Non-Autoregressive Sparse Transformer Networks for Pedestrian Trajectory Prediction

Di Liu 1,2,†, Qiang Li 1,†, Sen Li 1, Jun Kong 1,* and Miao Qi 3,*

1 College of Information Science and Technology, Northeast Normal University, Changchun 130117, China
2 School of Computer Science, Northeast Electric Power University, Jilin City 132012, China
3 Key Laboratory of Applied Statistics of MOE, Northeast Normal University, Changchun 130024, China
* Correspondence: kongjun@nenu.edu.cn (J.K.); qim801@nenu.edu.cn (M.Q.); Tel.: +86-186-8652-5736 (J.K.);
+86-138-4306-2946 (M.Q.)
† These authors contributed equally to this work.

Abstract: Pedestrian trajectory prediction is an important task in practical applications such as automatic driving and surveillance systems. It is challenging to effectively model social interactions among pedestrians and capture temporal dependencies. Previous methods typically emphasized social interactions among pedestrians but ignored the temporal consistency of predictions and suffered from superfluous interactions by dense undirected graphs, resulting in a considerable deviance from reality. In addition, autoregressive approaches predicted future locations conditioning on previous predictions one by one, which would lead to error accumulation and time consuming. To address these issues, we present Non-autoregressive Sparse Transformer (NaST) networks for pedestrian trajectory prediction. Specifically, NaST models sparse spatial interactions and sparse temporal dependency via a sparse spatial transformer and a sparse temporal transformer separately. Different from previous predictions such as RNN-based approaches, the transformer decoder works in non-autoregressive pattern and predicts all the future locations at one time from a query sequence, which could avoid the error accumulation and be less computationally intensive. We evaluate our proposed method on the ETH and UCY datasets, and the experimental results show our method outperforms comparative state-of-the-art methods.

Keywords: pedestrian trajectory prediction; sparse transformer; non-autoregressive

1. Introduction

Pedestrian trajectory prediction plays an important role in many fields such as autonomous driving [1,2], robotic motion planning [3], video surveillance [4,5], and computer vision [6–10]. It is a challenging task because the pedestrian motion is influenced by both historical trajectories and social interaction with neighbors [9]. Crowd interactions often obey some social norms in pedestrian trajectory prediction. For example, strangers usually keep their distance from others to avoid collisions, but fellows tend to walk together [11]. The motion of pedestrians is very easy to be influenced by neighbors in a scene [7]. Moreover, the environment will also affect pedestrian trajectory such as the surrounding obstacles or events happening suddenly. Such interactions are much more complex and harder to model in learning systems.

Traditional methods adopted handcrafted energy functions [2,12,13] to model human–human interaction, and these classic approaches are not easy to tune and are hard to generalize [9]. With the development of deep neural networks, a family of methods based on Recurrent Neural Networks (RNNs) [6–10] have been applied in pedestrian trajectory prediction. RNN-based approaches capture temporal features of pedestrians by their latent state and model the spatial interaction by merging the features of neighbors nearby. The prediction of trajectory can be achieved in an iterative way by repeatedly predicting
each next location. Nevertheless, this kind of autoregressive decoding pattern is not parallelizable, which may cause error accumulation and increase computational complexity.

Distance-based methods [6,7,14] capture crowd interaction by merging their latent states with a social pooling layer, while attention-based methods [10,15–17] dynamically generate the importance of neighbors using soft attention instead. The graph-based methods [8,9,18,19] consider the pedestrians in the scene as graph nodes and capture the spatial interaction using graph neural networks such as Graph Convolution Networks (GCN) [8,19,20] or Gated Attention (GAT) [21]. It is much more effective in modeling complex social interactions in the real world. However, most of the distance-based and attention-based methods consider an agent interacting with the others in the neighborhood to construct dense interactions. In fact, an agent is influenced by only a few neighbors in a real scenario. In addition, the distance-based methods usually consider that the interaction between two agents are identical to each other and model undirected interaction. In some cases, a pedestrian might change their path to avoid collision with another person. However, the other one might go straight along the original path without any change. Therefore, the identical undirected interaction between them is unreasonable. Most existing graph-based methods extract features by simply aggregating weighted features of nodes in the local spatial neighborhood and neglecting the relative relation among pedestrians. Therefore, how to effectively encode the spatial and temporal interaction of pedestrians remains a challenging problem.

Transformer neural networks were first applied in Natural Language Processing (NLP) domains [22,23] and made great breakthroughs. They were designed for sequential tasks and were notable for powerful self-attention mechanisms for modelling long-range dependencies of data. The great success in NLP led researchers to adopt transformers to a series of computer vision tasks such as object detection [24], image classification [25], and joint vision–language modeling [26]. Existing works adopted transformers to pedestrian trajectory prediction and obtained encouraging results [27]. Nevertheless, the self-attention in [27] is dense and brings superfluous interaction. In addition, the prediction is made in an autoregressive way since the future locations are generated sequentially one at time. Generally speaking, existing works for pedestrian trajectory prediction could be mainly classified into distance-based methods [6,7,14], attention-based methods [10,15–17], and graph-based methods [8,9,18,19]. According to the analysis above, these methods have shortcomings in effectively modeling social interactions and temporal dependencies.

To solve the problems mentioned above, we delved into non-autoregressive pedestrian trajectory prediction with transformers to propose a Non-autoregressive Sparse Transformer (NaST) network. Our work is in line with recent approaches [24,27,28]. In order to capture the spatial interaction, we designed a sparse spatial transformer. Instead of making dense interactions between a particular pedestrian with all of the others in a scene, we decided to let the model discover the exact neighbors who were involved in the interaction and learned the proper weights for them. More specifically, the spatial transformer first computed weights for each agent by multi-head self-attention. Then, the attention weights were filtered by a designed sparse interaction mask to prune the irrelevant neighbors. Finally, the remaining ones were aggregated to compute the spatial interaction. In addition, the spatial interaction was directed since the influence of social interaction was not identical to each other between pedestrian pairs. Therefore, the sparse spatial transformer can learn adaptive self-attention and find the exact set of pedestrians involved in social interactions. In the same way, the pedestrian motion feature is captured by a sparse temporal transformer. In particular, the sparse spatial transformer and sparse temporal transformer are jointly combined to model the spatial social interactions and temporal motions. Furthermore, we designed the transformer in a non-autoregressive way to predict all of the future locations at one time, which could avoid error accumulation and would be less computationally intensive. Finally, our proposed NaST is evaluated on the commonly used ETH [29] and UCY [30] pedestrian trajectory prediction datasets. The experiment results indicated that our method outperformed some of the state-of-the-art methods. Compared with the
existing methods, this is the first proposal of sparse transformers and the application of non-autoregressive inference in trajectory prediction tasks.

In summary, our contributions are three-fold: (1) We propose a Non-autoregressive Sparse transformer network for pedestrian trajectory prediction tasks. The advantage of the model is demonstrated by the experiments. (2) We propose a model of social interaction between pedestrians with a sparse spatial transformer encoder and extract the temporal dependencies with a sparse temporal transformer encoder. (3) We investigate the use of non-autoregressive inference with a transformer neural network to avoid error propagation.

2. Related Works

2.1. Pedestrian Trajectory Prediction

Pedestrian trajectory prediction can be considered as a sequential prediction task that anticipates the future path of an agent by using historical trajectories. A family of methods based on RNNs have been proposed to model temporal dependencies, such as Long Short-Term Memory (LSTM) networks and Gated Recurrent Unit (GRU). Social-LSTM [6] modeled the temporal dependency and predicted the future locations with LSTM, which computed the social interaction between the specific pedestrian and neighbors within a certain distance. Study [31] combined LSTM and the soft-attention mechanism [17] to model social interactions among pedestrians. Since the current prediction is computed based on the results of the previous time steps, RNN/LSTM-based methods are often time-consuming for predictions. In addition, they inevitably accumulate errors and make the long-term predictions inaccurate. Graph is another popular choice for trajectory prediction combined with CNN or RNN/LSTM. Social-STGCNN [19] considered the pedestrians as graph nodes and the edges were weighted by the relative distance among the agents. Combined with graph attention network, STGAT [9] constructed a spatial-temporal model to predict pedestrian future trajectories. However, the number of nodes in graph depends on the number of pedestrians in a scene. Hence, the size of graph will grow significantly in the crowded scenes. Moreover, there are some other works elaborated on in this task. SGAN [7] adopted Generative Adversarial Network (GAN) [32] for multi-modal trajectory predictions. DDPTP [33] evaluated the most likely destinations of the pedestrian using the destination classifier (DC) and predicted the future trajectory with the destination-specific trajectory model (DTM). Study [34] formulated the trajectory prediction task as a reverse process of motion indeterminacy diffusion. MemoNet [35] is an instance-based approach that predicted the movement intentions of agents by looking for similar scenarios in the training data. STAR [27] constructed spatial and temporal transformers [22] to model the spatial interactions and temporal dependencies, respectively. Inspired by [27], we adopted transformers to model the trajectory sequences. Different from existing works, we constructed a sparse temporal transformer and made predictions in a non-autoregressive way.

2.2. Human-Human Interactions

The early works on crowd interaction modeling are based on Social Force [12,36], which assume that the pedestrians are driven by virtual attractive and repulsive forces for making the future trajectory. Other handcrafted methods [37,38] have been applied to crowd simulation [39,40], behavior detection [41], and pedestrian trajectory prediction [42], etc. With the popularity of deep learning in recent years, many approaches based on deep neural networks have been proposed to model social interaction. Distance-based methods [6,7,14] computed the geometric relation of the agents and pooled the features of neighbors to obtain the representation of a particular agent. Attention-based methods [10,15–17] adopted soft attention mechanism to generate different weights for neighbors. The graph-based methods [8,9,18,19] considered the pedestrians in the scene as graph nodes and modeled social interaction with an adjacency matrix. STGAT [9] and Social-biGAT [18] are models that adopted graph attention networks to model the spatial interaction among the agents. Social-STGCNN [19] constructed a spatial–temporal graph and
designed a kernel function on the adjacency matrix to extract spatial interaction. However, previous works commonly model undirected interaction with all the other pedestrians or neighbors within a fixed distance. The needless dense interaction among the agents may have brought deviations from reality. In contrast, we designed a method to capture social interactions with a sparse spatial transformer that integrates the relative distance feature into the multi-head self-attention. It is capable of discovering the salient spatial interaction and finding the exact pedestrians involved in the social interaction.

2.3. Transformer Networks

Transformer [22] was first applied in Natural Language Processing [43–45] in the place of RNNs [46,47] to model sequential data, such as text generation [48], machine translation [49], etc. The core idea of the transformer consists of a multi-head self-attention method that computes the query, key, and value with the input embeddings. Compared with RNNs using iterative manners, transformers take advantage of parallel computation and capture long-range dependency. Due to the success in NLP, transformers have also been applied to other fields such as stock prediction [50], robot decision making [51], computer vision [52,53], etc. Studies [27,54] constructed spatial and temporal transformers for pedestrian trajectory predictions and achieved remarkable results compared with traditional sequence models. Some works [17,55,56] designed transformer-based modules for pedestrian and vehicle trajectory predictions and received encouraging results. However, the self-attention computed in these methods is dense and introduces superfluous interaction. Consequently, we design a novel sparse transformer that only pays attention to the agents who need to be focused on.

2.4. Non-Autoregressive Inference

For sequence modelling tasks, most deep neural networks generate predictions one by one in an autoregressive way, such as in RNN-based models. This indicates that the prediction at each time step is achieved based on the prediction results of the previous steps. These methods are not parallelizable and increase computational complexity in the case of works in machine translation using transformers [22,57]. In addition, the autoregressive pattern inevitably accumulates errors. The original design of self-attention in transformers is parallelizable in principle, while the autoregressive decoding at the inference stage makes it hard to be parallel. In light of this issue, some works have tried to attempt parallel transformer decoding in non-autoregressive pattern [24,28,58]. Study [28] adopted fertilities to parallelize decoding with transformers in machine translation. DETR [24] employed transformers with parallel decoding in end-to-end object detection. Study [58] explored non-autoregressive inference in human motion prediction. Non-autoregressive inference is also applied in pedestrian trajectory prediction tasks. NAP [59] designed a time-agnostic context generator and a time-specific context generator for non-autoregressive prediction. STPOTR [60] predicted human poses and trajectories by non-autoregressive transformer architecture. PReTR [61] extracted features from multi-agent scenes by employing a factorized spatio-temporal attention module and solved the trajectory prediction problem as a non-autoregressive task. We were inspired by this idea and constructed transformers in a non-autoregressive pattern for pedestrian trajectory predictions.

3. Methods

3.1. Overview

The goal of trajectory prediction aims to predict the future locations of all pedestrians in a scene. In fact, the future path of an agent depends on two factors, namely, historical trajectory, i.e., the temporal interaction, and the influence from neighbors, i.e., the spatial interaction. Therefore, both spatial and temporal features are the key information that should be considered for predicting trajectories.

Our proposed Non-autoregressive Sparse Transformer (NaST) networks comprises three main components: a sparse spatial transformer encoder, sparse temporal transformer
encoder, and a non-autoregressive transformer decoder. The transformer encoder and decoder comprise feed forward networks and multi-head attention modules as the original transformer [22]. The encoder architecture is built with a sparse strategy and is in charge of modeling social interaction and temporal dependency. The decoder makes inference in a non-autoregressive pattern to generate all the predicted locations in parallel. The overall NaST architecture is shown in Figure 1.

![Figure 1. Overview architecture of the proposed Non-autoregressive Sparse Transformer (NaST) networks for pedestrian trajectory prediction.](image)

Given a set of $N$ pedestrians in a scene with their corresponding observed positions in $T_{obs}$ steps, $X^i_t = (x^i_t, y^i_t)$ represents the location of pedestrian $i \in \{1, \ldots, N\}$ at time $t \in \ldots, T_{obs}\}$. Therefore, the history observation trajectories of pedestrian $i$ can be represented as $X^i_1, \ldots, X^i_{T_{obs}}$ and the future corresponding positions that need to be predicted are $Y^i_t, t \in \{T_{obs+1}, \ldots, T_{pred}\}$.

Pedestrian future trajectory is highly related to historical trajectory and the influence of neighbors nearby. Previous works based on dense interaction model constructed superfluous interactions and thus inevitably resulted in a considerable deviance from reality. The original transformer computes dense self-attention between a pedestrian with all neighbors. However, in a real scenario, a pedestrian is not influenced by all of the nearby neighbors. To find neighbors who are involved in interactions, we designed a sparse spatial transformer to model the social interactions. Due to the consistency of the trajectory in the temporal dimension, not all the temporal nodes are necessary for modeling temporal interactions, thus the temporal transformer is designed to find the most important time steps, which are helpful for predicting the future trajectory.

In Figure 1, the input history observation sequences are first embedded by a fully connected layer and then sent to the transformer encoder. The transformer encoder comprises a sparse spatial transformer and a sparse temporal transformer, which model the spatial and temporal interaction, respectively. The output features of the sparse spatial and temporal transformers are merged by a fully connected layer to form a set of new features with spatio-temporal encodings. The transformer decoder receives the outputs of the encoder along with a query sequence. The results are then concatenated with random Gaussian noise and embedded by a fully connected layer to generate the predictions sequence $\hat{Y}^i_t, t \in \{T_{obs+1}, \ldots, T_{pred}\}$. We elaborate on our modular design for each part in the rest of the section.

3.2. Sparse Spatial Transformer

The spatial transformer encoder is composed of $L$ layers, each with a multi-head adaptive sparse self-attention module and a feed forward network. The encoder receives the embeddings of historical trajectory as input and produces a sequence of embeddings of the same dimension.

Actually, pedestrians in one frame could be formulated as a directed spatial graph $G = (V^t, E^t)$, where each node $v^i_t \in V^t, i \in \{1, \ldots, N\}$ corresponds to the $i$th pedestrian.
at time $t$, and the weighted edge $(v_i^t, v_j^t) \in E^t$ represents the potential influence from pedestrian $v_i$ to $v_j$ at time $t$.

The spatial relation of pedestrians at each time is modeled as an asymmetric edge weight matrix in this task. It reviews the unequal influence from nodes $v_i$ to $v_j$ and $v_j$ to $v_i$. Therefore, instead of constructing graphs with undirected spatial distances, we introduce relative spatial locations as prior edge feature knowledge of the edge weight matrix at time $t$ (the superscript $t$ is omitted for simplification):

$$e_{ij} = \begin{cases} 
\varphi((x_i - x_j), (y_i - y_j)), & d(v_i, v_j) < D \\
0, & d(v_i, v_j) \geq D
\end{cases} (1)$$

where $d(v_i, v_j)$ represents the distance between pedestrians $i$ and $j$, and $D$ is the threshold. $\varphi(\cdot)$ embeds the relative distance features with a linear transformation. The process of message passing is illustrated in Figure 2. The feature of node $v_i$ is represented as $h_i$ ($i = 1, \ldots, N$) at time $t$.

![Figure 2. The illustration of message passing mechanism in directed spatial graph.](image)

Then, we feed the learnable edge weights $e_{ij}$ into the spatial transformer to compute the spatial interaction together with the node embedding. Given feature $h_i$ of node $v_i$, we can represent its corresponding query vector as $Q_i = f_Q(h_i)$, its key vector as $K_i = f_K(h_i)$, and its value vector as $V_i = f_V(h_i)$. The self-attention in the original transformer [22] is computed by

$$a_{ij}^0 = \frac{Q_i K_j^T}{\sqrt{d_k}} (2)$$

where $d_k$ represents the dimensions of the query vector $Q_i$ and the key vector $K_i$. However, the attention is only computed by the features (the coordinates embedded in the trajectory prediction task) of the node itself. It neglects the interaction from other nodes. In order to consider the directed interactions between nodes, we define the message from nodes $v_j$ to $v_i$ in the directed spatial graph as

$$\hat{a}_{ij} = \frac{Q_i K_j^T}{\sqrt{d_k}} + e_{ij} (3)$$

where $d_k$ is the dimensions of query vector $Q_i$ and the key vector $K_i$. The self-attention (Equation (2)) in the original transformer can be modified as

$$a_{ij} = \frac{\exp(\hat{a}_{ij})}{\sum_{n \in N_i \cup \{j\}} \exp(\hat{a}_{in})} (4)$$

where $N_i$ is the neighbor set of the node $v_i$. In this way, $a_{ij}$ gives the importance weight of neighbor $j$ to pedestrian $i$ dynamically via the node feature itself and the spatial relation (edge feature) between the neighbors. Hence, the spatial interaction at time $t$ can be computed by Equation (4) and represented by the asymmetric attention score matrix $A_{Si}$ where its $i$–$j$-th element, and $a_{ij}$ represents the influence from node $v_j$ to node $v_i$. The $i$–th row in the attention matrix $A_{Si}$ represents the $i$–th pedestrian’s initiative relations to others. The $i$–th column in $A_{Si}$ represents the passive relation from others.
Since \( A_{SI} \) is computed at every time step independently, it does not contain any temporal dependency information of the trajectories. Hence, we stack the dense interactions \( A_{SI}^t \) from every time step, and then fuse these stacked interactions with a 1x1 convolution along the temporal channel and a nonlinear function \( \sigma(\cdot) \). After that, we receive the spatial–temporal dense interaction \( A_{SI} \in \mathbb{R}^{T_{obs} \times N \times N} \).

\( A_{SI} \) models spatial interactions between the specific pedestrian and neighbors whose distances are smaller than \( D \). Nevertheless, not all these neighbors are involved in the interaction that might impact the future trajectory. In other words, there are superfluous interactions in \( A_{SI} \). To find the exact ones who are involved, we generate a sparse interaction mask \( M_{SP} \) to exclude the irrelevant neighbors.

\[
m_{ij} = \begin{cases} 
1 & a_{ij} \geq \gamma \text{ or } i = j \\
0 & a_{ij} < \gamma 
\end{cases}
\]

where \( m_{ij} \) and \( a_{ij} \) are the \((i, j)\)th element in \( M_{SP} \) and \( A_{SI} \) at time \( t \), respectively. We omitted the superscript \( t \) for simplification. \( \gamma \in [0, 1] \) is the element-wise threshold. After we obtained the sparse spatial mask \( M_{SP} \), the sparse spatial interaction matrix \( A_{SI}^{\text{sparse}} \) could be computed as

\[
A_{SI}^{\text{sparse}} = M_{SP} \odot A_{SI}
\]

where \( \odot \) denotes element-wise multiplication. Thus, a sparse spatial–temporal attention matrix representing the sparse directed interactions is eventually obtained from the inputs. The multi-head attention mechanism is also adopted to stabilize the process. Assume \( h_i \) is the embedding of node \( v_i \), and \( N_i \) is the neighbor set associated with \( v_i \). The final output feature of node \( v_i \) is calculated as:

\[
h_i' = f \left( h_i + \sum_{j \in N \cup \{i\}} a_{ij}' \cdot V_j \right) + \left( h_i + \sum_{j \in N \cup \{i\}} a_{ij}' \cdot V_j \right) i = 1, \ldots, N
\]

Where \( a_{ij}' \) is the \((i, j)\)th element in the sparse spatial interaction matrix \( A_{SI}^{\text{sparse}} \), and \( f(\cdot) \) is a fully connected layer. Finally, we can obtain nodes representation \( h' = \{h_1', \ldots, h_N'\} \), where \( h_i' \) is the updated embedding of node \( v_i \) output by the sparse spatial transformer. It captures the relative importance for different pedestrians. Therefore, the sparse spatial transformer can learn a self-adaptive attention that finds the exact set of pedestrians involved in social interactions in the scene.

**3.3. Sparse Temporal Transformer**

Similar to sparse spatial attention, we can also obtain sparse temporal attention. Similarly, the \( n \)th pedestrian across \( T_{obs} \) time steps can be formulated as a direct temporal graph \( G_{tp} = (V^n, E^n) \), \( n = 1, \ldots, N \), each node \( v^n_t \in V^n \), \( t = 1, \ldots, T_{obs} \) corresponding the \( n \)th pedestrian at time \( t \). The weighted edges \( (v^n_{\tau}, v^n_{\tau+\tau}) \in E^n \) represent the influence from time \( t - \tau \) to \( t \) for pedestrian \( n \). We represent the temporal relation of nodes as a lower triangular matrix since the current location of the agent is often related to the past. The edge feature of for pedestrian \( n \) is initialized as Equation (8). The superscript \( n \) is omitted for simplification.

\[
\phi'_{ij} = \begin{cases} 
\varphi((x_i - x_j), (y_i - y_j)) & i > j \\
1 & i = j \\
0 & i < j 
\end{cases} 
\]

where \( \phi'_{ij} \) represents the temporal influence from time \( j \) to \( i \). \( \varphi(\cdot) \) is a linear embedding function. The process of message passing is illustrated in Figure 3, where \( h_{ij} \) is the feature of node \( v^n_t \) for pedestrian \( n \) at time \( t \), the superscript \( n \) is omitted for simplification.
The temporal interaction between time $i$ and $j$ is learned from their embedding features and the edge feature $e_{ij}$. Given the node feature $h_t(l = 1, \ldots, T_{obs})$ for pedestrian $n$, we can represent the corresponding query vector as $Q_t = f_Q(h_t)$, the key vector as $K_t = f_K(h_t)$, and the value vector as $V_t = f_V(h_t)$. We define the message from time node $v^o_i$ to $v^o_j$ ($t, j = 0, \ldots, T_{obs}$) in the directed temporal graph as

$$
\hat{\beta}_{ij} = \frac{Q_t K_t^T}{\sqrt{d_k}} + e_{ij}
$$

where $d_k$ is the dimension of the query vector $Q_t$ and the key vector $K_t$. Then, we obtain the attention $\hat{\beta}_{ij}$ by the Softmax function by the temporal transformer shown in Equation (3). Thus, we can obtain the temporal interaction matrix $A_{TI}^n$ for pedestrian $n$, where its $(t, j)$-th element $\hat{\beta}_{ij}$ represents the influence at time $j$ to $t$. The $t-th$ column of $A_{TI}^n$ represents the initiative relation from the current time to other time steps. While the $t-th$ row in $A_{TI}^n$ represents the passive relations from the other time steps to the current time. $A_{TI}^n$ is computed by the location of pedestrian $n$ at each time step, and then we stack all the temporal interaction matrices of $N$ pedestrians and, finally, obtain the dense temporal interaction matrix $A_{TI} \in \mathbb{R}^{N \times T_{obs} \times T_{obs}}$.

Similarly, we generate a sparse temporal interaction mask $M_{TP}$ to filter the irrelevant time steps.

$$
m'_{ij} = \begin{cases} 
1 & \text{if } \hat{\beta}_{ij} \geq \delta \text{ and } i \geq j \\
0 & \text{if } \hat{\beta}_{ij} < \delta \text{ or } i < j 
\end{cases}
$$

where $m'_{ij}$ and $\hat{\beta}_{ij}$ are the $(i, j)$-th element in $M_{TP}$ and $A_{TI}^n$, respectively. $\delta \in [0, 1]$ is the element-wise threshold. After the sparse temporal mask $M_{TP}$ is obtained, the sparse temporal interaction $A_{TI}^{\text{sparse}}$ of all the pedestrians can be computed as

$$
A_{TI}^{\text{sparse}} = M_{TP} \odot A_{TI}
$$

Thus, a sparse temporal attention matrix representing the sparse temporal directed interactions is eventually obtained. We also adopt the multi-head attention mechanism to calculate the temporal feature at $t$ time step for pedestrian $n$:

$$
h'_t = f \left( h_t + \sum_{j=1}^{l} \hat{\beta}_{ij}' V_j \right) + \left( h_t + \sum_{j=1}^{l} \beta_{ij}' V_j \right) \quad t = 1, \ldots, T_{obs}
$$

where $\hat{\beta}_{ij}'$ is the element in $A_{TI}^{\text{sparse}}$. The sparse temporal transformer outputs the updated feature at each time step for pedestrian $n$, and the computation could be paralleled for all pedestrians. It captures the temporal dependencies for trajectory prediction. Finally, we concatenate the outputs of the sparse spatial transformer and sparse temporal transformer together and send them to the transformer decoder.

### 3.4. Non-Autoregressive Transformer Decoder

Traditional time series prediction methods often predict future location $\hat{y}_t$ based on $\hat{y}_{t-\tau}$. This kind of autoregressive fashion is prone to the propagation of errors in future predictions and is computationally expensive in practice. As the individual steps of the decoder must run sequentially rather than in parallel, autoregressive decoding prevents architectures such as the transformer from fully realizing their train–time performance.

![Figure 3. The illustration of message passing mechanism of each pedestrian in direct temporal graph.](image-url)
advantage during inference. An appropriate solution is to adopt a non-autoregressive inference pattern in the encoder–decoder model. Hence, we address these limitations by modelling the problem in a non-autoregressive pattern, as described in the following.

As illustrated in Figure 1, the transformer decoder comprises \( L \) layers, which receives the output of the sparse transformer encoder and a query sequence to produce the output embedding of the predictions. Similar to the transformer encoder, the decoder stacks are composed entirely of feed-forward networks (MLPs) and multi-head attention modules. Since no RNNs are used, there is no inherent requirement for sequential execution, making non-autoregressive decoding possible. Before decoding starts, the transformer decoder needs to know how many time steps the predicted trajectory will take in order to generate all the locations in parallel. As discussed in the machine translation work [28], the non-autoregressive decoding shows less conditional dependency between predicted elements \( \hat{y}_t \). Therefore, the input of the transformer decoder should contain as much as temporal dependency. The output of the transformer encoder contains spatial and temporal features, which are extracted by the sparse spatial transformer and sparse temporal transformer separately. It provides a reliable dependency on which to make predictions, especially to the relation of the locations at different times.

Additionally, given the observed sequence \( X_{1}, \ldots, X_{T_{\text{obs}}} \), the last observation \( X_{T_{\text{obs}}} \) is the most relevant to the following next time steps. Inspired by [28], we simply copy the input embedding of the transformer encoder at the last observed time step and fill the query sequence with it. Then, we can obtain the query sequence \( q_{T_{\text{obs}+1}}, \ldots, q_{T_{\text{pred}}} \). The transformer decoder generates the future trajectories in a non-autoregressive pattern and outputs the predictions of \( T_{\text{obs}+1}, \ldots, T_{\text{pred}} \) at one time. A random Gaussian noise is concatenated to the output of the transformer decoder to generate various future predictions [9]. The concatenated features are then embedded by a fully connected layer. The final predictions are obtained by adding the last time input of the transformer encoder with a residual connection. Given the residual connection, it can predict the location offsets from the last observed location \( X_{T_{\text{obs}}} \) to each predicted \( \hat{y}_{t}, t = T_{\text{obs}+1}, \ldots, T_{\text{pred}} \).

4. Experiments
4.1. Datasets and Metrics

Our model was evaluated on two public pedestrian trajectory datasets, ETH [29] and UCY [30], which are widely used for future trajectory prediction. The two datasets are composed of five outdoor scenes that were recorded from a top-view. ETH contains two scenes: ETH and HOTEL. UCY contains three scenes: UNIV, ZARA1, and ZARA2. Since our task does not involve activity prediction and multi future prediction, other datasets such as ActEV/VIRAT [62] and The Forking Paths dataset [63] were excluded. The number of pedestrians in each scene varied from 0 to 51 per frame. The datasets exhibited complex interactions such as nonlinear trajectories, collision avoidance, walking together, moving from different directions, etc. It was recorded at 25 frames every second, the pedestrian trajectory was sampled every 0.4 s. The datasets provided all the pedestrians’ location coordinates in each frame. We evaluated our model following the same “leave-one-out” [64] strategy as commonly adopted by previous works. More specifically, the model was trained and validated on four sets and tested on the remaining one. In order to be consistent with the previous works, the model predicts the next 12 frames’ (4.8 s) conditions on the 8 observed frames (3.2 s) ahead.

Two conventional metrics were employed to evaluate our model, namely, the Average Displacement Error (ADE) and the Final Displacement Error (FDE). The ADE is defined as the average Euclidean distance over all estimated positions in both the predicted trajectory and ground-truth trajectory.

\[
ADE = \frac{\sum_{n \in N} \sum_{t = T_{\text{obs}+1}}^{T_{\text{pred}}} || \hat{Y}_t^n - Y_t^n ||_2^2}{N \times T} \quad t = T_{\text{obs}+1}, \ldots, T_{\text{pred}}
\]
The FDE is the Euclidean distance between the predict position and the ground truth position at the final destination $T_{pred}$.

$$FDE = \sum_{n \in N} ||\hat{Y}_n^t - Y_n^t||_2^{N t} = T_{pred}$$  \hspace{1cm} (14)

4.2. Experimental Settings

Our proposed NaST network was trained on the Pytorch deep learning framework. In our experiment, the original coordinates data were first embedded into dimension 32 by a fully connected layer, followed by ReLU activation. During the training stage, the encoder received each frame embedded and extracted the features for the time steps observed. The sparse spatial and temporal transformers accepted inputs with a feature size of 32. The number of encoder layers in the sparse spatial and sparse temporal encoders were two and one, respectively. The number of self-attention heads was set to eight in each encoder layer. The number of layers in the decoder transformer was set to two, with four multi-heads.

The model was trained with the Adam optimizer [9] for 300 epochs with a batch size of 16. The learning rate was 0.0015. The sparse spatial threshold value $\gamma$ and sparse temporal value $\delta$ were, empirically, set to 0.1 and 0.5. The model was trained by minimizing the Mean Square Error (MSE) loss. During the inference stage, 20 samples were generated for each sample and the closest one to ground truth was used to compute the ADE and FDE metrics.

4.3. Comparison with State-of-the-Arts

We compare our proposed NaST with other models including Social GAN [7], Sophie [15], Social-BiGAT [18], SR-LSTM [8], Social-STGCNN [19], RSGB [64], STAR [27], GraphTCN [65], and SGCN [66]. The results evaluated by ADE and FDE metrics are shown in Table 1. Social GAN [7], Sophie [15], and Social-BiGAT [18] adopted Generative Adversarial Networks (GANs) to generalize trajectory prediction. Social GAN [7] improved over Social LSTM to generate multiple plausible trajectories. Sophie [15] introduced the social and physical attention mechanisms to an LSTM-based GAN model. Social-BiGAT [18] combined graph attention mechanism with GAN for prediction. SR-LSTM [8] computed social interactions by LSTM with pair-wise attention and motion gates. Social-STGCNN [19] modeled the interactions as a graph and proposed a kernel function to embed the social interactions between pedestrians within the adjacency matrix. STAR [27] constructed spatial and temporal transformers to capture social interactions among the crowd. SGCN [66] presented a Sparse Graph Convolution Network for pedestrian trajectory prediction. GraphTCN [65] was a CNN-based method which modeled the spatial interactions as social graphs and captured the spatio-temporal interactions with a modified temporal convolutional network.

<table>
<thead>
<tr>
<th>Models</th>
<th>ETH</th>
<th>HOTEL</th>
<th>UNIV</th>
<th>ZARA1</th>
<th>ZARA2</th>
<th>AVG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Social GAN [7]</td>
<td>0.87/1.62</td>
<td>0.67/1.37</td>
<td>0.76/1.52</td>
<td>0.35/0.68</td>
<td>0.42/0.84</td>
<td>0.61/1.21</td>
</tr>
<tr>
<td>Sophie [15]</td>
<td>0.70/1.43</td>
<td>0.76/1.67</td>
<td>0.54/1.24</td>
<td>0.30/0.63</td>
<td>0.38/0.78</td>
<td>0.51/1.15</td>
</tr>
<tr>
<td>Social-BiGAT [18]</td>
<td>0.69/1.29</td>
<td>0.49/1.01</td>
<td>0.55/1.32</td>
<td>0.30/0.62</td>
<td>0.36/0.75</td>
<td>0.48/1.00</td>
</tr>
<tr>
<td>SR-LSTM [8]</td>
<td>0.63/1.25</td>
<td>0.37/0.74</td>
<td>0.51/1.10</td>
<td>0.41/0.90</td>
<td>0.32/0.70</td>
<td>0.45/0.94</td>
</tr>
<tr>
<td>Social-STGCNN [19]</td>
<td>0.64/1.11</td>
<td>0.49/0.85</td>
<td>0.44/0.79</td>
<td>0.34/0.53</td>
<td>0.30/0.48</td>
<td>0.44/0.75</td>
</tr>
<tr>
<td>RSGB w/o context [64]</td>
<td>0.80/1.53</td>
<td>0.33/0.64</td>
<td>0.80/1.53</td>
<td>0.40/0.86</td>
<td>0.30/0.65</td>
<td>0.48/0.99</td>
</tr>
<tr>
<td>STAR [27]</td>
<td>0.36/0.65</td>
<td>0.17/0.36</td>
<td>0.31/0.62</td>
<td>0.26/0.55</td>
<td>0.22/0.46</td>
<td>0.26/0.53</td>
</tr>
<tr>
<td>SGCN [66]</td>
<td>0.63/1.03</td>
<td>0.32/0.55</td>
<td>0.37/0.70</td>
<td>0.37/0.70</td>
<td>0.25/0.45</td>
<td>0.37/0.65</td>
</tr>
<tr>
<td>GraphTCN [65]</td>
<td>0.59/1.12</td>
<td>0.27/0.52</td>
<td>0.42/0.87</td>
<td>0.30/0.62</td>
<td>0.23/0.48</td>
<td>0.36/0.72</td>
</tr>
<tr>
<td>NaST (Ours)</td>
<td>0.35/0.62</td>
<td>0.15/0.35</td>
<td>0.27/0.56</td>
<td>0.25/0.56</td>
<td>0.19/0.41</td>
<td>0.24/0.50</td>
</tr>
</tbody>
</table>

Table 1. Comparison with some models on datasets ETH and UCY of ADE/FDE metrics (the lower numerical result is better).
We observed that our method outperformed all the competitive methods on these benchmark datasets in terms of the average result of the ADE and FDE metrics. For the average ADE metric, our NaST surpassed the previous best method STAR [27] by 8%. For the average FDE metric, NaST is better than STAR [27] by a margin of 6%. In particular, the results showed that our model yielded better results than those dense methods such as Sophie [15], Social-STGCNN [19], and STAR [27], especially on the more complex datasets UNIV and ZARA2 containing dense crowd scenes. We speculated that the underlying reason is that the dense methods construct superfluous social interactions, which interfere the trajectory prediction. Fortunately, the superfluous interaction is removed by the sparse transformer and focuses on the exact neighbors who are involved in the interaction. Furthermore, the non-autoregressive inference pattern could avoid accumulating errors and thus improve the efficiency of the model.

4.4. Ablation Study

4.4.1. The Individual Module in NaST

To verify the contribution of each component in our model, we conducted exhaustive ablative experiments on both the ETH and UCY datasets. Specifically, we removed one of the components including the sparse Spatial Transformer Encoder (STE), sparse Temporal Transformer Encoder (TTE), and Transformer decoder (TD) in NaST each time and computed predictions, and the results are shown in Table 2. Detailed experiments are introduced in the following.

Table 2. Ablation Study on ETH and UCY datasets of ADE/FDE metrics (the lower numerical result is better).

<table>
<thead>
<tr>
<th>Components</th>
<th>Performance (ADE/FDE)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ETH</td>
</tr>
<tr>
<td>(a)</td>
<td>√</td>
</tr>
<tr>
<td>(b)</td>
<td>-</td>
</tr>
<tr>
<td>(c)</td>
<td>√</td>
</tr>
<tr>
<td>NaST</td>
<td>√</td>
</tr>
</tbody>
</table>

In the degraded model (a), the sparse Temporal transformer Encoder (TTE) is removed, and we fixed the model with the sparse Spatial Transformer Encoder (STE) and the Transformer Decoder (TD). We observed that without the temporal transformer, model (a) suffered from performance reduction compared with the NaST model, especially on HOTEL and ZARA1. We inferred that because the scenes in HOTEL are relatively less crowded, and the spatial interaction in ZARA1 is much simpler. Hence, temporal dependency is more important during the inferring procedure. This illustrated that the temporal transformer can provide effective temporal modeling ability. In model (b), the sparse Spatial Transformer Encoder (STE) is removed and the sparse Temporal Transformer Encoder (TTE) and Transformer decoder (TD) are kept fixed. From the results, we can see that model (b) accrues much worse results on UNIV, which contains dense crowd scenes, which suggests that the spatial transformer is important on modeling social interactions under crowd scenarios. Model (c) removes the Transformer Decoder (TD) and keeps the STE and TTE retained. It also receives a performance decrease. In conclusion, according to the experiment results in Table 2, we can see that removing any component from our model will result in a large performance reduction. The results clearly validate the contribution of each module to NaST for trajectory prediction.

4.4.2. Contribution of Spatial and Temporal Sparsity

In order to evaluate the contribution of the sparsity to the transformers, we conducted studies on the different extent of sparsity on transformer encoder. We designed experiments for sparsity hyper-parameters through a searching protocol. The spatial and temporal
sparsity was evaluated at intervals of 0.1 in the range of 0 to 1. Representative results are shown in Table 3. We can receive different variants of our model by setting different values in the spatial sparse threshold $\gamma$ and the temporal sparse threshold $\delta$. First, we set the temporal sparse threshold $\delta$ fixed with 0.5 and observe the effect of different spatial sparsity. Spatial sparsity $\gamma$ was evaluated at intervals of 0.1 in the range of 0 to 1. In NaST-sp0, the spatial sparse threshold $\gamma$ was set to 0, which meant dense spatial interaction. It modeled the social interaction among the specific pedestrian with all neighbors in a scene. The performance of NaST-sp0 degraded further on the complex scenes such as UNIV, since dense interactions might make the model suffer from overfitting. When the spatial sparse threshold $\gamma$ was set to 1 in NaST-sp1, it meant no interaction between pedestrian pairs. Evidently, the performances degraded, which illustrated the importance of social (spatial) interaction in predicting trajectories. In NaST-sp0.5, it worked well on the relative dense scenes such as UNIV and ZARA2, but it was not satisfactory on other simple scenes when the spatial sparse threshold $\gamma$ was set to 0.5. The reason may be that the sparse mask might filter the important neighbors who interact with the specific pedestrian in the less crowded scenes. When the spatial sparse threshold $\gamma$ is set to 0.1, the results are shown for NaST in Table 3.

Table 3. The results of different extent of spatial and temporal sparsity on ETH and UCY datasets of ADE/FDE metrics (the lower numerical result is better).

<table>
<thead>
<tr>
<th>Variants</th>
<th>ETH</th>
<th>HOTEL</th>
<th>UNIV</th>
<th>ZARA1</th>
<th>ZARA2</th>
<th>AVG</th>
</tr>
</thead>
<tbody>
<tr>
<td>NaST-sp$0$</td>
<td>0.39/0.65</td>
<td>0.18/0.36</td>
<td>0.36/0.67</td>
<td>0.26/0.58</td>
<td>0.23/0.44</td>
<td>0.28/0.54</td>
</tr>
<tr>
<td>NaST-sp$0.3$</td>
<td>0.35/0.63</td>
<td>0.16/0.35</td>
<td>0.28/0.59</td>
<td>0.25/0.56</td>
<td>0.19/0.41</td>
<td>0.25/0.51</td>
</tr>
<tr>
<td>NaST-sp$0.5$</td>
<td>0.37/0.64</td>
<td>0.17/0.36</td>
<td>0.27/0.58</td>
<td>0.26/0.56</td>
<td>0.20/0.41</td>
<td>0.25/0.51</td>
</tr>
<tr>
<td>NaST-sp$0.8$</td>
<td>0.40/0.67</td>
<td>0.21/0.42</td>
<td>0.33/0.67</td>
<td>0.27/0.58</td>
<td>0.23/0.40</td>
<td>0.29/0.55</td>
</tr>
<tr>
<td>NaST-sp$1$</td>
<td>0.41/0.67</td>
<td>0.21/0.42</td>
<td>0.32/0.68</td>
<td>0.27/0.60</td>
<td>0.22/0.43</td>
<td>0.29/0.56</td>
</tr>
<tr>
<td>NaST-tp$0$</td>
<td>0.38/0.65</td>
<td>0.18/0.37</td>
<td>0.31/0.65</td>
<td>0.25/0.59</td>
<td>0.22/0.45</td>
<td>0.27/0.54</td>
</tr>
<tr>
<td>NaST-tp$0.2$</td>
<td>0.35/0.64</td>
<td>0.16/0.37</td>
<td>0.29/0.62</td>
<td>0.26/0.57</td>
<td>0.20/0.42</td>
<td>0.25/0.52</td>
</tr>
<tr>
<td>NaST-tp$0.7$</td>
<td>0.40/0.66</td>
<td>0.19/0.44</td>
<td>0.33/0.61</td>
<td>0.29/0.63</td>
<td>0.21/0.44</td>
<td>0.28/0.56</td>
</tr>
<tr>
<td>NaST-tp$1$</td>
<td>0.43/0.68</td>
<td>0.20/0.44</td>
<td>0.35/0.62</td>
<td>0.29/0.64</td>
<td>0.25/0.47</td>
<td>0.30/0.57</td>
</tr>
<tr>
<td>NaST</td>
<td>0.35/0.62</td>
<td>0.15/0.35</td>
<td>0.27/0.56</td>
<td>0.25/0.56</td>
<td>0.19/0.41</td>
<td>0.24/0.50</td>
</tr>
</tbody>
</table>

Second, we fixed the spatial sparsity to $\gamma = 0.1$ and then evaluated the effect of temporal sparsity. Temporal sparsity $\delta$ is evaluated at intervals of 0.1 in the range of 0 to 1. Representative results are shown in Table 3. NaST-tp$0$ referred to the temporal sparse threshold $\delta = 0$, which meant dense temporal dependency. The prediction of the current time step depended on all the time steps before. In NaST-tp$1$, $\delta$ was set to 1, which indicated that no temporal interaction among different time steps. The model obtains the best average performance in the original NaST when $\gamma = 0.1$ and $\delta = 0.5$. This indicated that the proper sparsity was conducive to making precise predictions.

Different sparse spatial and temporal interaction matrices are obtained according to different spatial and temporal sparsity. Figure 4 shows the sparse spatial interaction matrices of different spatial sparsity in part of ETH scene. In Figure 4a, spatial sparsity $\gamma = 0$, representing dense spatial interaction; each pedestrian has interactions with others. In Figure 4b, the spatial sparsity $\gamma = 0.1$, representing sparse spatial interaction; each pedestrians have interactions with neighbors. In Figure 4c, the spatial sparsity $\gamma = 0.5$; the pedestrians have interactions with fewer neighbors.
shown in Table 4. Hence, each next location is predicted based on the trajectories of previous time steps. The results are shown in Table 4.

Different temporal interaction matrices are obtained according to different temporal sparsity $\delta$. Figure 5 shows part of the sparse temporal interaction matrices of the different temporal sparsity for a pedestrian. In Figure 5a, temporal sparsity $\delta = 0$, representing dense temporal interaction; the current time has interactions with all the other time steps. In Figure 5b, temporal sparsity $\delta = 0.5$, representing sparse temporal interaction; the current time has interaction with part of the other time steps. In Figure 5c, temporal sparsity $\delta = 0.7$; the current time has interaction with fewer other time steps.

4.4.3. Contribution of Non-Autoregressive Prediction

The effectiveness of the non-autoregressive pattern of the transformer decoder was also evaluated. For comparison, we constructed an autoregressive version of our model, named NaST-auto. In NaST, the decoder receives a query sequence generated in advance by using the last observed time step $X_{obs}$ (see Figure 1). Nevertheless, NaST-auto does not receive the query sequence from the same inputs as NaST. NaST-auto predicts one future location each time. Once a prediction is made, it will be added to the historical trajectory sequence and then sent to the transformer encoder for subsequent processing. Hence, each next location is predicted based on the trajectories of previous time steps. The results are shown in Table 4.
Table 4. The comparaison of non-autoregressive and autoregressive inference pattern on ETH and UCY datasets of ADE/FDE metrics (the lower numerical result is better).

<table>
<thead>
<tr>
<th>Variants</th>
<th>Performance (ADE/FDE)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ETH</td>
</tr>
<tr>
<td>NaST-auto</td>
<td>0.33/0.69</td>
</tr>
<tr>
<td>NaST (ours)</td>
<td>0.35/0.62</td>
</tr>
</tbody>
</table>

The original non-autoregressive model NaST exhibited lower errors than its counterpart. The results of the autoregressive version NaST-auto showed a 17% error increase in ADE and 22% in FDE, and it worked even worse on FDE metrics in particular. The reason could be attributed to the fact that the autoregressive inference procedure was prone to error accumulation during prediction. Apart from that, the use of the last time step observation as a query sequence likely helps to predict the future locations and significantly reduces the error. The contribution of the non-autoregressive pattern to the final performance of the pedestrian trajectory prediction is clearly validated.

4.5. Visualization

4.5.1. Trajectory Prediction Visualization

We visualized some common interaction scenes in Figure 6, where yellow dot lines represented observed trajectories, and the blue and red lines were the trajectories predicted by STAR [27] and our proposed NaST model, respectively. The green lines were the ground truth. The visualization revealed that our prediction (red dot line) had better tendencies toward the ground truth. The prediction from STAR [27] (blue dot line) had more deviations from the ground truth.

![Figure 6](image_url)

Figure 6. Trajectory visualization in different scenes. The yellow dot lines represent the observed trajectories, and the blue and red lines are the trajectories predicted by STAR [27] and our proposed model. The green lines are the ground truth. (a) Pedestrians walked in parallel from the same direction. (b) Pedestrians walked in perpendicular direction. (c) Multi pedestrians walked from the same direction. (d) Multi pedestrians walked from different direction.

In Figure 6 scenario (a), two pedestrians were walking in parallel with the same direction. NaST (red dot line) could better match the ground truth compared with STAR (blue dot line). In scenario (b), pedestrians were walking in a perpendicular direction. NaST captured the intention and predicted more accurate directions for each pedestrian. Since one of the pedestrians in scenario (b) had a sharp turn, both methods failed to achieve accurate prediction because only observed trajectory information was given. From scenario (c) and (d), we can see that STAR [27] suffered from the overlap issue with a high possibility.
of collision, while NaST considered both spatial interaction and temporal tendency to avoid collision.

STAR [27] also adopted transformers for feature extraction. It only constructed the encoder stack and predicted future locations in an autoregressive pattern. Errors accumulated during the inference procedure and made the final destination prediction deviate largely from the ground truth. Examples are shown in Figure 6a,c. From the pictures in the top of Figure 6a,c, we can see that the final destination predictions by STAR [27] (the last blue dot of each predicted trajectory) were far away from the ground truth (the last green dot of each predicted trajectory). The results were much better for NaST (pictures in the bottom of Figure 6a,c). The final destination predictions by NaST (the last red dot of each trajectory) were much closer to the ground truth (the last green dot of each trajectory).

4.5.2. Sparse Directed Interaction Visualization

NaST can successfully extract the sparse social interaction of a crowd. We visualized the sparse directed interactions in different scenes, as shown in Figure 7. The images in the top row are different scenes in the ETH [29] and UCY datasets [30]. The sparse directed interaction graphs of each scene are shown in the bottom row. In the scene images, the solid lines represent the observed trajectories, the colored dots indicate the current location, and the dash lines represent the trajectories in the future. Two graphs under each scene are the corresponding sparse directed interaction graphs, which illustrate that one pedestrian is only influenced by a part of the surrounding neighbors. In Figure 7a, the red pedestrian is only influenced by the blue and green ones (the interaction is represented by the directed edges in the left sparse directed interaction graph in Figure 7a). However, the degree of influence from the two neighbors to the red node is different: the thickness of the edges is proportional to the interaction weights. From the left sparse directed interaction graph in Figure 7a, we can see that the influence from the blue node to the red is much larger than from the green one. The right sparse directed interaction graph in Figure 7a shows the influence from the yellow and red nodes to the blue one. In addition, the two sparse directed interaction graphs in Figure 7a illustrate that the interaction between the red and blue nodes is different to each other. It is also demonstrated in Figure 7c that the red and blue pedestrians head from the opposite direction, the red one is influenced by the green and blue pedestrians, and the blue one is also influenced by the red pedestrian. However, the influence between the red and blue pedestrians is not the same degree to each other. When they meet, the red pedestrian is walking straight and keeps their direction without change, and the blue one detours to avoid collision. Therefore, the interaction weight from the red pedestrian to the blue one is larger than blue to red. It is confirmed in the directed graph that the edge from red node to the blue node is much thicker than that from blue to red.

Figure 7. Sparse direct interaction visualization in different scenes. Solid lines represent observed trajectories, colored dots indicate the current location, and the dash lines represent the trajectories in the future. (a) The scene in ZARA1. (b) The scene in ETH. (c) The scene in HOTEL.
5. Conclusions

In this paper, we present Non-autoregressive Sparse Transformer (NaST) networks for Pedestrian Trajectory Prediction. NaST captures the sparse directed spatial and temporal interactions by sparse spatial and sparse temporal transformer encoders separately, which generate trajectory predictions by a transformer decoder in a non-autoregressive pattern and could effectively avoid error accumulation and reduce computational cost. Beyond that, we decoded predictions from a query sequence generated in advance by using the last observed time step for making more accurate predictions. It was found that the sparse spatial transformer could effectively model the directed social interaction in real scenes. The sparse temporal transformer could capture the motion dependency. By combining sparsity and the self-attention in the transformer, we constructed the transformer encoder to extract the features for future trajectory prediction. Beyond that, we investigated the use of non-autoregressive inference with transformer neural networks to avoid error propagation. Extensive experimental studies illustrated that our proposed model achieved better performances than previous methods. The information that NaST used was only the past trajectories of pedestrians who might have failed to predict unexpected sharp turns. Additional information such as roads, vehicles, buildings, and surrounding environment could be considered to solve this issue. We leave it for future study.

Author Contributions: Conceptualization, D.L. and S.L.; methodology, D.L.; software, D.L.; validation, D.L.; Q.L.; formal analysis, M.Q.; investigation, Q.L.; resources, J.K.; data curation, S.L.; writing—original draft preparation, D.L.; writing—review and editing, Q.L.; visualization, D.L.; supervision, J.K.; project administration, J.K.; funding acquisition, J.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the National Natural Science Foundation of China, grant number 62272096, and the Fund of Jilin Provincial Science and Technology Department, grant number 20210201077GX.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.


Conflicts of Interest: The authors declare no conflict of interest.

References


**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.