Spatio-Temporal Graph Transformer Networks for Pedestrian Trajectory Prediction

Cunjun Yu[1], Xiao Ma[1,2](✉), Jiawei Ren[1], Haiyu Zhao[1], and Shuai Yi[1]

[1] SenseTime Research, Beijing, China
yucunjun@sensetime.com
[2] National University of Singapore, Singapore, Singapore
xiao-ma@comp.nus.edu.sg

Abstract. Understanding crowd motion dynamics is critical to real-world applications, e.g., surveillance systems and autonomous driving. This is challenging because it requires effectively modeling the socially aware crowd spatial interaction and complex temporal dependencies. We believe attention is the most important factor for trajectory prediction. In this paper, we present *STAR*, a Spatio-Temporal grAph tRansformer framework, which tackles trajectory prediction by only attention mechanisms. STAR models intra-graph crowd interaction by *TGConv*, a novel Transformer-based graph convolution mechanism. The inter-graph temporal dependencies are modeled by separate temporal Transformers. STAR captures complex spatio-temporal interactions by interleaving between spatial and temporal Transformers. To calibrate the temporal prediction for the long-lasting effect of disappeared pedestrians, we introduce a read-writable external memory module, consistently being updated by the temporal Transformer. We show that with only attention mechanism, STAR achieves the state-of-the-art performance on 5 commonly used real-world pedestrian prediction datasets (code available at https://github.com/Majiker/STAR).

Keywords: Trajectory prediction · Transformer · Graph neural networks

1 Introduction

Crowd trajectory prediction is of fundamental importance to both the computer vision [1,16,21,22,52] and robotics [33,34] community. This task is challenging because 1) human-human interactions are multi-modal and extremely hard to

C. Yu and X. Ma—Equal contribution, listed in alphabetical order.

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-58610-2_30) contains supplementary material, which is available to authorized users.

© Springer Nature Switzerland AG 2020
A. Vedaldi et al. (Eds.): ECCV 2020, LNCS 12357, pp. 507–523, 2020.
https://doi.org/10.1007/978-3-030-58610-2_30

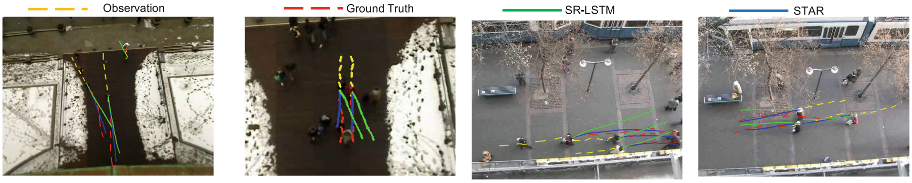
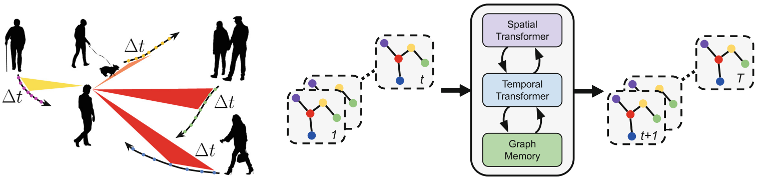


Fig. 1. STAR successfully models spatio-temporal crowd dynamics with only a strong Transformer-based attention mechanism. STAR produces more accurate prediction trajectories compared to the state-of-the-art model, SR-LSTM.



(a) Crowd Motion Modeling (b) STAR Overview

Fig. 2. (a) People decide their future motions by paying different attentions (light yellow for less attention and dark red for more attention) to the potential future motions of their neighbors up to a certain time interval (Δt). (b) STAR models the crowd as a graph and learns spatio-temporal interaction of the crowd motion by interleaving between a graph-based spatial Transformer and a temporal Transformer. An external read-writable graph memory module is applied to improve the smoothness of the temporal predictions.

capture, e.g., strangers would avoid intimate contact with others, while fellows tend to walk in group [52]; 2) the complex temporal prediction is coupled with the spatial human-human interaction, e.g., humans condition their motions on the history and future motion of their neighbors [21].

Classic models capture human-human interaction by handcrafted energy-functions [18,19,34], which require significant feature engineering effort and normally fail to build crowd interactions in crowded spaces [21]. With the recent advances in deep neural networks, Recurrent Neural Networks (RNNs) have been extensively applied to trajectory prediction and demonstrated promising performance [1,16,21,22,52]. RNN-based methods capture pedestrian motion by their latent state and model the human-human interaction by merging latent states of spatially proximal pedestrians. Social-pooling [1,16] treat pedestrians in a neighborhood area equally and merge their latent state by a pooling mechanism. Attention mechanisms [21,22,52] relax this assumption and weigh pedestrians according to a learned function, which encodes unequal importance of neighboring pedestrians for trajectory prediction. However, existing predictors have two shared limitations: 1) the attention mechanisms used are still simple, which fails to fully model the human-human interaction, 2) RNNs normally have difficulty modeling complex temporal dependencies [43] (Fig. 1).

Recently, Transformer networks have made ground-breaking progress in Natural Language Processing domains (NLP) [10,26,43,49,51]. Transformers discard the sequential nature of language sequences and model temporal dependencies with only the powerful self-attention mechanism. The major benefit of Transformer architecture is that self-attention significantly improves temporal modeling, especially for horizon sequences, compared to RNNs [43]. Nevertheless, Transformer-based models are restricted to normal data sequences and it is hard to generalize them to more structured data, e.g., graph sequences.

In this paper, we introduce the Spatio-Temporal grAph tRansformer (STAR) framework, a novel framework for spatio-temporal trajectory prediction based purely on self-attention mechanism. We believe that learning the temporal, spatial and temporal-spatial attentions is the key to accurate crowd trajectory prediction, and Transformers provide a neat and efficient solution to this task. STAR captures the human-human interaction with a novel spatial graph Transformer. In particular, we introduce *TGConv*, a Transformer-based graph convolution mechanism. TGConv improves the attention-based graph convolution [44] by self-attention mechanism with Transformers and can capture more complex social interactions. Specifically, TGConv tends to improve more on datasets with higher pedestrian densities (ZARA1, ZARA2, UNIV). We model pedestrian motions with separate temporal Transformers, which better captures temporal dependencies compared to RNNs. STAR extracts spatio-temporal interaction among pedestrians by interleaving between spatial Transformer and temporal Transformer, a simple yet effective strategy. Besides, as Transformers treat a sequence as a bag of words, they normally have problem modeling time series data where strong temporal consistency is enforced [29]. We introduce an additional read-writable graph memory module that continuously performs smoothing over the embeddings during prediction. An overview of STAR is given by Fig. 2(b)

We experimented on 5 commonly used real-world pedestrian trajectory prediction datasets. With only attention mechanism, STAR achieves the state-of-the-art on all 5 datasets. We conduct extensive ablation studies to better understand each proposed component.

2 Background

2.1 Self-Attention and Transformer Networks

Transformer networks have achieved great success in the NLP domain, such as machine translation, sentiment analysis, and text generation [10]. Transformer networks follow the famous encoder-decoder structure widely used in the RNN seq2seq models [3,6].

The core idea of Transformer is to replace the recurrence completely by multi-head self-attention mechanism. For embeddings $\{h_t\}_{t=1}^T$, the self-attention of Transformers first learns the query matrix $Q = f_Q(\{h_t\}_{t=1}^T)$, key matrix $K = f_K(\{h_t\}_{t=1}^T)$ and a corresponding value matrix $V = f_V(\{h_t\}_{t=1}^T)$ of all embeddings from $t = 1$ to T. It computes the attention by

$$Att(Q, K, V) = \frac{\mathrm{Softmax}(QK^{\mathrm{T}})}{\sqrt{d_k}} V \tag{1}$$

where d_k is the dimension of each query. The $1/\sqrt{d_k}$ implements the scaled-dot product term for numerical stability for attentions. By computing the self-attention between embeddings across different time steps, the self-attention mechanism is able to learn temporal dependencies over long time horizon, in contrast to RNNs that remember the history with a single vector with limited memory. Besides, decoupling attention into the query, key and value tuples allows the self-attention mechanism to capture more complex temporal dependencies.

Multi-head attention mechanism learns to combine multiple hypotheses when computing attentions. It allows the model to jointly attend to information from different representations at different positions. With k heads, we have

$$\begin{aligned} \mathrm{MultiHead}(Q, K, V) &= f_O([\mathrm{head}_i]_{i=1}^{k}) \\ \text{where } \mathrm{head}_i &= Att_i(Q, K, V) \end{aligned} \tag{2}$$

where f_O is a fully connected layer merging the output from k heads and $Att_i(Q, K, V)$ denote the self-attention of the i-th head. Additional positional encoding is used to add positional information to the Transformer embeddings. Finally, Transformer outputs the updated embeddings by a fully connected layer with two skip connections.

However, one major limitation of current Transformer-based models is they only apply to non-structured data sequences, e.g., word sequences. STAR extends Transformers to more structured data sequences, as a first step, graph sequences, and apply it to trajectory prediction.

2.2 Related Works

Graph Neural Networks. Graph Neural Networks (GNNs) are powerful deep learning architectures for graph-structured data. Graph convolutions [9,15,24,27,47] have demonstrated significant improvement on graph machine learning tasks, e.g., modeling physical systems [4,28], drug prediction [31] and social recommendation systems [11]. In particular, Graph Attention Networks (GAT) [44] implement efficient weighted message passing between nodes and achieved state-of-the-art results across multiple domains. From the sequence prediction perspective, temporal graph RNNs allow learning spatio-temporal relationship in graph sequences [8,17]. Our STAR improves GAT with TGConv, a transformer boosted attention mechanism and tackles the graph spatio-temporal modeling with transformer architecture.

Sequence Prediction. RNNs and its variants, e.g., LSTM [20] and GRU [7], have achieved great success in sequence prediction tasks, e.g., speech recognition [39,46], robot localization [14,36], robot decision making [23,37], and etc.

RNNs have been also successfully applied to model the temporal motion pattern of pedestrians [1,16,21,22,52]. RNNs-based predictors make predictions with a Seq2Seq structure [41]. Additional structure, e.g., social pooling [1,16], attention mechanism [22,45,48] and graph neural networks [21,52], are used to improve the trajectory prediction with social interaction modeling.

Transformer networks have dominated Natural Language Processing domains in recent years [10,26,43,49,51]. Transformer models completely discard the recurrence and focus on the attention across time steps. This architecture allows long-term dependency modeling and large-batch parallel training. Transformer architecture has also been applied to other domains with success, e.g., stock prediction [30], robot decision making [12] etc. STAR applies the idea of Transformer to the graph sequences. We demonstrate it on a challenging crowd trajectory prediction task, where we consider crowd interaction as a graph. STAR is a general framework and could be applied to other graph sequence prediction tasks, e.g., event prediction in social networks [35] and physical system modeling [28]. We leave this for future study.

Crowd Interaction Modeling. As the pioneering work, Social Force models [19,32], has been proven effective in various applications, e.g., crowd analysis [18] and robotics [13]. They assume the pedestrians are driven by virtual forces for goal navigation and collision avoidance. Social Force models work well on interaction modeling while performing poorly on trajectory prediction [25]. Geometry based methods, e.g., ORCA [42] and PORCA [34], consider the geometry of the agent and convert the interaction modeling into an optimization problem. One major limitation of classic approaches is that they rely on hand-crafted features, which is non-trivial to tune and hard to generalize.

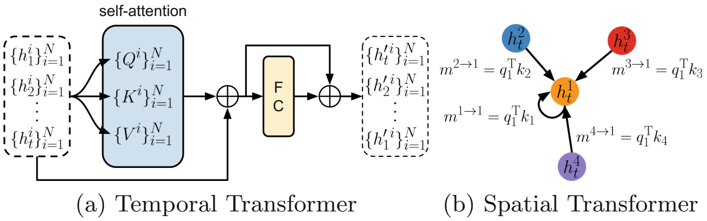
Deep learning based models achieve automatic feature engineering by directly learning the model from data. Behavior CNNs [50] capture crowd interaction by CNNs. Social-Pooling [1,16] further encodes the proximal pedestrian states by a pooling mechanism that approximates the crowd interaction. Recent works consider crowd as a graph and merge information of the spatially proximal pedestrians with attention mechanisms [22,45,48]. Attention mechanism models pedestrians with importance compared to the pooling methods. Graph neural networks are also applied to address crowd modeling [21,52]. Explicit message passing allows the network to model more complex social behaviors.

3 Method

3.1 Overview

In this section, we introduce the proposed spatio-temporal graph Transformer based trajectory prediction framework, STAR. We believe attention is the most important factor for effective and efficient trajectory prediction.

STAR decomposes the spatio-temporal attention modeling into temporal modeling and spatial modeling. For temporal modeling, STAR considers each



(a) Temporal Transformer (b) Spatial Transformer

Fig. 3. STAR has two main components, Temporal Transformer and Spatial Transformer. (a) Temporal Transformer treats each pedestrians independently and extracts the temporal dependencies by Transformer model (h is the embedding of pedestrian positions, Q, K and V are the query, key, value matrix in Transformers). (b) Spatial Transformer models the crowd as a graph, and applies TGConv, a Transformer-based message passing graph convolution, to model the social interactions ($m^{i\to j}$ is the message from node i to j represented by Transformer attention)

pedestrian independently and applies a standard temporal Transformer network to extract the temporal dependencies. The temporal Transformer provides a better temporal dependency modeling protocol compared to RNNs, which we validate in our ablation studies. For spatial modeling, we introduce *TGConv*, a Transformer-based message passing graph convolution mechanism. TGConv improves the state-of-the-art graph convolution methods with a better attention mechanism and gives a better model for complex spatial interactions. In particular, TGConv tends to improve more on datasets with higher pedestrian densities (ZARA1, ZARA2, UNIV) and complex interactions. We construct two encoder modules, each including a pair of spatial and temporal Transformers, and stack them to extract spatio-temporal interactions.

3.2 Problem Setup

We are interested in the problem of predicting future trajectories starting at time step $T_{obs} + 1$ to T of total N pedestrians involved in a scene, given the observed history during time steps 1 to T_{obs}. At each time step t, we have a set of N pedestrians $\{p_t^i\}_{i=1}^N$, where $p_t^i = (x_t^i, y_t^i)$ denotes the position of the pedestrian in a top-down view map. We assume the pedestrian pairs (p_t^i, p_t^j) with distance less than d would have an undirected edge (i, j). This leads to an *interaction graph* at each time step t: $G_t = (V_t, E_t)$, where $V_t = \{p_t^i\}_{i=1}^N$ and $E_t = \{(i, j) \mid i, j \text{ is connected at time } t\}$. For each node i at time t, we define its neighbor set as $Nb(i, t)$, where for each node $j \in Nb(i, t)$, $e_t(i, j) \in E_t$.

3.3 Temporal Transformer

The temporal Transformer block in STAR uses a set of pedestrian trajectory embeddings $\{h_1^i\}_{i=1}^N, \{h_2^i\}_{i=1}^N, \dots, \{h_t^i\}_{i=1}^N$ as input, and output a set of updated embeddings $\{h'^i_1\}_{i=1}^N, \{h'^i_2\}_{i=1}^N, \dots, \{h'^i_t\}_{i=1}^N$ with temporal dependencies as output, considering each pedestrian independently.

The structure of a temporal Transformer block is given by Fig. 3(a). The self-attention block first learns the query matrices $\{Q^i\}_{i=1}^N$, key matrix $\{K^i\}_{i=1}^N$ and the value matrix $\{V^i\}_{i=1}^N$ given the inputs. For i-th pedestrian, we have

$$Q^i = f_Q(\{h_j^i\}_{j=1}^t), \quad K^i = f_K(\{h_j^i\}_{j=1}^t), \quad V^i = f_V(\{h_j^i\}_{j=1}^t) \tag{3}$$

where f_Q, f_K and f_V are the corresponding query, key and value functions shared by pedestrians $i = 1, \ldots, N$. We could parallel the computation for all pedestrians, benefiting from the GPU acceleration.

We compute the attention for each single pedestrian separately, following Eq. 1. Similarly, we have the multi-head attention (k heads) for pedestrian i represented as

$$Att(Q^i, K^i, V^i) = \frac{\text{Softmax}(Q^i K^{i\mathrm{T}})}{\sqrt{d_k}} V^i \tag{4}$$

$$\text{MultiHead}(Q^i, K^i, V^i) = f_O([head_j]_{j=1}^k) \tag{5}$$

$$\text{where head}_j = Att_j(Q^i, K^i, V^i) \tag{6}$$

where f_O is a fully connected layer that merges the k heads and Att_j indexes the j-th head. The final embedding is generated by two skip connections and a final fully connected layers, as shown in Fig. 3(a).

The temporal Transformer is a simple generalization of Transformer networks to a data sequence set. We demonstrate in our experiment that Transformer based architecture provides better temporal modeling.

3.4 Spatial Transformer

The spatial Transformer block extracts the spatial interaction among pedestrians. We propose a novel Transformer based graph convolution, TGConv, for message passing on a graph.

Our key observation is that the self-attention mechanism can be regarded as message passing on an undirected fully connected graph. For a feature vector h_i of feature set $\{h_i\}_{i=1}^n$, we can represent its corresponding query vector as $q_i = f_Q(h_i)$, key vector as $k_i = f_K(h_i)$ and value vector as $v_i = f_V(h_i)$. We define the message from node j to i in the fully connected graph as

$$m^{j \to i} = q_i^{\mathrm{T}} k_j \tag{7}$$

and the attention function (Eq. 1) can be rewritten as

$$Att(Q, K, V) = \frac{\text{Softmax}\left(\left[m^{j \to i}\right]_{i,j=1:n}\right)}{\sqrt{d_k}} [v_i]_{i=1}^n \tag{8}$$

Built upon the above insight, we introduce *Transformer-based Graph Convolution (TGConv).* TGConv is essentially an attention-based graph convolution mechanism, similar to GATConv [44], but with a better attention mechanism powered by Transformers. For an arbitrary graph $G = (V, E)$ where

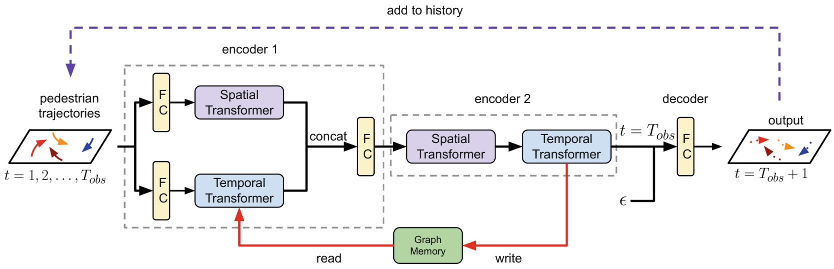


Fig. 4. Network structure of STAR with application to trajectory prediction. In STAR, trajectory prediction is achieved completely by attention mechanisms. STAR interleaves spatial Transformer and temporal Transformer in two encoder blocks to extract spatio-temporal pedestrian dependencies. An external read-writable graph memory module helps to smooth the graph embeddings and improve the consistency of temporal predictions. The prediction at $T_{obs} + 1$ is added back to history to predict the pedestrian poses at $T_{obs} + 2$.

$V = \{1, 2, \ldots, n\}$ is the node set and $E = \{(i, j) \mid i, j \text{ is connected}\}$. Assume each node i is associated with an embedding h_i and a neighbor set $Nb(i)$. The graph convolution operation for node i is written as

$$Att(i) = \frac{\text{Softmax}\left(\left[m^{j \to i}\right]_{j \in Nb(i) \bigcup \{i\}}\right)}{\sqrt{d_k}} [v_j]^{\mathrm{T}}_{j \in Nb(i) \bigcup \{i\}} + h_i \tag{9}$$

$$h'_i = f_{out}(Att(i)) + Att(i) \tag{10}$$

where f_{out} is the output function, in our case, a fully connected layer, and h'_i is the updated embedding of node i by TGConv. We summarize the TGConv function for node i by $TGConv(h_i)$. In a Transformer structure, we would normally apply layer normalization [2] after each skip connection in the above equations. We ignored them in the equations for a clean notation.

The spatial Transformer, as shown in Fig. 3(b), can be easily implemented by the TGConv. A TGConv with shared weights is applied to each graph G_t separately. We believe TGConv is general and can be applied to other tasks and we leave it for future study.

3.5 Spatio-Temporal Graph Transformer

In this section, we introduce the Spatio-Temporal grAph tRansformer (STAR) framework for pedestrian trajectory prediction.

Temporal transformer can model the motion dynamics of each pedestrian separately, but fails to incorporate spatial interactions; spatial Transformer tackles crowd interaction with TGConv but can be hard to generalize to temporal sequences. One major challenge of pedestrian prediction is modeling coupled spatio-temporal interaction. The spatial and temporal dynamics of a pedestrian is tightly dependent on each other. For example, when one decides her next action, one would first predict the future motions of her neighbors, and choose an action that avoids collision with others in a time interval Δt.

STAR addresses the coupled spatio-temporal modeling by interleaving the spatial and temporal Transformers in a single framework. Figure 4 shows the network structure of STAR. STAR has two encoder modules and a simple decoder module. The input to the network is the pedestrian position sequences from $t = 1$ to $t = T_{obs}$, where the pedestrian positions at time step t is denoted by $\{p_t^i\}_{i=1}^N$ with $p_t^i = (x_t^i, y_t^i)$. In the first encoder, we embed the positions by two separate fully connected layers and pass the embeddings to spatial Transformer and Temporal Transformer, to extract independent spatial and temporal information from the pedestrian history. The spatial and temporal features are then merged by a fully connected layer, which gives a set of new features with spatio-temporal encodings. To further model spatio-temporal interaction in the feature space, we perform post-processing of the features with the second encoder module. In encoder 2, spatial Transformer models spatial interaction with temporal information; the temporal Transformer enhances the output spatial embeddings with temporal attentions. STAR predicts the pedestrians positions at $t = T_{obs}+1$ using a simple fully connected layer with the $t = T_{obs}$ embeddings from the second temporal Transformer as input, concatenated with a random Gaussian noise to generate various future predictions [21]. We construct $G_{T_{obs}+1}$ by connecting the nodes with distance smaller than d according to the predicted positions. The prediction is added to the history for the next step prediction.

The STAR architecture significantly improves the spatio-temporal modeling ability compared to naively combining spatial and temporal Transformers.

3.6 External Graph Memory

Although Transformer networks improve long-horizon sequence modeling by self-attention mechanism, it would potentially have difficulties handling continuous time-series data which requires a strong temporal consistency [29]. Temporal consistency, however, is a strict requirement for trajectory prediction, because pedestrian positions normally would not change sharply during a short period.

We introduce a simple external *graph memory* to tackle this dilemma. A graph memory $M_{1:T}$ is read-writable and learnable, where $M_t(i)$ has the same size with h_t^i and memorizes the embeddings of pedestrian i. At time step t, in encoder 1, the temporal Transformer first reads from memory M the past graph embeddings with function $\{\tilde{h}_1^i, \tilde{h}_2^i, \ldots, \tilde{h}_{t-1}^i\}_{i=1}^N = f_{read}(M)$ and concatenate it with the current graph embedding $\{h_t^i\}_{i=1}^N$. This allows the Temporal Transformers to condition current embeddings on the previous embedding for a consistent prediction. In encoder 2, we write the output $\{{h'}_1^i, {h'}_2^i, \ldots, {h'}_t^i\}_{i=1}^N$ of Temporal Transformer to the graph memory by function $M' = f_{write}(\{{h'}_1^i, {h'}_2^i, \ldots, {h'}_t^i\}_{i=1}^N, M)$, which performs a smoothing over the time series data. For any $t' < t$, the embeddings will be updated by the information from $t'' > t$, which gives temporally smoother embeddings for a more consistent trajectory.

For implementing f_{read} and f_{write}, many potential function forms could be adopted. In this paper, we only consider a very simple strategy

$$\{\tilde{h}^i_1, \tilde{h}^i_2, \dots, \tilde{h}^i_{t-1}\}_{i=1}^N = f_{read}(M) = \{M_1(i), M_2(i), \dots, M_{t-1}(i)\}_{i=1}^N \tag{11}$$

$$M' = f_{write}(\{h'^i_1, h'^i_2, \dots, h'^i_t\}_{i=1}^N, M) = \{h'^i_1, h'^i_2, \dots, h'^i_t\}_{i=1}^N \tag{12}$$

that is, we directly replace the memory with the embeddings and copy the memory to generate the output. This simple strategy works well in practice. More complicated functional form of f_{read} and f_{write} could be considered, e.g., fully connected layers or RNNs. We leave this for future study.

4 Experiments

In this section, we first report our results on five pedestrian trajectory datasets which serve as the major benchmark for the task of trajectory prediction: ETH (ETH and HOTEL) and UCY (ZARA1, ZARA2, and UNIV) datasets. We compare STAR to 9 trajectory predictors, including the SOTA model, SR-LSTM [52]. We follow the leave-one-out cross-validation evaluation strategy which is commonly adopted by previous works. We also perform extensive ablation studies to understand the effect of each proposed component and try to provide deeper insights for model design in the trajectory prediction task.

As a brief conclusion, we show that: 1) STAR outperforms the SOTA model on 4 out of 5 datasets and have a comparable performance to the SOTA model on the other dataset; 2) the spatial Transformer improves crowd interaction modeling compared to existing graph convolution methods; 3) the temporal Transformer generally improves the LSTM; 4) the graph memory gives a smoother temporal prediction and a better performance.

4.1 Experiment Setup

We follow the same data prepossessing strategy as SR-LSTM [52] for our method. The origin of all the input is shifted to the last observation frame. Random rotation is adopted for data augmentation.

- Average Displacement Error (ADE): the mean square error (MSE) overall estimated positions in the predicted trajectory and ground-truth trajectory.
- Final Displacement Error (FDE): the distance between the predicted final destination and the ground-truth final destination.

We take 8 frames (3.2s) as an sequence and 12 frames(4.8s) as the target sequence for prediction to have a fair comparison with all the existing works.

4.2 Implementation Details

Coordinates as input would be first encoded into a vector in size of 32 by a fully connected layer followed with ReLU activation. The dropout ratio at 0.1 is applied when processing the input data. All the transformer layers accept input with feature size at 32. Both spatial transformer and temporal transformer consists of encoding layers with 8 heads. We performed a hyper-parameter search over the learning rate, from 0.0001 to 0.004 with interval 0.0001 on a smaller network and choose the best-performed learning rate (0.0015) to train all the other models. As a result, we train the network using Adam optimizer with a learning rate of 0.0015 and batch size 16 for 300 epochs. Each batch contains around 256 pedestrians in different time windows indicated by an attention mask to accelerate the training and inference process.

4.3 Baselines

We compare STAR with a wide range of baselines, including: 1) LR: A simple temporal linear regressor; 2) LSTM: a vanilla temporal LSTM; 3) S-LSTM [1]: each pedestrian is modeled with an LSTM, and the hidden state is pooled with neighbors at each time-step; 4) Social Attention [45]: it models the crowd as a spatio-temporal graph and uses two LSTMs to capture spatial and temporal dynamics; 5) CIDNN [48]: a modularized approach for spatio-temporal crowd trajectory prediction with LSTMs; 6) SGAN [16]: a stochastic trajectory predictor with GANs; 7) SoPhie [40]: one of the SOTA stochastic trajectory predictors with LSTMs. 8) TrafficPredict [38]: LSTM-based motion predictor for heterogeneous traffic agents. Note that TrafficPredict in [38] reports isometrically normalized results. We scale them back for a consistent comparison; 9) SR-LSTM: the SOTA trajectory predictor with motion gate and pair-wise attention to refine the hidden state encoded by LSTM to obtain social interactions.

4.4 Quantitative Results and Analyses

We compare STAR with state-of-the-art approaches as mentioned in Sect. 4.3. All the stochastic method samples 20 times and reports the best-performed sample.

The main results are presented in Table 1. We observe that STAR-D outperforms SOTA deterministic models on the overall performance, and the stochastic STAR significantly outperforms all SOTA models by a large margin.

One interesting finding is that the simple model LR significantly outperforms many deep learning approaches including the SOTA model, SR-LSTM, in the HOTEL scene, which mostly contains straight-line trajectories and is relatively less crowded. This indicates that these complex models might overfit to those complex scenes like UNIV. Another example is that STAR significantly outperforms SR-LSTM on ETH and HOTEL, but is only comparable to SR-LSTM on UNIV, where the crowd density is high. This can potentially be explained by that SR-LSTM has a well-designed gated-structure for message passing on the graph, but has a relatively weak temporal model, a single LSTM. The design of

SR-LSTM potentially improves spatial modeling but might also lead to overfitting. In contrast, our approach performs well in both simple and complex scenes. We then will further demonstrate this in Sect. 4.5 with visualized results.

Table 1. Comparison with baselines models. STAR-D denotes the deterministic version of STAR. †: The results marked with † are calculated on 20 samples since they are stochastic models. $*$: SoPhie takes extra image input.

Deterministic	Performance (ADE/FDE)					
	ETH	HOTEL	ZARA1	ZARA2	UNIV	AVERAGE
LR	1.33/2.94	0.39/0.72	0.62/1.21	0.77/1.48	0.82/1.59	0.79/1.59
LSTM	1.13/2.39	0.69/1.47	0.64/1.43	0.54/1.21	0.73/1.60	0.75/1.62
S-LSTM [1]	0.77/1.60	0.38/0.80	0.51/1.19	0.39/0.89	0.58/1.28	0.53/1.15
CIDNN [48]	1.25/2.32	1.31/1.86	0.90/1.28	0.50/1.04	**0.51/1.07**	0.89/1.73
SocialAttention [45]	1.39/2.39	2.51/2.91	1.25/2.54	1.01/2.17	0.88/1.75	1.41/2.35
TrafficPredict [38]	5.46/9.73	2.55/3.57	4.32/8.00	3.76/7.20	3.31/6.37	3.88/6.97
SR-LSTM [52]	0.63/1.25	0.37/0.74	**0.41/0.90**	0.32/**0.70**	**0.51**/1.10	0.45/0.94
STAR-D	**0.56/1.11**	**0.26/0.50**	**0.41/0.90**	**0.31**/0.71	0.52/1.15	**0.41/0.87**
Stochastic	ETH	HOTEL	ZARA1	ZARA2	UNIV	AVERAGE
SGAN† [16]	0.81/1.52	0.72/1.61	0.34/0.69	0.42/0.84	0.60/1.26	0.58/1.18
SoPhie*† [40]	0.70/1.43	0.76/1.67	0.30/0.63	0.38/0.78	0.54/1.24	0.54/1.15
STGAT† [21]	0.65/1.12	0.35/0.66	0.34/0.69	0.29/0.60	0.52/1.10	0.43/0.83
STAR†	**0.36/0.65**	**0.17/0.36**	**0.26/0.55**	**0.22/0.46**	**0.31/0.62**	**0.26/0.53**

4.5 Qualitative Results and Analyses

We present our qualitative results in Fig. 5 and Fig. 6.

- *STAR is able to predict temporally consistent trajectories.* In Fig. 5(a), STAR successfully captures the intention and velocity of the single pedestrian, where no social interaction exists.
- *STAR successfully extracts the social interaction of the crowd.* We visualize the attention values of the second spatial Transformer in Fig. 6. We notice that pedestrians are paying high attention to themselves and the neighbors who might potentially collide with them, e.g., Fig. 6(c) and (d); less attention is paid to spatially far away pedestrians and pedestrians without conflict of intentions, e.g., Fig. 6(a) and (b).
- *STAR is able to capture spatio-temporal interaction of the crowd.* In Fig. 5(b), we can see that the prediction of pedestrian considers the future motions of their neighbors. In addition, STAR better balances the spatial modeling and temporal modeling, compared to SR-LSTM. SR-LSTM potentially overfits on the spatial modeling and often tends to predict curves even when pedestrians are walking straight. This also corresponds to our findings in the quantitative

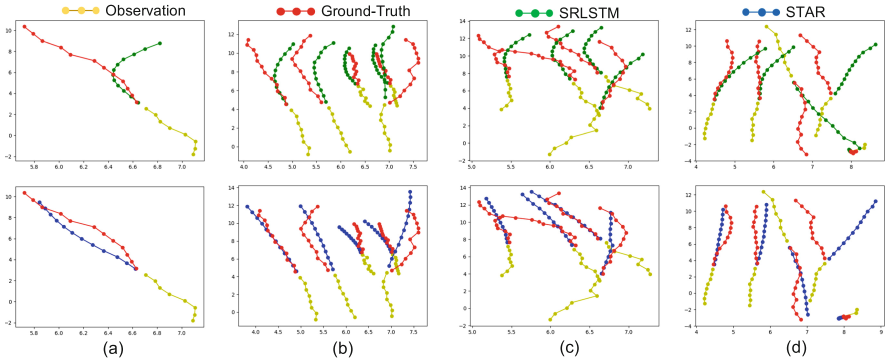


Fig. 5. Trajectory visualization. STAR successfully models the spatio-temporal interaction of the crowd and makes better predictions than the SOTA model, SR-LSTM. (a) STAR accurately extracts the temporal dynamics of the agent; (b, c, d) STAR is able to model crowd interaction and spatio-temporal interactions.

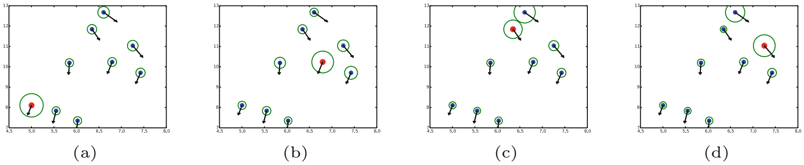


Fig. 6. Attention visualization of the spatial Transformer in encoder 2. We visualize the attention of all pedestrians with respect to the red dotted pedestrian. The size of circles represents the attention value and bigger circles indicate higher attention. STAR learns reasonable spatial attention, the pedestrians have higher attentions over themselves and their neighbors.

analyses section, that deep predictors overfits onto complex datasets. STAR better alleviates this issue with the spatial-temporal Transformer structure.

- *Auxiliary information is required for more accurate trajectory prediction.* Although STAR achieves the SOTA results, prediction can be still inaccurate occasionally, e.g., Fig. 5(d). The pedestrian takes a sharp turn, which makes it impossible to predict future trajectory purely based on the history of locations. For future work, additional information, e.g., environment setup or map, should be used to provide extra information for prediction.

4.6 Ablation Studies

We conduct extensive ablation studies on all 5 datasets to understand the influence of each STAR component. Specifically, we choose deterministic STAR to remove the influence of random sample and focus on the effect of the proposed components. The results are presented in Table 2.

- *The temporal Transformer improves the temporal modeling of pedestrian dynamics compared to RNNs.* In (4) and (5), we remove the graph mem-

Table 2. Ablation Study on SR-LSTM. We replace components in STAR with existing works. **SP** denotes spaital encoder. **TP** denotes temporal encoder. **GM** denotes Graph Memory. **GAT** denotes Graph Attention Network [44], **MHA** denotes Multi-Head Additive attention [5].**STAR** denotes components in original STAR. **VSTAR** denotes simplified STAR without encoder2.

Components				Performance (ADE/FDE)					
	SP	TP	GM	ETH	HOTEL	ZARA1	ZARA2	UNIV	AVG
(1)	GCN	STAR	✓	3.06/5.57	0.99/1.80	2.49/4.58	1.37/2.52	1.38/2.47	1.86 /3.34
(2)	GAT	STAR	✓	0.64/1.25	0.34/0.72	0.47/1.09	0.37/0.86	0.55/1.19	0.48/1.02
(3)	MHA	STAR	✓	0.58/1.15	**0.25/0.48**	0.50/0.98	0.35/0.76	0.60/1.24	0.56/0.92
(4)	STAR	LSTM	-	0.66/1.29	0.34/0.68	0.45/0.96	0.34/0.74	0.60/1.29	0.48/0.99
(5)	STAR	STAR	×	0.60/1.18	0.28/0.60	0.53/1.13	0.36/0.76	0.57/1.20	0.47/0.97
(6)	VSTAR	VSTAR	✓	0.61/1.18	0.29/0.56	0.48/1.00	0.36/0.76	0.58/1.24	0.46/0.95
(7)	STAR	STAR	✓	**0.56/1.11**	0.26/0.50	**0.41/0.90**	**0.31/0.71**	**0.52/1.15**	**0.41/0.87**

ory and fix the STAR for spatial encoding. The temporal prediction ability of these two models is only dependent on their temporal encoders, LSTM for (4) and STAR for (5). We observe that the model with temporal Transformer encoding outperforms LSTM in its overall performance, which suggests that Transformers provide a better temporal modeling ability compared to RNNs.

- *TGConv outperforms the other graph convolution methods on crowd motion modeling.* In (1), (2), (3) and (7), we change the spatial encoders and compare the spatial Transformer by TGConv (7) with the GCN [24], GATConv [44] and the multi-head additive graph convolution [5]. We observe that TGConv, under the scenario of crowd modeling, achieves higher performance gain compared to the other two alternative attention-based graph convolutions.
- *Interleaving spatial and temporal Transformer is able to better extract spatio-temporal correlations.* In (6) and (7), we observe that the two encoder structures proposed in the STAR framework (7), generally outperforms the single encoder structure (6). This empirical performance gain potentially suggests that interleaving the spatial and temporal Transformers is able to extract more complex spatio-temporal interactions of pedestrians.
- *Graph memory gives a smoother temporal embedding and improves performance.* In (5) and (7), we verify the embedding smoothing ability of the graph memory module, where (5) is the STAR variant without GM. We first noticed that graph memory improves the performance of STAR on all datasets. In addition, we noticed that on ZARA1, where the spatial interaction is simple and temporal consistency prediction is more important, graph memory improves (6) to (7) by the largest margin. According to the empirical evidence, we can conclude that the embedding smoothing of graph memory is able to improve the overall temporal modeling for STAR.

5 Conclusion

We have introduced STAR, a framework for spatio-temporal crowd trajectory prediction with only attention mechanisms. STAR consists of two encoder modules, composed of spatial Transformers and temporal Transformers. We also have introduced TGConv, a novel powerful Transformer based graph convolution mechanism. STAR, using only attention mechanisms, achieves SOTA performance on 5 commonly used datasets.

STAR makes prediction only with the past trajectories, which might fail to detect the unpredictable sharp turns. Additional information, e.g., environment configuration, could be incorporated into the framework to solve this issue.

STAR framework and TGConv are not limited to trajectory prediction. They can be applied to any graph learning task. We leave it for future study.

References

1. Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., Savarese, S.: Social LSTM: Human trajectory prediction in crowded spaces. In: CVPR (2016)
2. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. arXiv preprint arXiv:1607.06450 (2016)
3. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
4. Battaglia, P., Pascanu, R., Lai, M., Rezende, D.J., et al.: Interaction networks for learning about objects, relations and physics. In: Advances in Neural Information Processing Systems (2016)
5. Chen, B., Barzilay, R., Jaakkola, T.: Path-augmented graph transformer network (2019). https://doi.org/10.26434/chemrxiv.8214422
6. Cho, K., et al.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (2014)
7. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555 (2014)
8. Cui, Z., Henrickson, K., Ke, R., Wang, Y.: Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting. IEEE Trans. Intell. Transp. Syst. (2019)
9. Defferrard, M., Bresson, X., Vandergheynst, P.: Convolutional neural networks on graphs with fast localized spectral filtering. In: Advances in Neural Information Processing Systems (2016)
10. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
11. Fan, W., et al.: Graph neural networks for social recommendation. In: WWW (2019)
12. Fang, K., Toshev, A., Fei-Fei, L., Savarese, S.: Scene memory transformer for embodied agents in long-horizon tasks. In: CVPR (2019)
13. Ferrer, G., Garrell, A., Sanfeliu, A.: Robot companion: a social-force based approach with human awareness-navigation in crowded environments. In: IROS (2013)

14. Förster, A., Graves, A., Schmidhuber, J.: RNN-based learning of compact maps for efficient robot localization. In: ESANN (2007)
15. Gilmer, J., Schoenholz, S.S., Riley, P.F., Vinyals, O., Dahl, G.E.: Neural message passing for quantum chemistry. In: ICML (2017)
16. Gupta, A., Johnson, J., Fei-Fei, L., Savarese, S., Alahi, A.: Social Gan: socially acceptable trajectories with generative adversarial networks. In: CVPR (2018)
17. Hajiramezanali, E., Hasanzadeh, A., Narayanan, K., Duffield, N., Zhou, M., Qian, X.: Variational graph recurrent neural networks. In: Advances in Neural Information Processing Systems (2019)
18. Helbing, D., Buzna, L., Johansson, A., Werner, T.: Self-organized pedestrian crowd dynamics: experiments, simulations, and design solutions. Transp. Sci. **39**, 1–24 (2005)
19. Helbing, D., Molnar, P.: Social force model for pedestrian dynamics. Phys. Rev. E **51**, 4282 (1995)
20. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. (1997)
21. Huang, Y., Bi, H., Li, Z., Mao, T., Wang, Z.: Stgat: modeling spatial-temporal interactions for human trajectory prediction. In: ICCV (2019)
22. Ivanovic, B., Pavone, M.: The trajectron: probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs. In: ICCV (2019)
23. Karkus, P., Ma, X., Hsu, D., Kaelbling, L.P., Lee, W.S., Lozano-Pérez, T.: Differentiable algorithm networks for composable robot learning. arXiv preprint arXiv:1905.11602 (2019)
24. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
25. Kuderer, M., Kretzschmar, H., Sprunk, C., Burgard, W.: Feature-based prediction of trajectories for socially compliant navigation. In: RSS (2012)
26. Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R.: Albert: a lite bert for self-supervised learning of language representations. arXiv preprint arXiv:1909.11942 (2019)
27. Li, Y., Tarlow, D., Brockschmidt, M., Zemel, R.: Gated graph sequence neural networks. arXiv preprint arXiv:1511.05493 (2015)
28. Li, Y., Wu, J., Tedrake, R., Tenenbaum, J.B., Torralba, A.: Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids. arXiv preprint arXiv:1810.01566 (2018)
29. Lim, B., Arik, S.O., Loeff, N., Pfister, T.: Temporal fusion transformers for interpretable multi-horizon time series forecasting. arXiv preprint arXiv:1912.09363 (2019)
30. Liu, J., et al.: Transformer-based capsule network for stock movement prediction. In: Proceedings of the First Workshop on Financial Technology and Natural Language Processing (2019)
31. Liu, K., et al.: Chemi-Net: a molecular graph convolutional network for accurate drug property prediction. Int. J. Mol. Sci. **20**, 3389 (2019)
32. Löhner, R.: On the modeling of pedestrian motion. Appl. Math. Model. **34**, 366–382 (2010)
33. Luo, Y., Cai, P.: Gamma: A general agent motion prediction model for autonomous driving. arXiv preprint arXiv:1906.01566 (2019)
34. Luo, Y., Cai, P., Bera, A., Hsu, D., Lee, W.S., Manocha, D.: Porca: modeling and planning for autonomous driving among many pedestrians. IEEE Robot. Autom. Lett. **3**, 3418–3425 (2018)
35. Ma, X., Gao, X., Chen, G.: Beep: a Bayesian perspective early stage event prediction model for online social networks. In: ICDM (2017)

36. Ma, X., Karkus, P., Hsu, D., Lee, W.S.: Particle filter recurrent neural networks. arXiv preprint arXiv:1905.12885 (2019)
37. Ma, X., Karkus, P., Hsu, D., Lee, W.S., Ye, N.: Discriminative particle filter reinforcement learning for complex partial observations. arXiv preprint arXiv:2002.09884 (2020)
38. Ma, Y., Zhu, X., Zhang, S., Yang, R., Wang, W., Manocha, D.: Trafficpredict: trajectory prediction for heterogeneous traffic-agents. In: AAAI (2019)
39. Miao, Y., Gowayyed, M., Metze, F.: EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding. In: ASRU (2015)
40. Sadeghian, A., Kosaraju, V., Sadeghian, A., Hirose, N., Rezatofighi, H., Savarese, S.: Sophie: an attentive Gan for predicting paths compliant to social and physical constraints. In: CVPR (2019)
41. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in Neural Information Processing Systems (2014)
42. Van Den Berg, J., Guy, S.J., Lin, M., Manocha, D.: Reciprocal n-body collision avoidance. In: Pradalier, C., Siegwart, R., Hirzinger, G. (eds.) Robotics Research. Springer Tracts in Advanced Robotics, vol. 70, pp. 3–19. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19457-3_1
43. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems (2017)
44. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. arXiv preprint arXiv:1710.10903 (2017)
45. Vemula, A., Muelling, K., Oh, J.: Social attention: modeling attention in human crowds. In: ICRA (2018)
46. Xiong, W., Wu, L., Alleva, F., Droppo, J., Huang, X., Stolcke, A.: The Microsoft 2017 conversational speech recognition system. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (2018)
47. Xu, K., Hu, W., Leskovec, J., Jegelka, S.: How powerful are graph neural networks? arXiv preprint arXiv:1810.00826 (2018)
48. Xu, Y., Piao, Z., Gao, S.: Encoding crowd interaction with deep neural network for pedestrian trajectory prediction. In: CVPR (2018)
49. Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R.R., Le, Q.V.: Xlnet: generalized autoregressive pretraining for language understanding. In: Advances in Neural Information Processing Systems (2019)
50. Yi, S., Li, H., Wang, X.: Pedestrian behavior understanding and prediction with deep neural networks. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 263–279. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_16
51. Young, T., Hazarika, D., Poria, S., Cambria, E.: Recent trends in deep learning based natural language processing. IEEE Comput. Intel. Mag. **13**, 55–75 (2018)
52. Zhang, P., Ouyang, W., Zhang, P., Xue, J., Zheng, N.: SR-LSTM: state refinement for LSTM towards pedestrian trajectory prediction. In: CVPR (2019)