants. These displacement error curves show the effectiveness of the MRG decoder in alleviating error accumulation.

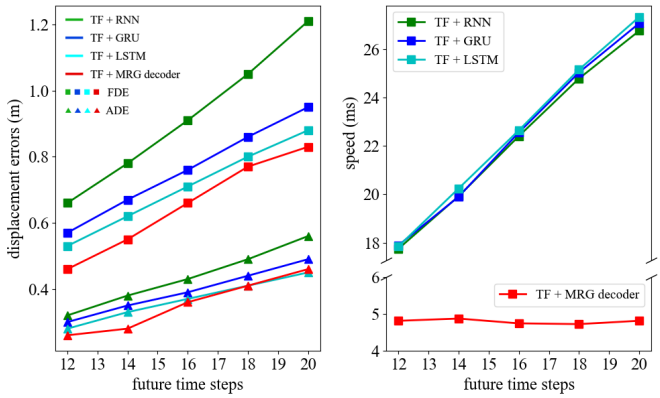


Fig. 8. Comparison of the displacement errors at each future time step of the MRGTraj model and its variants. These displacement error curves show the effectiveness of the MRG decoder in alleviating error accumulation.

inference speed of the MRG decoder-based model is more than four times faster than that of the model based on the autoregressive decoder. These experimental results completely prove that the MRG decoder has excellent prediction ability and efficiency.

The MRG decoder operates with the inference at all future time steps being independent of each other, in contrast to the autoregressive decoder that performs step-by-step inference. As a result, the MRG decoder effectively mitigates the problem of error accumulation that arises from the step-wise inference of autoregressive decoders. To support this claim, we conduct a statistical analysis of the displacement error at each time step for each variant model mentioned in Table III. The results are presented in Fig. 7, which clearly demonstrates the effectiveness of the MRG decoder. While some models may exhibit smaller displacement errors than MRGTraj (TF + MRG decoder) in the first five time steps, MRGTraj consistently achieves the lowest displacement errors for all subsequent time steps. Furthermore, upon careful observation, it can be noted that, for the same encoder, the displacement error of the MRG decoder in the later time steps is generally smaller compared to that of autoregressive decoders. These findings serve as evidence of the MRG decoder's effectiveness in reducing error accumulation.

To further validate the capability of the MRG decoder, we conduct validation experiments on longer prediction time steps, specifically at 12, 14, 16, 18, and 20 future time steps for comparison. The results are presented in Fig. 8, which includes the average ADE, FDE, and inference speed across five scenarios for four variant models that utilize TF as the encoder with RNN, GRU, and LSTM as decoders, as well as the proposed MRGTraj model (TF + MRG decoder). To ensure fairness, all variant models incorporate a TF-based social refiner module. For careful observation, it can be noticed that the MRGTraj model generally achieves lower ADE and FDE values on predictions with longer time steps compared to other variant models. Furthermore, even though the MRGTraj model have a higher ADE value compared to the model using LSTM as the decoder on predictions with a time step of 20, its FDE value still remains the lowest. These results provide evidence that the MRG decoder remains competitive in trajectory prediction on longer time steps compared to autoregressive decoders. From the results in the right figure, it can be observed that as the prediction time steps increase, the inference speed of the autoregressive decoders also exhibits close to linear growth. In contrast, the MRG decoder's inference speed shows little to no significant increase trend. From the results in the right figure, it can be observed that as the prediction time steps

TABLE IV
RESULTS OF THE ABLATION STUDY ON LATENT CODE GENERATOR. THE LATENT CODE GENERATION METHODS IN VARIANTS 3 AND 4 ARE ALL PROPOSED IN THIS WORK.

Variant ID	1	2	3 (Ours)	4 (Ours)
Training	$\phi(z\|X,Y)$	$\phi(z\|Y_{end})$	$\phi(z\|Y)$	$\phi(z\|Y_s)$
Testing	$\phi(z\|X)$	$N(0,1)$	$N(0,1)$	$N(0,1)$
ETH	0.301/0.532	0.288/0.495	0.294/0.507	**0.279/0.474**
HOTEL	0.210/0.391	0.224/0.411	0.216/0.402	**0.211/0.388**
UNIV	0.398/0.769	0.335/0.620	0.332/0.614	**0.326/0.604**
ZARA1	0.259/0.482	0.243/0.444	**0.241/0.440**	**0.241**/0.442
ZARA2	0.232/0.457	0.221/0.414	0.222/**0.411**	**0.219/0.411**
AVG	0.280/0.526	0.262/0.477	0.261/0.475	**0.255/0.464**

TABLE V
RESULTS OF THE ABLATION STUDY ON SOCIAL REFINER.

	Social Refiner Modules		
Dataset	×	GAT	Transformer
ETH	0.33/0.56	0.29/0.49	**0.28/0.48**
HOTEL	0.24/0.42	0.27/0.47	**0.20/0.37**
UNIV	0.40/0.71	0.37/0.65	**0.33/0.62**
ZARA1	0.26/0.48	0.28/0.51	**0.24/0.44**
ZARA2	0.24/0.44	0.25/0.46	**0.22/0.41**
AVG	0.29/0.52	0.29/0.52	**0.25/0.46**

increase, the inference speed of the autoregressive decoders also exhibits close to linear growth. In contrast, the MRG decoder's inference speed shows little to no significant increase trend. For longer prediction time steps, the MRG decoder maintains a faster inference speed, which fully demonstrates the advantage of the MRG decoder in prediction efficiency.

2) Contribution of Latent Code Generator: To verify the validity of the proposed social interaction-aware latent code generator, we will compare it with some latent code generation methods that have emerged from existing work. Specifically, the method of variant 1 appears in [14], [40] and uses $KL(\phi(z|X,Y),\phi(z|X))$ to train, the method of variant 2 appears in [43], [61] and uses $KL(\phi(z|Y_{end}),N(0,1))$ to train. We propose the methods of variants 3 and 4 and use $KL(\phi(z|Y),N(0,1))$ and $KL(\phi(z|Y_s),N(0,1))$ to train, respectively. All the results of this variant model are listed in Tabel IV. To better observe the differences in prediction performance among the variant models, all ADE and FDE values are rounded to three decimal places for statistical analysis. The proposed MRGTraj model can always achieve satisfactory prediction results regardless of the latent code generator used. Moreover, the prediction performance of variant models (3, 4) using our contributed latent code generation methods are superior to the other two variants. Particularly, the model based on the social interaction-aware latent code generator achieves the best prediction performance on almost all data sets.

3) Contribution of Social Refiner: The social refiner module plays a crucial role in incorporating social interaction information among pedestrians into the decoding features, thereby generating more plausible future trajectories. To evaluate the effectiveness of this module, we conduct experiments comparing the model without the social refiner and the model with the GAT as the social refiner. The results in Table V demonstrate that the Transformer-based social refiner outperforms other methods in terms of prediction performance. Surprisingly, the model with the GAT-based social refiner performs worse than the model without any social refiner. This highlights the importance of selecting an appropriate model as the social refiner, as it significantly impacts the prediction performance of the overall model.

E. Qualitative Evaluation

In this section, we present visualization results to demonstrate the prediction performance of the MRGTraj model compared to baseline methods. Specifically, we compare the performance of MRGTraj with the STGAT [32] and SRAI-LSTM [52] models on four scenarios: ETH, HOTEL, UNIV, and ZARA1 datasets. The visualization results depict the best predicted future trajectories among the 20 generated trajectories for each model. In Fig. 9, we show the comparison results for the ZARA1 dataset in the first column. The future trajectory predicted by the STGAT model exhibits significant deviations from the ground-truth, while the trajectory predicted by the SRAI-LSTM model has deviations in the middle trajectories, although the endpoints match. In contrast, the predictions of the MRGTraj model align perfectly with the ground-truth. For the ETH scenario in the second column, the trajectories predicted by the STGAT and SRAI-LSTM models are somewhat mismatched with the ground-truth. In contrast, the trajectories predicted by the MRGTraj model are closer to the ground-truth.

We also present visualization results to highlight the differences in the future trajectories generated by the decoding features before and after the social refiner. In the first scenario from the ETH dataset (shown in the left figure of Fig. 10), it can be observed that the future trajectory of the left pedestrian (P1) is more aligned with the ground-truth after the refinement process, indicating an improvement in prediction accuracy. However, for the right pedestrian, the differences between the future trajectories before and after refinement are not significant. Similarly, in the second scenario from the ZARA2 dataset (right figure), we can see that the future trajectories generated after the refinement operation for pedestrians P2 and P4 are also closer to the ground-truth trajectories compared to the trajectories before refinement. These results demonstrate that the social refiner in the MRGTraj model has a positive impact on improving the prediction performance for certain samples. While the MRGTraj model without the social refiner already exhibits excellent prediction results compared to baseline models, the refiner may not always provide additional improvement for every sample. Therefore, the effectiveness of the refiner may vary depending on the specific scenarios and pedestrian trajectories.

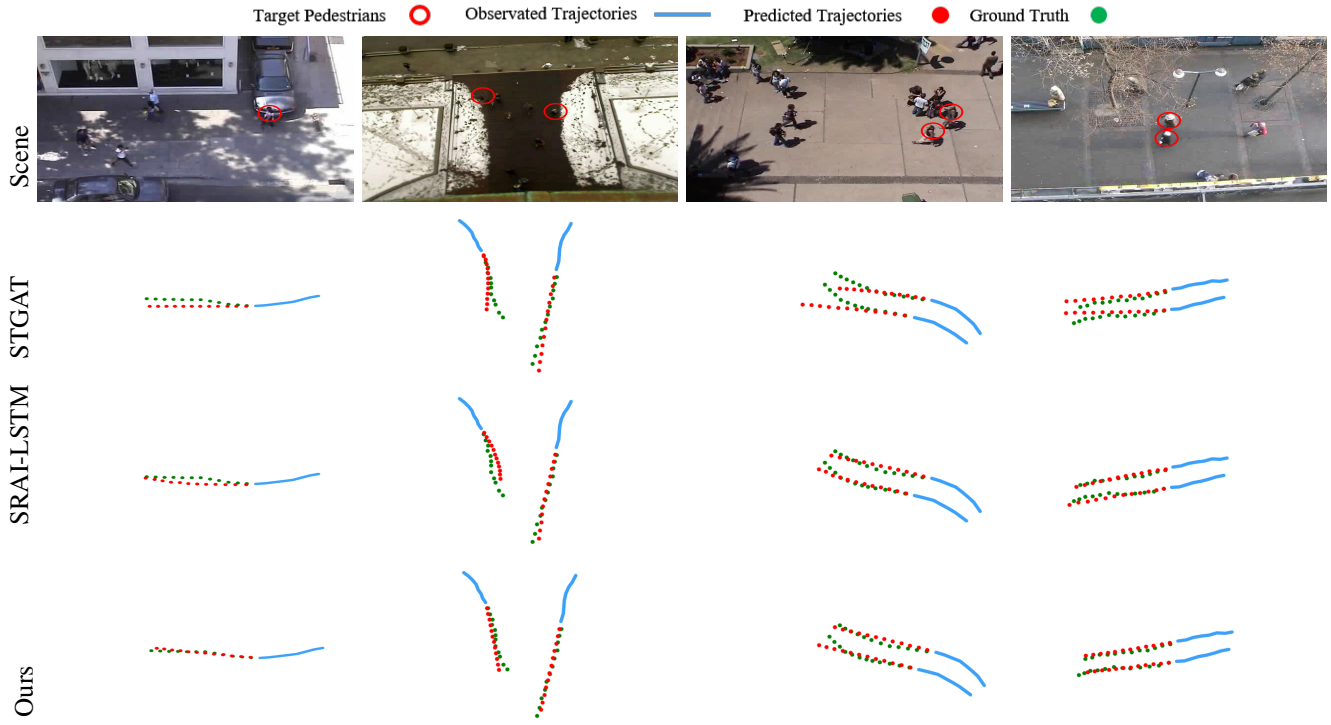


Fig. 9. Comparison of visualizations of prediction results between the STGAT, SRAI-LSTM, and MRGTraj models. The scenario pictures are listed in the first row, and these scenes are from the ZARA1 and ETH datasets. The visualizations of prediction results of STGAT, SRAI-LSTM, and MRGTraj models are drawn in the second, third, and fourth rows.

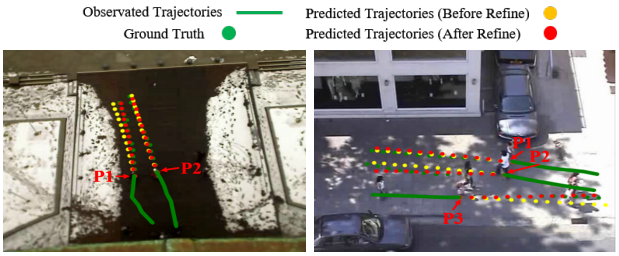


Fig. 10. Comparison of future trajectories generated by decoding features before and after social refiner. Two scenarios from the ETH and ZARA2 datasets are chosen for illustrated.

VI. Conclusion

In this paper, we present the MRGTraj model, which utilizes a non-autoregressive decoder called the MRG decoder. The MRG decoder consists of a temporal mapper, a social refiner, and a generator, and offers several advantages over autoregressive decoders. The temporal mapper enables simultaneous generation of decoding features for all future time steps, reducing inference time and mitigating the error accumulation problem associated with autoregressive decoding. We employ a Transformer as the encoder and the MRG decoder as the decoder in the MRGTraj model for trajectory prediction. Notably, we introduce an interaction-aware latent code generator that captures latent codes from the future social interaction context among pedestrians. These latent codes, combined with the decoding features from the temporal mapper, are used to generate multiple socially acceptable future trajectories. The quantitative evaluation results demonstrate the effectiveness and superior performance of the MRGTraj model. Ablation experiments highlight the benefits of the MRG decoder in reducing error accumulation and achieving faster inference. Moreover, the social interaction-aware latent code generator proves its superiority in generating multimodal trajectories. In future work, we plan to explore the integration of the MRG decoder and the social interaction-aware latent code generator into state-of-the-art trajectory prediction models to further assess their effectiveness and versatility. This will allow us to validate their performance across different prediction frameworks and potentially enhance the overall prediction capabilities. Exploring the deployment of the model into online prediction systems is a challenging and worthwhile research direction.

References

[1] X. T. Truong and T. D. Ngo, "Toward socially aware robot navigation in dynamic and crowded environments: A proactive social motion model," *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 4, pp. 1743–1760, Aug. 2017.

[2] X. T. Truong and T.-D. Ngo, ""To approach humans?": A unified framework for approaching pose prediction and socially aware robot navigation," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 10, no. 3, pp. 557–572, Sep. 2018.

[3] Y. Che, A. M. Okamura, and D. Sadigh, "Efficient and trustworthy social navigation via explicit and implicit robot–human communication," *IEEE Trans. Robot.*, vol. 36, no. 3, pp. 692–707, Jun. 2020.

[4] S. S. Samsani and M. S. Muhammad, "Socially compliant robot navigation in crowded environment by human behavior resemblance using deep reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5223–5230, Jul. 2021.

[5] F. Camara, N. Bellotto, S. Cosar, F. Weber, D. Nathanael, M. Althoff, and et al., "Pedestrian models for autonomous driving part ii: High-level models of human behavior," *IEEE Transs Intell. Transp. Sys.*, vol. 22, no. 9, pp. 5453–5472, Sep. 2021.

[6] L. Hou, L. Xin, S. E. Li, B. Cheng, and W. Wang, "Interactive trajectory prediction of surrounding road users for autonomous driving using structural-lstm network," *IEEE Transs Intell. Transp. Sys.*, vol. 21, no. 11, pp. 4615–4625, Nov. 2020.

[7] L. Qin, Z. Huang, C. Zhang, H. Guo, M. H. Ang, and D. Rus, "Deep imitation learning for autonomous navigation in dynamic pedestrian environments," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Jun. 2021, pp. 4108–4115.

[8] Y. Xu, D. Ren, M. Li, Y. Chen, M. Fan, and H. Xia, "Tra2tra: Trajectory-to-trajectory prediction with a global social spatial-temporal attentive neural network," *IEEE Robot. Autom. Let.*, vol. 6, no. 2, pp. 1574–1581, Apr. 2021.

[9] N. Shafiee, T. Padir, and E. Elhamifar, "Introvert: Human trajectory prediction via conditional 3d attention," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 16 815–16 825.

[10] P. Kothari, B. Sifringer, and A. Alahi, "Interpretable social anchors for human trajectory forecasting in crowds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 15 556–15 566.

[11] G. Chen, J. Li, N. Zhou, L. Ren, and J. Lu, "Personalized trajectory prediction via distribution discrimination," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 15 580–15 589.

[12] Y. Peng, G. Zhang, X. Li, and L. Zheng, "SRGAT: social relational graph attention network for human trajectory prediction," in *Proc. Int. Conf. Neural Inf. Process. (ICONIP)*, Dec. 2021, pp. 632–644.

[13] F. Giuliari, I. Hasan, M. Cristani, and F. Galasso, "Transformer networks for trajectory forecasting," in *Proc. Int. Conf. Pattern Recognit. (ICPR)*, Jan. 2021, pp. 10 335–10 342.

[14] Y. Yuan, X. Weng, Y. Ou, and K. Kitani, "AgentFormer: Agent-aware transformers for socio-temporal multi-agent forecasting," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9813–9823.

[15] L. Li, M. Pagnucco, and Y. Song, "Graph-based spatial transformer with memory replay for multi-future pedestrian trajectory prediction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 2231–2241.

[16] L. Zhou, D. Yang, X. Zhai, S. Wu, Z. Hu, and J. Liu, "GA-STT : Human Trajectory Prediction with Group Aware Spatial-Temporal Transformer," *IEEE Robot. Autom. Let.*, vol. 7, no. 3, pp. 7660–7667, 2022.

[17] W. Chen, Z. Yang, L. Xue, J. Duan, H. Sun, and N. Zheng, "Multimodal Pedestrian Trajectory Prediction using Probabilistic Proposal Network," *IEEE Trans. Circuits Syst. Video Technol., early access*, pp. 1–15, 2022, doi: 10.1109/TCSVT.2022.3229694.

[18] A. Mohamed, K. Qian, M. Elhoseiny, and C. Claudel, "Social-STGCNN: A social spatio-temporal graph convolutional neural network for human trajectory prediction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 14 412–14 420.

[19] H. Zhou, D. Ren, H. Xia, M. Fan, X. Yang, and H. Huang, "AST-GNN: An attention-based spatio-temporal graph neural network for interaction-aware pedestrian trajectory prediction," *Neurocomputing*, vol. 445, pp. 298–308, Jul. 2021.

[20] K. Li, S. Eiffert, M. Shan, F. Gomez-Donoso, S. Worrall, and E. M. Nebot, "Attentional-GCNN: Adaptive pedestrian trajectory prediction towards generic autonomous vehicle use cases," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Jun. 2021, pp. 14 241–14 247.

[21] 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 *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 8994–9003.

[22] I. Bae and H. Jeon, "Disentangled multi-relational graph convolutional network for pedestrian trajectory prediction," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, Feb. 2021, pp. 911–919.

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

[24] K. Cho, B. v. Merrienboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proc. Conf. Empirical Methods in Natural Language Process. (EMNLP)*, Oct. 2014, pp. 1724–1734.

[25] L. Wu, Y. Wang, X. Li, and J. Gao, "Deep attention-based spatially recursive networks for fine-grained visual recognition," *IEEE Trans. Cyber.*, vol. 49, no. 5, pp. 1791–1802, 2019.

[26] D. Liu, L. Wu, X. Li, and L. Qi, "Medi-care ai: Predicting medications from billing codes via robust recurrent neural networks," *Neural Networks*, vol. 124, pp. 109–116, 2020.

[27] K. Zhang, N. Liu, X. Yuan, X. Guo, C. Gao, Z. Zhao, and Z. Ma, "Fine-grained age estimation in the wild with attention lstm networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 9, pp. 3140–3152, 2020.

[28] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social LSTM: Human trajectory prediction in crowded spaces," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 961–971.

[29] K. Xu, Z. Qin, G. Wang, K. Huang, S. Ye, and H. Zhang, "Collision-free LSTM for human trajectory prediction," in *Proc. Int. Conf. MultiMedia Model.*, vol. 10704, Feb. 2018, pp. 106–116.

[30] N. Bisagno, C. Saltori, B. Zhang, F. G. De Natale, and N. Conci, "Embedding group and obstacle information in lstm networks for human trajectory prediction in crowded scenes," *Computer Vision and Image Understanding*, vol. 203, p. 103126, Feb. 2021.

[31] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social GAN: Socially acceptable trajectories with generative adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 2255–2264.

[32] Y. Huang, H. Bi, Z. Li, T. Mao, and Z. Wang, "Stgat: Modeling spatial-temporal interactions for human trajectory prediction," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6271–6280.

[33] P. Zhang, W. Ouyang, P. Zhang, J. Xue, and N. Zheng, "SR-LSTM: State refinement for LSTM towards pedestrian trajectory prediction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12 085–12 094.

[34] P. Zhang, J. Xue, P. Zhang, N. Zheng, and W. Ouyang, "Social-aware pedestrian trajectory prediction via states refinement LSTM," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 5, pp. 2742–2759, May. 2022.

[35] Z. Su, S. Zhang, and W. Hua, "Cr-lstm: Collision-prior guided social refinement for pedestrian trajectory prediction," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. and Syst. (IROS)*, Oct. 2021, pp. 1427–1433.

[36] A. Bertugli, S. Calderara, P. Coscia, L. Ballan, and R. Cucchiara, "Ac-vrnn: Attentive conditional-vrnn for multi-future trajectory prediction," *Comput. Vis. Image Und.*, vol. 210, p. 103245, Sep. 2021.

[37] 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 *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1349–1358.

[38] V. Kosaraju, A. Sadeghian, R. Martín-Martín, I. D. Reid, H. Rezatofighi, and S. Savarese, "Social-BiGAT: Multimodal trajectory forecasting using bicycle-GAN and graph attention networks," in *Proc. Adv. Neural Inf. Process. Syst (NeurIPS)*, 2019, pp. 137–146.

[39] T. Zhao, Y. Xu, M. Monfort, W. Choi, C. L. Baker, Y. Zhao, Y. Wang, and Y. Wu, "Multi-agent tensor fusion for contextual trajectory prediction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12 126–12 134.

[40] B. Ivanovic and M. Pavone, "The Trajectron: Probabilistic Multi-Agent Trajectory Modeling With Dynamic Spatiotemporal Graphs," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 2375–2384.

[41] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, "Trajectron ++ : Dynamically-Feasible Trajectory Forecasting with Heterogeneous Data," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Springer, Aug. 2020, pp. 683–700.

[42] H. Cheng, W. Liao, X. Tang, M. Y. Yang, M. Sester, and B. Rosenhahn, "Exploring Dynamic Context for Multi-path Trajectory Prediction," *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pp. 12 795–12 801, Jun. 2021.

[43] K. Mangalam, H. Girase, S. Agarwal, K. Lee, E. Adeli, J. Malik, and A. Gaidon, "It is not the journey but the destination: Endpoint conditioned trajectory prediction," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Aug. 2020, pp. 759–776.

[44] H. Sun, Z. Zhao, and Z. He, "Reciprocal twin networks for pedestrian motion learning and future path prediction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 3, pp. 1483–1497, Mar. 2022.

[45] Y. Xu, Z. Piao, and S. Gao, "Encoding Crowd Interaction with Deep Neural Network for Pedestrian Trajectory Prediction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*. IEEE, Jun. 2018, pp. 5275–5284.

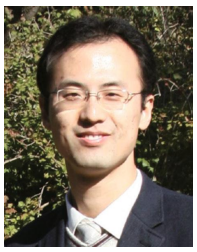
[46] Y. Hu, S. Chen, Y. Zhang, and X. Gu, "Collaborative motion prediction via neural motion message passing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6318–6327.

[47] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, and et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst (NeurIPS)*, Dec. 2017, pp. 5998–6008.

[48] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, Jun. 2019, pp. 4171–4186.

[49] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Aug. 2020, pp. 213–229.
[50] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, and et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, May. 2021.
[51] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *Proc. Int. Conf. Mach. Learn. (ICML)*, vol. 139, Jul. 2021, pp. 10 347–10 357.
[52] Y. Peng, G. Zhang, J. Shi, B. Xu, and L. Zheng, "Srai-lstm: A social relation attention-based interaction-aware LSTM for human trajectory prediction," *Neurocomputing*, vol. 490, pp. 258–268, Jun. 2022.
[53] A. D. Berenguer, M. Alioscha-Pérez, M. C. Oveneke, and H. Sahli, "Context-aware human trajectories prediction via latent variational model," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 5, pp. 1876–1889, May. 2021.
[54] S. Pellegrini, A. ESS, K. Schindler, and L. V. Gool, "You'll never walk alone: Modeling social behavior for multi-target tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Sep. 2009, pp. 261–268.
[55] A. Lerner, Y. Chrysanthou, and D. Lischinski, "Crowds by example," *Comput. Graph. Forum*, vol. 26, no. 3, pp. 655–664, Aug. 2007.
[56] J. Sekhon and C. Fleming, "SCAN: A spatial context attentive network for joint multi-agent intent prediction," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, Feb. 2021, pp. 6119–6127.
[57] C. Li, H. Yang, and J. Sun, "Intention-interaction graph based hierarchical reasoning networks for human trajectory prediction," *IEEE Trans. Multimedia, early access*, Jun. 13, 2022, doi: 10.1109/TMM.2022.3182151.
[58] Y. Peng, G. Zhang, X. Li, and L. Zheng, "STIRNet: A spatial-temporal interaction-aware recursive network for human trajectory prediction," in *Proc. IEEE Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2021, pp. 2285–2293.
[59] P. Dendorfer, S. Elflein, and L. Leal-Taixé, "MG-GAN: A multi-generator model preventing out-of-distribution samples in pedestrian trajectory prediction," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 13 158–13 167.
[60] C. Yu, X. Ma, J. Ren, H. Zhao, and S. Yi, "Spatio-temporal graph transformer networks for pedestrian trajectory prediction," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Aug. 2020, pp. 507–523.
[61] X. Li, Y. Peng, W. Wu, G. Zhang, and L. Zheng, "Pecgan: Endpoint conditioned trajectory prediction via generative adversarial network," in *Proc. China Automation Congress (CAC)*, Oct. 2021, pp. 7411–7416.

Yusheng Peng received the M.S. degree in Computational Mathematics from Anqing Normal University, China, in 2017. He is currently pursuing an Ph.D. degree at Hefei University of Technology, China. His current research interests include trajectory prediction, crowd evacuation and simulation, and software engineering.



Gaofeng Zhang received the Ph.D. degree in ICT from Swinburne University of Technology (SUT), Australia, in 2013. Now, he is an associate professor at Hefei University of Technology, China. His research interests include cloud/edge computing, software security, public safety and software engineering.

Jun Shi received the Ph.D. degree in pattern recognition and intelligent systems from Beihang University, China, in 2011. Now, he is an associate professor at Hefei University of Technology, China. His research interests include machine learning, medical image analysis and remote sensing image understanding.

Xiangyu Li Xiangyu Li received the M.S. degreee in computer science and technology from Hefei University of Technology, China, in 2022. He is currently working at iFLYTEK. His current research interests include trajectory prediction and software engineering.

Liping Zheng received the M.S. and Ph.D. degrees in computer science from Hefei University of Technology, Hefei, China, in 2003 and 2008, respectively. He is currently a professor at Hefei University of Technology, China. His research interests computer graphics, computer simulation, deep learning, and computer vision.