2023 IEEE/CVF International Conference on Computer Vision (ICCV) | 979-8-3503-0718-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICCV51070.2023.00887

Trajectory Unified Transformer for Pedestrian Trajectory Prediction

Liushuai Shi[1] Le Wang[1*] Sanping Zhou[1] Gang Hua[2]
[1]National Key Laboratory of Human-Machine Hybrid Augmented Intelligence,
National Engineering Research Center for Visual Information and Applications,
Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University
[2]Wormpex AI Research

Abstract

Pedestrian trajectory prediction is an essential link to understanding human behavior. Recent work achieves state-of-the-art performance gained from hand-designed post-processing, e.g., clustering. However, this post-processing suffers from expensive inference time and neglects the probability that the predicted trajectory disturbs downstream safety decisions. In this paper, we present Trajectory Unified TRansformer, called TUTR, which unifies the trajectory prediction components, social interaction, and multimodal trajectory prediction, into a transformer encoder-decoder architecture to effectively remove the need for post-processing. Specifically, TUTR parses the relationships across various motion modes using an explicit global prediction and an implicit mode-level transformer encoder. Then, TUTR attends to the social interactions with neighbors by a social-level transformer decoder. Finally, a dual prediction forecasts diverse trajectories and corresponding probabilities in parallel without post-processing. TUTR achieves state-of-the-art accuracy performance and improvements in inference speed of about $10\times$ - $40\times$ *compared to previous well-tuned state-of-the-art methods using post-processing.*

1. Introduction

Pedestrian trajectory prediction aims to predict the future trajectory based on an observed trajectory. It is an essential link that connects the perception system upward and the planning system downward [13, 18]. Due to the randomness of human motion, there are diverse plausible future trajectories that pedestrians could take [6]. The popular predictor addresses this multimodal prediction task in a generative style. They model the multimodality of the future trajectory in a specific space, such as an explicit Gaussian space [17, 27, 5], a latent space [34, 16, 37, 6], a hand-planning space [26], or a memory bank [33].

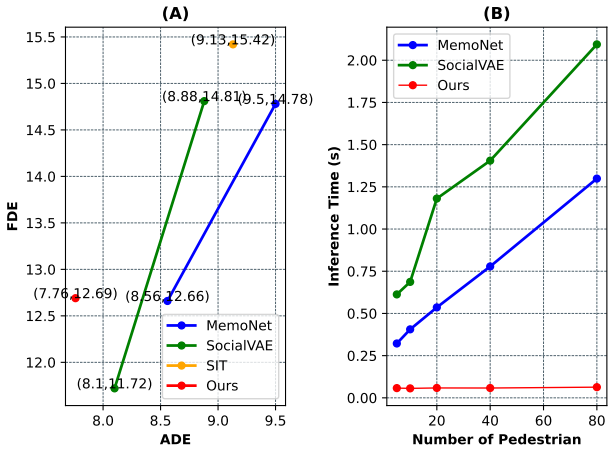

Figure 1. (A) shows the comparison against the methods (MemoNet[33], SocialVAE[34]) with post-processing and the SIT[26] without post-processing, Simultaneously, it presents the accuracy performance variances of MemoNet and SocialVAE that use post-processing or not. (B) shows the inference speed variances as the number of pedestrians increases. Our method achieves a balance of accuracy performance and inference speed.

Recently, some works [33, 34] have achieved significant advances benefiting from hand-designed post-processing, as illustrated in Figure 1 (A). Most of the time, they first sample more plausible future trajectories than the desired number of predictions K. Then, a clustering algorithm (e.g., K-means) is operated on sampled trajectories to generate the desired K predictions, similar to NMS [9] in object detection. However, this post-processing suffers from expensive inference time due to the non-parallel loop iteration in clustering, especially for a dense scene.

As shown in Figure 1 (B), the methods with post-processing lead to being more and more time-consuming as the number of pedestrians increases. Furthermore, the post-processing neglects probability information, disturbing safety decisions. Actually, most works forecast diverse

*Corresponding author.

2380-7504/23/$31.00 ©2023 IEEE
DOI 10.1109/ICCV51070.2023.00887

trajectories equally (*i.e.*, without probability information) in pedestrian trajectory prediction. Similarly, the clustering operation also obtains K centers with equal weights. Although these predictors have significant performance in best-of-K prediction, there is no information on which is the best. It is a disadvantage for the safety decision of an intelligent system such as autonomous driving. Our goal aims to bridge the gap between accuracy and inference speed, keeping the corresponding probabilities of predicted trajectories simultaneously.

To address the above problems, we propose Trajectory Unified TRansformer (named TUTR) to effectively eliminate the need for post-processing in pedestrian trajectory prediction. TUTR unifies the components of pedestrian trajectory prediction, such as social interaction and multimodal trajectory prediction, into a transformer encoder-decoder architecture as illustrated in Figure 2.

TUTR first parses relationships across various motion modes using an explicit global prediction and an implicit mode-level transformer decoder. Specifically, global prediction employs two rigid transformations on training trajectories to obtain general motion modes, which are considered as the input token of the mode-level transformer encoder. Afterward, TUTR attends to the social interactions with neighbors using a social-level transformer decoder to prepare a social-acceptable prediction. Finally, a dual prediction is used to forecast diverse trajectories and corresponding probabilities in parallel to cover the multimodality of future trajectories without any post-processing.

We evaluate TUTR on the most popular datasets for pedestrian trajectory prediction, ETH [19], UCY [14], and SDD [22]. The experimental results show that our proposed method achieves a comparable accuracy performance and faster inference speed without any post-processing step compared with existing state-of-the-art methods. Moreover, TUTR performs the best performance in brier-ADE and brier-FDE, which are two metrics that consider the probabilities of predicted trajectories.

In summary, the contributions of this paper are summarized as follows.

- We propose a new pedestrian trajectory prediction framework (TUTR) based on encoder-decoder transformer architecture entirely to unify the pedestrian trajectory prediction.
- TUTR parses the relationship across various motion modes by the explicit global prediction and implicit mode-level transformer encoder to effectively remove the need for post-processing.
- TUTR achieves state-of-the-art ADE/brier-ADE/brier-FDE, and comparable performance in FDE. Moreover, TUTR performs a faster inference speed to balance accuracy performance and inference speed.

2. Related Works

Research on pedestrian trajectory prediction is briefly categorized into two classes: prediction based on environment information (*e.g.*, semantic map) [2, 15, 25, 21, 38, 30] and prediction based on social interaction from neighbors. In this paper, we focus on the latter to effectively remove the need for post-processing by unifying the pedestrian trajectory prediction into an encoder-decoder architecture.

2.1. Pedestrian Trajectory Prediction

Physical Models. Before deep learning, many works design specific physical models to forecast a deterministic future trajectory. Social force [8], motion velocity [28], and energy [11] are commonly used to model the motion behavior of pedestrians. Also, some works employ the statistical model, such as Gaussian processes [31, 12], to deal with the uncertainty of future trajectories. However, they suffer from bad generalizations when facing various motion patterns and social interactions.

Deep Learning Models. As deep learning develops in the community, most deep models in pedestrian trajectory prediction forecast future trajectories via feature extraction and multimodal trajectory prediction. In feature extraction, many works use deep models, such as recurrent neural networks (RNNs) [1, 6, 34], attention mechanisms [23, 27, 26, 36, 37], and graph neural networks (GCNs) [17], to model the temporal sequential features from the observed trajectory and spatial interaction features from neighbors.

Multimodal Trajectory Prediction. Pedestrians could take various future trajectories due to their motion randomness [6]. To deal with such multimodal prediction tasks, many works employ generative models, such as generative adversarial networks (GANs) [6] and conditional variational autoencoder (CVAE) [16, 34, 37, 24], to generate diverse future trajectories. Besides, some works [26, 17, 5] model the possible future trajectories into a Gaussian distribution or a Gaussian Mixture Model (GMM). A tree-based model [26] covers the possible future trajectories by an interpretable tree. In addition, the memory-based methods [33] store the diverse trajectories in a memory bank. Recently, the post-processing step is commonly used to improve the diversity of predicted trajectories. PECNet [16] changes the variance of latent space. AgentFormer [37] penalizes the pairwise distance of predicted trajectories. However, a more effective post-processing step [33, 34] is sampling a large number of predicted trajectories and then clustering them into the desired number of centers. Unfortunately, the post-processing step, especially for the clustering, suffers from the expensive inference time and loses the probability of predicted trajectories. In contrast, TUTR can directly forecast diverse trajectories without any post-processing step and achieves a balance between inference time and accuracy performance.

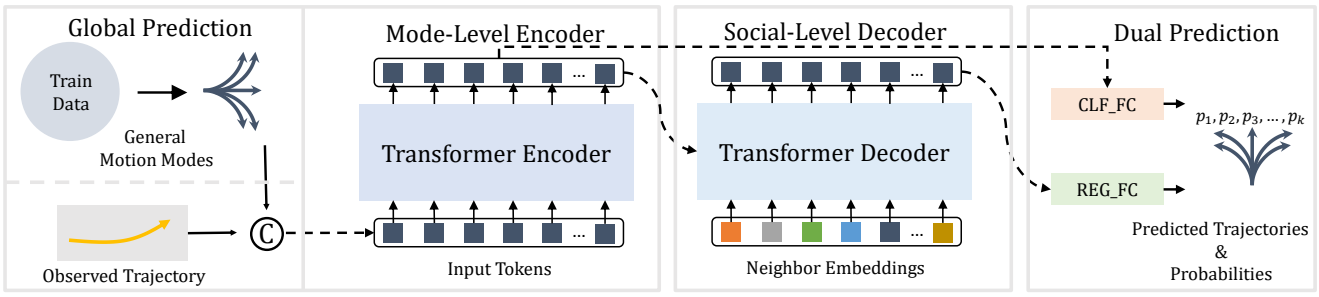

Figure 2. An overview of TUTR. TUTR employs an encoder-decoder transformer architecture to forecast future motion behaviors. Firstly, the global prediction generates general motion modes. Then, the general motion modes concatenated with the observed embedding are considered as the input tokens of a mode-level transformer encoder. Subsequently, the encoder output attends to the social interactions by a social-level decoder. Finally, two shared prediction heads in dual prediction are used to obtain the dual results, *i.e.*, predicted trajectories, and corresponding probabilities.

2.2. Transformers

The Transformer model [29] is first proposed in the machine translation task to replace the recurrent neural networks (RNNs) [10]. Transformers are now popular in many tasks, such as in natural language processing [35, 20] and computer vision [3, 4]. Transformers encode global features through the self-attention mechanism in parallel. Then, the encoder-decoder attention (cross-attention) in the Transformer decoder generates the desired output. In the naive Transformer, the decoder is an auto-regressive model to output tokens one by one.

Unlike previous applications of the transformer to extract global features, TUTR is mainly used to address the question of output, *i.e.*, multimodal trajectory prediction similar to [3] in object detection. Concretely, TUTR first design a global prediction on the whole training trajectories to obtain general motion behaviors, which are considered as the input tokens of the encoder of the transformer. Then, a decoder attends to social interactions with neighbors and the results of the decoder to forecast diverse trajectories in parallel, not the autoregressive style.

Some methods [37, 36] have explored the Transformer [29] architecture in the prediction of pedestrian trajectory. However, the transformer is only used to extract temporal and spatial features. Besides, they employ an auto-regressive model to output trajectory points one by one. Compared to them, TUTR unifies the pedestrian trajectory prediction modules, such as feature extraction and multimodal trajectory prediction, into an encoder-decoder transformer architecture, which includes a mode-level encoder, a social-level decoder, and two dual prediction heads. It achieves better performance and contributes to compatibility with other modules, such as upward motion perception and downstream motion planning. What's more, TUTR employs parallel decoding to generate diverse trajectories, further improving the inference speed compared with auto-regressive decoding.

3. Our Method

3.1. Problem Definition

Pedestrian trajectory prediction aims to forecast the future trajectory of a pedestrian based on the observed trajectories of the pedestrian and its neighbors. Assume that a sequence of traffic scenes with the length T contains N pedestrians. We extract N trajectory coordinate sequences $\{x_t^n, y_t^n\}_{t=1,n=1}^{T,N}$ for each pedestrian at each time step. The trajectory model observes the front of sub-trajectories $\{x_t^n, y_t^n\}_{t=1,n=1}^{T_{obs},N}$ and predicts the next sub-trajectories $\{x_t^n, y_t^n\}_{t=T_{obs}+1,n=1}^{T,N}$. Due to the multimodality of pedestrian motion behavior, there are multiple future trajectories that the pedestrian could take. Therefore, the trajectory model is required to forecast diverse future trajectories, while only a single true future trajectory (ground truth) is provided for model training.

3.2. TUTR Architecture

Here, we introduce our proposed Trajectory Unified TRansformer (TUTR), which contains four components packed into a transformer encoder-decoder architecture to forecast diverse future behaviors as illustrated in Figure 2. The explicit *global prediction* and the implicit *mode-level transformer encoder* are used to parse the relationships across various motion modes. Subsequently, the encoder output attends to the social interactions with neighbors using a *social-level transformer decoder*. Finally, a *dual prediction* is used to obtain dual results (diverse future trajectories and corresponding probabilities) in parallel by two shared prediction heads.

Recall that previous transformer-based methods [36, 37] employ the transformer architecture only on the observed trajectory and its neighbors. Namely, the trajectory points of an observed trajectory are the input tokens of a temporal transformer encoder to obtain temporal features. The trajectory points of the neighbors are the input tokens of a spatial

transformer encoder to obtain spatial features. However, the multimodality of the future trajectory is the main challenge that affects prediction accuracy. Unlike them, TUTR parses the relationship across various modes of future behavior by a *mode-level* transformer encoder. Then, TUTR attends to the social interactions using a *social-level* transformer decoder to directly output the diverse future trajectories without any post-processing step.

Global Prediction. TUTR parses the relationships across various motion modes using an explicit global prediction and an implicit mode-level transformer encoder. Global prediction obtains general motion modes to cover common motion behaviors of a pedestrian, and the results are considered as the input token of the next mode-level transformer encoder. Here, we first employ two rigid transformations to generate normalized trajectories and then use a distance measurement to obtain the general motion modes.

Given a fixed view, the trajectory is invariant for the rigid transformation. For example, a pedestrian shows going straight and then turning left. It also shows the same behaviors after translation or rotation for the trajectory of the pedestrian. For the training trajectories with length T, the front sub-trajectories with length T_{obs} are the observed trajectories, while the next sub-trajectories with length T_{pred} are the future trajectories. We first translate the T_{obs} trajectory points of the trajectories into the origin of the coordinate system. Then, the initial trajectory points of the translated trajectories are rotated to the positive X-axis. In this case, the direction of most future trajectories is normalized to a relatively fixed region. Namely, the trajectories with similar motion behaviors could have a small distance. Thus, we can obtain diverse trajectories explicitly in a distance measurement strategy to cover the common motion behaviors.

Therefore, a clustering operation is used on the normalized future trajectories to obtain L centers $C \in \mathbb{R}^{L\times T_{pred}\times 2}$, where $C = \{c_1, ..., c_L\}$ and each $\{c_l | l \in 1, ..., L\}$ is a trajectory with length T_{pred}. Thus, the centers C represent the general motion modes, which are the input tokens of the next mode-level transformer encoder. Note that C is invariant in the inference step. Namely, global prediction does not lead to additional inference time.

Observed embedding. The general motion modes are considered as the input token of the next described mode-level transformer encoder. They are first reshaped into a $L \times 2T_{pred}$ features and then embedded by a learnable linear transformation to obtain input embeddings E_c as follows:

$$E_c = \phi(C, \boldsymbol{W}_c), \tag{1}$$

where $\phi(\cdot, \cdot)$ is a linear transformation with a learnable parameter matrix $\boldsymbol{W}_c \in \mathbb{R}^{2T_{pred}\times D_e}$, $E_c \in \mathbb{R}^{L\times D_e}$ is the input embeddings.

The previous input embeddings of the transformers need an extra positional embedding [29] to deal with the permutation-invariant of the self-attention mechanism. Unlike them, the elements in C are not limited by their sequences. Thus, the positional embedding is not necessary for the input embedding E_c. However, our goal is to predict diverse trajectories of a pedestrian based on its observed motion states. The input embeddings E_c are required to fit the given input observed trajectory $X \in \mathbb{R}^{B\times T_{obs}\times 2}$, where B is the batch size. Hence, the observed trajectory X is embedded and added to the input embeddings E_c as follows:

$$\begin{aligned} E_o &= \phi(X, \boldsymbol{W}_o), \\ E_e &= E_c + E_o, \end{aligned} \tag{2}$$

where X is reshaped into $B \times 2T_{obs}$ before the linear transformation, $\boldsymbol{W}_o \in \mathbb{R}^{2T_{obs}\times D_e}$ is the learnable parameter matrix. We broadcast the dimensions of E_c and E_o to $B \times L \times D_e$ and perform an add operation between them to obtain the final embedding $E_e \in \mathbb{R}^{B\times L\times D_e}$.

Mode-Level Transformer Encoder. Unlike feature-level transformer encoders to build the global dependence of trajectory points, the mode-level transformer encoder parses the relationships across various modes. Given the input embedding E_e that represents general motion modes based on the observed trajectory, the mode-level transformer encoder employs the standard encoder architecture of a naive transformer on E_e to parse the relationships across various motion modes. Each encoder block includes a multi-head self-attention layer and a Feed-Forward Network (FFN) with the residual connection [7]. Unlike the naive transformer encoder, which adds positional embedding at each encoder block, the observed embedding occurs only once.

Social-Level Transformer Decoder. This decoder is used to extract social interactions with neighbors. It follows the standard decoder architecture of a naive transformer, including an attention layer and a Feed-Forward Network (FFN). The differences with naive transformers lie in four aspects: First, the decoder receives neighboring embeddings, not masked output embeddings. Second, TUTR keeps the encoder-decoder attention and empirically removes the self-attention. Third, the positional embedding is not necessary for the input embedding because the trajectory coordinates have shown the position relationships between pedestrians and their neighbors. Finally, the output embeddings are decoded into diverse future trajectories and corresponding probabilities by the next described dual-prediction in parallel, not the autoregressive style.

Assume that a pedestrian has N neighbors, represented by the neighbor observed trajectories $X_s \in \mathbb{R}^{N\times T_{obs}\times 2}$. Each trajectory in X_s is flattened into a feature vector, leading to a feature matrix $\hat{X}_s \in \mathbb{R}^{N\times 2T_{obs}}$. Then, we embed the feature matrix by a learnable linear transformation to

obtain the input embeddings of the social-level transformer decoder as follows:

$$E_s = \phi(\hat{X}_s, \boldsymbol{W}_s), \tag{3}$$

where $E_s \in \mathbb{R}^{N \times D_e}$ is the input embeddings of the decoder, $\boldsymbol{W}_s \in \mathbb{R}^{2T_{obs} \times D_e}$ is the learnable parameter matrix. After that, the input embeddings E_s are transformed into output embeddings with the subsequent encoder-decoder attention layer and an FFN layer with the residual connection. In this case, these output embeddings attend to the social interactions to forecast social-acceptable trajectories and corresponding probabilities by the next dual prediction.

Dual Prediction. Most previous methods [16, 37, 33, 34] predict diverse future trajectories but neglect the probabilities of predicted trajectories. It is a disadvantage to the safety decision. Here, we use a dual prediction to achieve regression and classification tasks simultaneously. As illustrated in Figure 2, a shared regression prediction head (REG_FC) and a shared classification prediction head (CLF_FC) are used to forecast diverse future trajectories and corresponding probabilities, respectively. In the implementation, we empirically find that placing the classification prediction head to the back of the mode-level encoder could bring better accuracy performance.

Model Training. Due to a single provided true future trajectory (ground truth) $\hat{Y}$ for multimodal trajectory prediction, we employ a greedy training strategy. Specifically, we first obtain the closest clustering centers $c_i, i \in \{1, ..., L\}$ by distance measurement between the ground truth $\hat{Y}$ and L clustering centers $C = \{c_1, ..., c_L\}$ as follows:

$$i = \underset{i \in \{1,...,L\}}{\operatorname{argmin}} (||\hat{Y} - c_i||_2^2). \tag{4}$$

Next, we employ a *nearest neighbor hypothesis*, which means the ground truth $\hat{Y}$ can be obtained by a (deep) transformation of the closest centers c_i, and the predicted trajectory obtained from c_i is the most likely one, *i.e.*, owning the maximum probability. The soft probability $\hat{p}$ of c_i can be represented by the normalized negative distance as follows:

$$p = \operatorname{softmax}(\{-||\hat{Y} - c_i||_2^2 \mid i \in \{1, ..., L\}\}). \tag{5}$$

Consequently, TUTR is used to transform c_i into the desired $\hat{Y}$. In this case, we could predict the future trajectory and its probability with the current motion mode by the ith output embedding of the decoder in the training step, resulting in a predicted trajectory Y and the corresponding soft probabilities p. Finally, TUTR can be trained in an end-to-end way as follows:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{reg}(Y, \hat{Y}) + \lambda_2 \mathcal{L}_{clf}(p, \hat{p}), \tag{6}$$

where λ_1 and λ_2 are used to balance the loss function, $\mathcal{L}_{reg}$ is the Huber loss, and $\mathcal{L}_{clf}$ is the cross entropy loss.

In the inference step, TUTR outputs multiple predicted trajectories and selects K predicted trajectories with Top-K probabilities to cover diverse motion behaviors.

4. Experiments and Discussions

In this section, we show that TUTR achieves a comparable accuracy performance and faster inference speed compared to existing state-of-the-art methods that benefit from the well-designed post-processing step. In addition, we provide detailed ablation studies on components of the proposed method. Finally, we further evaluate the effectiveness of TUTR by qualitative visualization evaluation.

4.1. Experiments Setting

Datasets. We conduct experiments on two benchmark datasets, *i.e.*, ETH-UCY [19, 14], and Stanford Drone Dataset (SDD) [22], to evaluate our proposed method. ETH-UCY is the most widely used benchmark for pedestrian trajectory prediction. It contains trajectories of 1,536 pedestrians collected in four different scenarios with a bird's eye view and divided into five subsets, ETH, HOTEL, UNIV, ZARA1, and ZARA2. On ETH-UCY, we follow prior works [33, 34] that use a leave-one-out method for model training. Namely, we train the proposed model on four subsets and test it on the rest of the subsets. SDD is a larger benchmark dataset in pedestrian trajectory prediction, also captured by bird's eye view. It contains the trajectories of 5,232 pedestrians recorded in eight different scenarios. On SDD, we use the previous train-test split [16] to train and test our proposed model. The model observes the trajectory with length $T_{obs} = 8$ (3.2 seconds) and predicts the next $T_{pred} = 12$ (4.8 seconds) trajectory.

Evaluation Metrics. We evaluate our proposed and compared methods by four metrics, *i.g.*, Average Displacement Error (ADE) and Final Displacement Error (FDE), brier-ADE and brier-FDE. Given a true future trajectory (ground truth) $\{x_t, y_t\}_{t=T_{obs}+1}^{T}$ and the corresponding predicted K trajectories, ADE and FDE are used to measure the $\ell 2$ distance between ground truth and the corresponding closest predicted trajectory $\{\hat{x}_t, \hat{y}_t\}_{t=T_{obs}+1}^{T}$,as shown in Eq. (7).

$$\begin{aligned} \text{ADE} &= \frac{1}{T_{pred}} \sum_{t=T_{obs}}^{T} \sqrt{(x_t - \hat{x}_t)^2 + (y_t - \hat{y}_t)^2}, \\ \text{FDE} &= \sqrt{(x_T - \hat{x}_T)^2 + (y_T - \hat{y}_T)^2}. \end{aligned} \tag{7}$$

brier-ADE and brier-FDE [32] are similar to ADE and FDE but add the probability p of the closest predicted trajectory, as shown in Eq. (8):

Method	Venue/Year	ETH	HOTEL	UNIV	ZARA1	ZARA2	AVG
Social GAN [6]	CVPR2018	0.87/1.62	0.67/1.37	0.76/1.52	0.35/0.68	0.42/0.84	0.61/1.21
SoPhie [23]	CVPR2019	0.70/1.43	0.76/1.67	0.54/1.24	0.30/0.63	0.38/0.78	0.51/1.15
STAR [36]	ECCV2020	**0.36**/0.64	0.17/0.36	0.31/0.62	0.29/0.52	0.22/0.46	0.26/0.53
SGCN [27]	CVPR2021	0.63/1.03	0.32/0.55	0.37/0.70	0.29/0.53	0.25/0.45	0.37/0.65
CAGN [5]	AAAI2022	0.41/0.65	0.13/0.23	0.32/0.54	0.21/0.38	0.16/0.33	0.25/0.43
SIT [26]	AAAI2022	0.39/0.62	0.14/0.22	0.27/0.47	0.19/**0.33**	0.16/0.29	0.23/0.38
SocialVAE [34]	ECCV2022	0.47/0.76	0.14/0.22	0.25/0.47	0.20/0.37	0.14/0.28	0.24/0.42
PECNet [16]	ECCV2020	0.54/0.87	0.18/0.24	0.35/0.60	0.22/0.39	0.17/0.30	0.29/0.48
AgentFormer [37]	ICCV2021	0.45/0.75	0.14/0.22	0.25/0.45	0.18/0.30	0.14/0.24	0.23/0.39
MemoNet [33]	CVPR2022	**0.40**/0.61	**0.11/0.17**	0.24/0.43	0.18/0.32	0.14/0.24	**0.21**/0.35
SocialVAE+FPC [34]	ECCV2022	0.41/**0.58**	0.13/0.19	**0.21/0.36**	**0.17/0.29**	**0.13/0.22**	**0.21/0.32**
Ours (TUTR)	-	**0.40**/0.61	**0.11**/0.18	0.23/0.42	0.18/0.34	**0.13**/0.25	**0.21**/0.36

Table 1. Comparison with state-of-the-art methods on ETH-UCY in ADE/FDE. The first block is the comparisons against the methods without the post-processing step, while the second block is the comparisons against the methods with the post-processing step.

$$\begin{aligned} \text{brier-ADE} &= \text{ADE} + (1-p)^2, \\ \text{brier-FDE} &= \text{FDE} + (1-p)^2. \end{aligned} \tag{8}$$

Implementation Details. In our conducted experiments, the trajectories are translated to the origin and then rotated to the X-axis to be consistent with the general motion modes. On ETH-UCY, the number of general motion modes $L = 50, 90, 50, 70, 50$ for the ETH, HOTEL, UNIV, ZARA1, and ZARA2, respectively. The embedding dimension D_e is equal to 128. We stack 2 mode-level transformer encoders with 4 attention heads and 128 FFN hidden dimensions. We stack 1 social-level transformer decoder with 4 attention heads and 128 FFN hidden dimensions. On SDD, the number of general motion modes $L = 100$ and the embedding dimension $D_e = 64$. We stack 2 mode-level transformer encoders with 4 attention heads and 128 FFN hidden dimensions. We stack 1 social-level transformer decoder with 4 attention heads and 128 FFN hidden dimensions. All experiments are conducted on a single RTX 3090 GPU.

4.2. Comparison with State-of-art Methods

Comparison in ADE/FDE on ETH-UCY. As shown in Table 1, the first block shows the comparisons against methods without post-processing. TUTR achieves state-of-the-art performance in both average ADE and average FDE. Specifically, TUTR improves the average ADE/FDE from $0.23/0.38$ to $0.21/0.35$ compared to the previous best method, SIT [26]. The second block in Table 1 shows the comparison against the methods with the post-processing step. TUTR shows competitive performance in average ADE metrics, being on par with the methods (MemoNet [33] and SocialVAE+FPC [34]) with a post-processing step. However, TUTR still shows a performance gap (0.04) in average FDE metrics, against the previous best methods, SocialVAE+FPC [34].

Method	Venue/Year	ADE/FDE
Social GAN [6]	CVPR2018	27.23/41.44
SoPhie [23]	CVPR2019	16.27/29.38
CAGN [5]	AAAI2022	9.42/15.93
SIT [26]	AAAI2022	9.13/15.42
MemoNet [33]	CVPR2022	9.50/14.78
SocialVAE [34]	ECCV2022	8.88/14.81
PECNet [16]	ECCV2020	9.96/15.88
MemoNet [33]	CVPR2022	8.56/12.66
SocialVAE+FPC [34]	ECCV2022	8.10/**11.72**
Ours (TUTR)	-	**7.76**/12.69

Table 2. Comparison with state-of-the-art methods on SDD in ADE/FDE. The first block is the comparisons against the methods without the post-processing step, while the second block is the comparisons against the methods with the post-processing step.

Comparison in ADE/FDE on SDD. As shown in Table 2, the first block shows the comparisons against the methods without post-processing. TUTR also achieves state-of-the-art performance both in ADE and FDE. Specifically, TUTR improves the ADE/FDE from $8.88/14.81$ to $7.79/12.73$ compared with the previous best method, SocialVAE [34]. The second block in Table 2 shows the comparisons against the methods with the post-processing step. TUTR shows state-of-the-art performance in ADE metrics, improving the ADE from 8.10 to 7.79 compared with previous methods, SocialVAE+FPC [34]. However, TUTR also shows a performance gap (1.01) in FDE metrics, against the previous best method, SocialVAE+FPC [34].

Comparison in brier-ADE/FDE. Since many works ne-

Method	Venue/Year	ETH	HOTEL	UNIV	ZARA1	ZARA2	AVG
CAGN* [5]	AAAI2022	1.43/1.78	1.18/1.44	1.47/2.04	1.29/1.78	1.23/1.65	1.32/1.73
SIT [26]	AAAI2022	1.29/1.49	1.03/1.14	1.38/1.82	1.08/1.23	0.99/1.13	1.15/1.36
SocialVAE+FPC* [34]	ECCV2022	1.37 / 1.61	1.02/1.09	1.12/1.31	1.07/1.20	1.04/1.17	1.12/1.27
Ours (TUTR)	-	**1.21/1.41**	**0.8/0.86**	**0.99/1.19**	**1.03/1.19**	**0.73/0.85**	**0.95/1.10**

Table 3. Comparisons on ETH-UCY in brier-ADE/brier-FDE. * represents the conducted model variant.

glect probabilities, we select three methods with different multimodal trajectory prediction strategies to make a comparison in brier-ADE/FDE. SIT [26] provides probability information without post-processing. CAGN [5] uses the Gaussian Mixture Model (GMM) to model diverse future trajectories but without probability information. SocialVAE+FPC [34] is the previous state-of-the-art method with post-processing but without probability information. We conduct two variants of VAGN and SocialVAE+FPC to make a comparison with TUTR. CAGN predicts 20 Gaussian components and the weights of each component as probabilities. SocialVAE+FPC predicts abundant trajectories and clusters them into a GMM, where the weights of each component are considered as probabilities.

As shown in Table 3 and Table 4, TUTR achieves state-of-the-art performance in both brier-ADE and brier-FDE. Specifically, TUTR reduces the brier-ADE/brier-FDE from 1.12/1.17 to 0.95/1.1 on ETH-UCY compared to SocialVAE+FPC [34]. On SDD, TUTR reduces brier-ADE/brier-FDE from 9.57/14.75 to 8.44/13.53 compared to SocialVAE+FPC.

Comparison in Inference Speed. We compare the inference speed with previous state-of-the-art methods in sparse and dense pedestrian motion scenes, respectively. We set the number of pedestrians N as equal to $5, 10, 20, 40$, and 80, respectively. The larger N represents more dense scenes. As shown in Table 5, TUTR significantly outperforms the methods (MemoNet [33], SocialVAE+FPC [34]) with post-processing step significantly. Specifically, MemoNet and SocialVAE+FPC suffer from the higher prediction delays that they cost $1.2989s$ and $2.0939s$ to predict a $4.8s$ trajectory in a dense scene, respectively. In contrast, TUTR achieves about $10\times$ speed improvement in sparse scenes and $40\times$ speed improvement in dense scenes. The inference speed variance is also shown in Figure 1 (B). In conclusion, TUTR achieves a balance between accuracy performance and inference speed.

4.3. Ablation Studies

Importance of Global Prediction. We conduct a variant to evaluate the importance of global prediction, referring to object queries [3]. Specifically, the L general motion modes obtained from the global prediction are replaced by L learnable latent vectors. In this case, the nearest neighbor hypothesis is not available because the latent vectors cannot provide information on which latent vector is closest to the ground truth. Therefore, we use a variety loss [6] to predict trajectories. The experimental results demonstrate that the latent vectors suffer from a large performance reduction, enlarging the average ADE/FDE from 0.21/0.36 to 0.34/0.64 on ETH-UCY and from 7.76/12.69 to 17.26/34.64 on SDD. The reason lies that latent vectors cannot provide useful information to guide neural networks to generate diverse trajectories compared with general motion modes.

Number of General Motion Modes. The general motion modes are used to represent the common motion behaviors of a pedestrian. Here. we analyze the impact of the number of general motion modes L as shown in Figure 4, where the experimental results show that $L = 100$ achieves the best performance. The reason could be that too few general motion modes can not cover common motion behaviors, and too many general motion modes disturb the neural network to search for effective modes.

Importance of Model Components. We conduct three variants to evaluate the components of TUTR. As shown in Table 6, GP is the global prediction, MTE is the mode-level transformer encoder, and STD is the social-level transformer decoder. GP is replaced by multiple learnable latent vectors similar to the before-mentioned ablation study of global prediction. The MTE is replaced by a feed-forward network [29] to perform an ablation study. The experimental results show that each component is effective in predicting diverse future trajectories.

4.4. Qualitative Analysis

General Motion Modes. Here, we provide an intuitive visualization of general motion modes to evaluate their ability to cover common motion behaviors of a pedestrian. Note that the general motion modes are obtained on normalized trajectories, *i.e.*, the direction of pedestrian motion is from right to left. As shown in Figure 3, the general motion modes could represent the common motion behaviors, *e.g.*, going straight, turning left/right, or turning back.

Predicted Diverse Trajectories. As shown in Figure 5, the predicted trajectories have a good diversity to cover various motion behaviors of pedestrians, such as turning left/right (1,4), going straight (3), keeping standing (6) and sharp turning (2, 5). Moreover, TUTR can predict the best

Method	Venue/Year	ADE/FDE
CAGN* [5]	AAAI2022	17.82/35.78
SIT [26]	AAAI2022	10.06/16.33
SocialVAE+FPC* [34]	ECCV2022	9.57/14.75
Ours (TUTR)	-	**8.44/13.53**

Table 4. Comparisons on SDD in brier-ADE/brier-FDE. * represents the methods without probability prediction.

N	MemoNet [33]	SocialVAE+FPC [34]	Ours
5	0.3221	0.6127	**0.0577**
10	0.4058	0.6869	**0.0561**
20	0.5358	1.1807	**0.0586**
40	0.7784	1.4053	**0.0582**
80	1.2989	2.0939	**0.0533**

Table 5. Comparisons in inference time recorded by seconds. Our method significantly outperforms the compared methods.

Variant	GP	MTE	STD	ADE/FDE
(1)	✗	✓	✓	17.26/34.64
(2)	✓	✗	✗	8.14/13.46
(3)	✓	✓	✗	7.85/12.91
(4)	✓	✓	✓	**7.76/12.69**

Table 6. Ablation study of TUTR on SDD dataset in ADE/FDE.

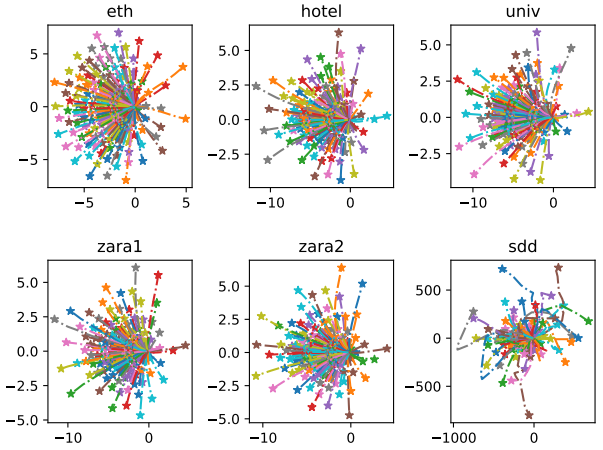

Figure 3. Visualizations of the general motion modes on ETH-UCY and SDD. The motion direction is from right to left.

trajectory with high probability.

5. Conclusion

In this paper, we present a trajectory-unified framework named TUTR, which unifies the social interaction and multimodal trajectory prediction into an encoder-decoder transformer architecture to remove the need for post-processing. The experimental results show that TUTR achieves competitive accuracy performance compared with previous state-of-the-art methods that gain from the well-designed post-processing. What's more, TUTR performs about $10\times - 40\times$ inference speed improvements. However, the clustering algorithm is hard to match complex data structures, such as map information. How to learn more robust mode representations is worth exploring in the future.

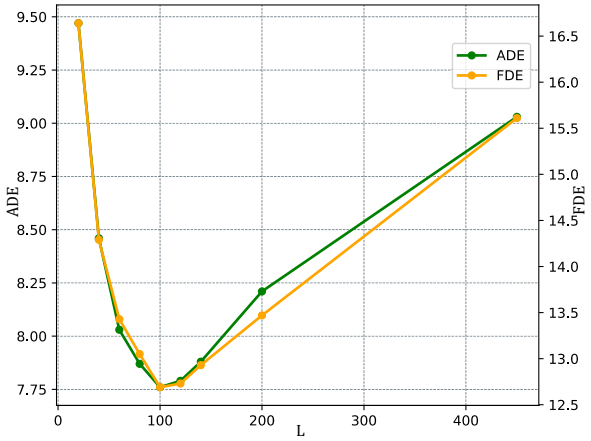

Figure 4. Ablation study of the number of general motion modes L on SDD dataset. $L = 100$ is the best performance.

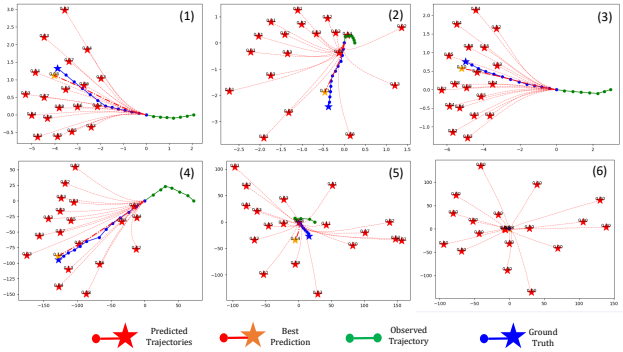

Figure 5. Visualization of predicted trajectories and corresponding probabilities.

Acknowledgement

This work was supported partly by the National Key R&D Program of China under Grant 2021YFB1714700, NSFC under Grants 62088102 and 62106192, Natural Science Foundation of Shaanxi Province under Grant 2022JC-41, and Fundamental Research Funds for the Central Universities under Grant XTR042021005.

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 *CVPR*, pages 961–971, 2016. 2

[2] Zhe Cao, Hang Gao, Karttikeya Mangalam, Qi-Zhi Cai, Minh Vo, and Jitendra Malik. Long-term human motion prediction with scene context. In *ECCV*, pages 387–404. Springer, 2020. 2

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, pages 213–229, 2020. 3, 7

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 3

[5] Jinghai Duan, Le Wang, Chengjiang Long, Sanping Zhou, Fang Zheng, Liushuai Shi, and Gang Hua. Complementary attention gated network for pedestrian trajectory prediction. In *AAAI*, pages 542–550, 2022. 1, 2, 6, 7, 8

[6] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social GAN: Socially acceptable trajectories with generative adversarial networks. In *CVPR*, pages 2255–2264, 2018. 1, 2, 6, 7

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 4

[8] Dirk Helbing and Peter Molnar. Social force model for pedestrian dynamics. *Physical Review E*, 51(5):4282, 1995. 2

[9] Jan Hosang, Rodrigo Benenson, and Bernt Schiele. Learning non-maximum suppression. In *CVPR*, pages 4507–4515, 2017. 1

[10] Rafal Jozefowicz, Wojciech Zaremba, and Ilya Sutskever. An empirical exploration of recurrent network architectures. In *ICML*, pages 2342–2350, 2015. 3

[11] Ioannis Karamouzas, Brian Skinner, and Stephen J Guy. Universal power law governing pedestrian interactions. *Physical Review Letters*, 113(23):238701, 2014. 2

[12] Kihwan Kim, Dongryeol Lee, and Irfan Essa. Gaussian process regression flow for analysis of motion trajectories. In *ICCV*, pages 1164–1171, 2011. 2

[13] Florin Leon and Marius Gavrilescu. A review of tracking, prediction and decision making methods for autonomous driving. *arXiv preprint arXiv:1909.07707*, 2019. 1

[14] Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski. Crowds by example. *Computer Graphics Forum*, 26(3):655–664, 2007. 2, 5

[15] Karttikeya Mangalam, Yang An, Harshayu Girase, and Jitendra Malik. From goals, waypoints & paths to long term human trajectory forecasting. In *ICCV*, pages 15233–15242, 2021. 2

[16] 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 *ECCV*, pages 759–776, 2020. 1, 2, 5, 6

[17] Abduallah Mohamed, Kun Qian, Mohamed Elhoseiny, and Christian Claudel. Social-STGCNN: A social spatio-temporal graph convolutional neural network for human trajectory prediction. In *CVPR*, pages 14424–14432, 2020. 1, 2

[18] Sajjad Mozaffari, Omar Y Al-Jarrah, Mehrdad Dianati, Paul Jennings, and Alexandros Mouzakitis. Deep learning-based vehicle behavior prediction for autonomous driving applications: A review. *IEEE T-ITS*, 23(1):33–47, 2020. 1

[19] Stefano Pellegrini, Andreas Ess, Konrad Schindler, and Luc Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *ICCV*, pages 261–268, 2009. 2, 5

[20] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018. 3

[21] Amir Rasouli, Mohsen Rohani, and Jun Luo. Bifold and semantic reasoning for pedestrian behavior prediction. In *ICCV*, pages 15600–15610, 2021. 2

[22] Alexandre Robicquet, Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Learning Social Etiquette: Human trajectory understanding in crowded scenes. In *ECCV*, pages 549–565, 2016. 2, 5

[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 *CVPR*, pages 1349–1358, 2019. 2, 6

[24] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *ECCV*, pages 683–700, 2020. 2

[25] Nasim Shafiee, Taskin Padir, and Ehsan Elhamifar. Introvert: Human trajectory prediction via conditional 3d attention. In *CVPR*, pages 16815–16825, 2021. 2

[26] Liushuai Shi, Le Wang, Chengjiang Long, Sanping Zhou, Fang Zheng, Nanning Zheng, and Gang Hua. Social interpretable tree for pedestrian trajectory prediction. In *AAAI*, pages 2235–2243, 2022. 1, 2, 6, 7, 8

[27] 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 *CVPR*, pages 8990–8999, 2021. 1, 2, 6

[28] Jur Van Den Berg, Stephen J Guy, Ming Lin, and Dinesh Manocha. Reciprocal n-body collision avoidance. In *ISRR*, pages 3–19, 2011. 2

[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, pages 5998–6008, 2017. 3, 4, 7

[30] Chuhua Wang, Yuchen Wang, Mingze Xu, and David J Crandall. Stepwise goal-driven networks for trajectory prediction. *IEEE RA-L*, 7(2):2716–2723, 2022. 2

[31] Jack M Wang, David J Fleet, and Aaron Hertzmann. Gaussian process dynamical models for human motion. *IEEE T-PAMI*, 30(2):283–298, 2007. 2

[32] Mingkun Wang, Xinge Zhu, Changqian Yu, Wei Li, Yuexin Ma, Ruochun Jin, Xiaoguang Ren, Dongchun Ren, Mingxu Wang, and Wenjing Yang. GANet: Goal area network for motion forecasting. In *ICRA*, pages 1609–1615, 2023. 5

[33] Chenxin Xu, Weibo Mao, Wenjun Zhang, and Siheng Chen. Remember intentions: Retrospective-memory-based trajectory prediction. In *CVPR*, pages 6488–6497, 2022. 1, 2, 5, 6, 7, 8

[34] Pei Xu, Jean-Bernard Hayet, and Ioannis Karamouzas. SocialVAE: Human trajectory prediction using timewise latents. In *ECCV*, pages 511–528, 2022. 1, 2, 5, 6, 7, 8

[35] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. XLNet: Generalized autoregressive pretraining for language understanding. In *NeurIPS*, pages 5753–5763, 2019. 3

[36] Cunjun Yu, Xiao Ma, Jiawei Ren, Haiyu Zhao, and Shuai Yi. Spatio-temporal graph transformer networks for pedestrian trajectory prediction. In *ECCV*, pages 507–523, 2020. 2, 3, 6

[37] Ye Yuan, Xinshuo Weng, Yanglan Ou, and Kris Kitani. AgentFormer: Agent-aware transformers for socio-temporal multi-agent forecasting. In *ICCV*, pages 9793–9803, 2021. 1, 2, 3, 5, 6

[38] Jiangbei Yue, Dinesh Manocha, and He Wang. Human trajectory prediction via neural social physics. In *ECCV*, pages 376–394, 2022. 2