Abstract: Trajectory prediction is one of the core functions of autonomous driving. Modeling spatial-aware interactions and temporal motion patterns for observed vehicles are critical for accurate trajectory prediction. Most recent works on trajectory prediction utilize recurrent neural networks (RNNs) to model temporal patterns and usually need convolutional neural networks (CNNs) additionally to capture spatial interactions. Although Transformer, a multi-head attention-based network, has shown its notable ability in many sequence-modeling tasks (e.g., machine translation in natural language processing), it has not been explored much in trajectory prediction. This paper presents a Spatial Interaction-aware Transformer-based model, which uses the multi-head self-attention mechanism to capture both interactions of neighbor vehicles and temporal dependencies of trajectories. This model applies a GRU-based encoder-decoder module to make the prediction. Besides, different from methods considering the spatial interactions only among observed trajectories in both encoding and decoding stages, our model will also consider the potential spatial interactions between future trajectories in decoding. The proposed model was evaluated on the NGSIM dataset. Compared with other baselines, our model exhibited better prediction precision, especially for long-term prediction.

Keywords: trajectory prediction; Transformer; attention mechanism; GRU; autonomous driving
performance in long-term trajectory prediction. Third, the trajectory tends to be highly nonlinear over time due to the driver’s decisions [2], which poses a severe challenge for both traditional dynamic models and machine learning models.

Most of the recent studies on trajectory prediction use deep learning methods. To model the interactions among vehicles, previous studies have attempted to represent the spatial information of vehicles as lane-based social tensors or graph structures and apply pooling layers to obtain the social context encoding. Although these methods capture the spatial interaction of historical trajectories of the target vehicle and its neighbors in the encoding stage, they only predict the target vehicle’s future trajectory when decoding and ignore the potential future interactions between the target vehicle and its neighbors. While Transformer [3], a multi-head attention-based network, has shown its more notable ability in many sequence-modeling tasks (e.g., machine translation in natural language processing) than RNNs, it has not been explored much in trajectory prediction. Moreover, previous works usually use two Transformer layers to separately model the temporal dependency of trajectory and the spatial interdependency of vehicles [4,5].

In this paper, we present a spatial interaction-aware Transformer-based model. Unlike the standard Transformer layer containing only one multi-head self-attention module, the novel spatial interaction-aware Transformer (SIT) contains two multi-head self-attention modules. Specifically, these two attention modules have two different attention masks, one for capturing temporal dependencies of trajectories and another for modeling spatial interactions among vehicles. The proposed SIT provides a neat and efficient solution to integrate temporal and spatial context information based on the self-attention mechanism only. By stacking multiple SIT layers, our model can capture more complex and abstract temporal and spatial information. Moreover, the proposed model contains a GRU-based encoder-decoder module on top of SIT layers for making the final prediction. When decoding, for each time step, the decoder will access its last output hidden states of all observed vehicles and use a multi-head self-attention module to guide the message-passing and model the potential future interactions among these vehicles.

We evaluate our method on the public NGSIM US-101 and I-80 datasets. The experimental results show that our method outperforms other baselines with substantial performance improvement. We further conduct ablation studies to demonstrate the superiority of our method over its variants that use the standard Transformer layers or standard GRU encoder-decoder.

The main contributions of this work are summarized as follows:

- A spatial interaction-aware Transformer-based model is proposed to efficiently capture and integrate temporal dependencies of trajectories and spatial interactions among vehicles.
- A decoder that considers message passing for all vehicles is applied to model the potential future interactions among observed vehicles.

2. Literature Review
2.1. Sequence Prediction

RNNs, e.g., GRU [6] and LSTM [7], have achieved great success in sequence prediction tasks, e.g., speech recognition, machine translation, robot decision-making, etc. RNNs also have broad applications in modeling temporal movement patterns of vehicles [2,8–12] and pedestrians [13–17]. RNNs-based trajectory predictors usually have an encoder-decoder architecture. Due to the limitation in spatial interaction modeling, which is essential for trajectory prediction, RNNs usually need to cooperate with an additional structure, such as convolutional neural networks (CNNs) [2,18,19], attention mechanism [4,18] and graph neural networks (GNNs) [8,20,21].

Transformers, based on attention mechanisms, have dominated Natural Language Processing (NLP) in recent years [22–26]. Due to the absence of recurrence, this architecture is more capable of long-term dependency modeling and parallelization training than RNNs. Yu et al. [4] apply two separate Transformers to, respectively, extract spatial and temporal
interactions among pedestrians. However, the Transformer architecture has not been explored much in vehicle trajectory prediction.

2.2. Spatial Interaction Modeling

Conventional approaches [27–29] usually predict the future trajectory of the target object only based on its current state and track history. However, in a crowded road environment, relying only on the trajectory history of the target may lead to inaccurate prediction results, especially for long-term predictions [1]. To model the spatial interaction among vehicles or pedestrians, some studies feed the track history of the target and its surrounding objects to the predictor and use CNNs [2,18,19], attention mechanism [4,18,30,31] or GNNs [8,20,21] to implement message passing among these objects.

Alahi et al. [13] connect neighboring LSTMs through the social-pooling strategy, which allows spatially proximal LSTMs to share information with each other. Deo et al. [2] represent neighboring objects by a social tensor and propose a convolutional social pooling to improve the social pooling method proposed in [13].

Compared to the pooling methods, the attention mechanism can estimate the importance of different neighbors to a given object. Zhang et al. [14] propose a motion gate and a pedestrian-wise attention module to adaptively focus on the most useful neighboring information and guide the message passing. Yu et al. [4] capture spatio-temporal interactions by two separate spatial and temporal Transformers.

In a driving environment, we can regard the vehicles or pedestrians and their interactions as a graph in which the nodes and edges, respectively, represent the objects and the spatial interactions among them. As GNNs naturally fit for graph-structured data, they are also applied to address spatial interaction modeling. Li et al. [20] use a graph to represent the interactions of neighboring objects and apply several graph convolutional blocks to extract features. Yu et al. [4] and Pang et al. [5] utilize a spatial Transformer to model the neighboring objects as a graph and apply a Transformer-based message-passing graph convolution to capture the social interactions. Peng et al. [32] utilize social relation attentions to model spatial interactions based on the relative positions of pedestrians.

To avoid modeling multi-agent trajectories in the time and social dimensions separately, Yuan et al. [33] propose an Agent-aware Transformer to leverage a sequence representation of multi-agent trajectories by flattening trajectory features across time and agents.

Although these studies recognize the interactions among neighboring objects by modeling their spatial relationships, they only consider the interactions among the observed trajectories and ignore the potential interactions between the future trajectories of the target vehicle and its neighbors in the prediction phase.

3. Problem Formulation

This work formulates the trajectory prediction problem as predicting the future trajectories of all objects in an observed scene based on their historical trajectories. Considering that it is easier to predict the velocity of an object than to predict its location [20], we feed historical locations and velocities into our model and let the model predict the future velocities. Then, we accumulate the predicted velocities and the last observed locations to get the final location predictions.

As described above, the inputs $X$ of our model are historical trajectories and velocities of all observed vehicles over $t_h$ time steps:

$$X = [p^{(t-t_h)}, ..., p^{(t-1)}, p^{(t)}]$$

(1)

where

$$p^{(t)} = [(x_0^{(t)}, y_0^{(t)}, u_0^{(t)}, v_0^{(t)}), (x_1^{(t)}, y_1^{(t)}, u_1^{(t)}, v_1^{(t)}), ..., (x_n^{(t)}, y_n^{(t)}, u_n^{(t)}, v_n^{(t)})]$$

(2)

represents the coordinates $(x, y)$ and velocities $(u, v)$ of all vehicles in the observed scene at time $t$; $n$ is the number of observed vehicles. The outputs $Y$ of our model are the
predicted future velocities of all observed vehicles from time step \( t_h + 1 \) to \( t_h + t_f \), and \( t_f \) is the predicted horizon:

\[
Y = [q^{(t_h + 1)}, q^{(t_h + 2)}, \ldots, q^{(t_h + t_f)}]
\]  

(3)

where

\[
q^{(t)} = [(u^{(t)}_0, v^{(t)}_0), (u^{(t)}_1, v^{(t)}_1), \ldots, (u^{(t)}_n, v^{(t)}_n)]
\]  

(4)

Following [2,20], the vehicles are observed within 90 feet from the center of the target vehicle.

4. Methodology

Figure 1 shows our proposed model, which consists of three components: an input preprocessing module, the spatial interaction-aware Transformer layers, and a trajectory prediction model.

Figure 1. The architecture of the proposed method SIT-ID. Given a traffic scene with \( t_h \) observed frames, it first preprocesses the raw trajectory data into the input representation \( X \in \mathbb{R}^{n \times t_h \times c} \). After the following two operations: Embedding and Positional Encoding, we use the proposed SIT layers to capture the temporal dependency and the spatial interaction. Then, a GRU-based encoder-decoder module is used to make the final predictions. For each decoding step, the decoder allows the message-passing between all objects to capture the potential interactions. \( X[i, -1; 2] \) represents the last observed velocities of all observed vehicles. The images of the car logos in this figure are from here (https://www.flaticon.com/, accessed on 11 November 2021).

4.1. Input Preprocessing Module

4.1.1. Input Representation

Following [20], for subsequent efficient computation, we do not directly feed the raw trajectory data of objects into our model. Given a traffic scene, assuming there are \( n \) objects observed in the past \( t_h \) time steps, we preprocess the raw data into a three-dimensional tensor \( X \in \mathbb{R}^{n \times t_h \times c} \), as shown in Figure 1. We set \( c = 4 \) to mark an object’s coordinate \((x, y)\) and velocity \((u, v)\) at a time step, and normalize all coordinates and velocities to the range of \((-1, 1)\).

4.1.2. Spatial Graph Construction

In traffic scenarios, a vehicle’s movement is greatly affected by that of its surrounding vehicles. Therefore, we think it is efficient to represent the interdependencies among vehicles as undirected graphs. Specifically, for each observed time step \( t \), we construct
an undirected graph $G_t = \{V_t, E_t\}$, in which the nodes $V_t$ and the edges $E_t$, respectively, represent the objects and the spatial interactions among them. The node set at time step $t$ is defined as $V_t = \{v_t^i | i = 1, 2, ..., n\}$, while the edge set $E_t$ at time step $t$ is denoted as $E_t = \{v_t^i, v_t^j | v_t^i, v_t^j \in V_t\}$.

At each time step $t$, we consider a spatial interaction only happens when the current distance between two objects is shorter than a threshold $T_{\text{close}}$ and these two objects are on the neighboring lanes, e.g., $\text{abs}(\text{lane}^i_t - \text{lane}^j_t) <= T_{\text{lane}} = 1$. For computation efficiency, we can represent $E_t$ as an adjacency matrix $A_t \in \mathbb{R}^{n \times n}$. Thus, at each time step $t$,

$$A_t[i][j] = A_t[j][i] = \begin{cases} 1 & \text{if distance}(v_t^i, v_t^j) <= T_{\text{close}} \text{ and abs}(\text{lane}^i_t - \text{lane}^j_t) <= 1 \\ 0 & \text{else} \end{cases}$$

where $n$ is the number of observed vehicles. Given $n$ vehicles’ observed trajectories with a length of $t_h$ time steps, we can obtain the adjacency matrices $A = \{A_t\}_{t=1}^{t_h}$ as described above. These adjacency matrices are parts of our model’s inputs.

4.2. Spatial Interaction-Aware Transformer

Given an input data $X \in \mathbb{R}^{n \times t_h \times c}$ obtained from the preprocessing module, we first perform the following two operations:

4.2.1. Embedding

We mark $x_t^i = X[i, t]$ and apply this embedding network $\phi$ to map $x_t^i \in \mathbb{R}^c$, which represents the state of object $i$ at time step $t$, into a hidden representation $e_t^i \in \mathbb{R}^{d_{\text{model}}}$, in which the coordinate and velocity are unified to ease the subsequent context modeling task:

$$e_t^i = \phi(x_t^i, W_e)$$

where $W_e$ is the embedding weight. This paper uses a multiple layer perceptron (MLP) as the embedding network $\phi$.

4.2.2. Positional Encoding

Although the Transformer architecture can capture longer sequence dependencies and obtain massive speed-up when training by avoiding the RNNs’ method of recurrence mechanism, it does not have any sense of order for each element in a sequence. Consequently, it is vital to incorporate the order of the input elements into the Transformer model, especially when we handle time-series data, e.g., trajectory data. Therefore, in this paper, each input embedding $e_t^i$ is time-stamped with its time $t$ by adding a positional encoding vector $\text{pos}^t$ to form $h_t^i$. Both $e_t^i$ and $\text{pos}^t$ have the same dimensionality of $d_{\text{model}}$. For simplicity, we initialize the positional encoding vectors as a matrix $P \in \mathbb{R}^{t_h \times d_{\text{model}}}$, in which each row vector $P[t]$ represents the positional encoding vector of time step $t$. Thus, $h_t^i = e_t^i + P[t]$. This ensures a unique time stamp for each historical location of an object. The matrix $P$ will be optimized in company with the model when training.

By performing the above two operations on each $x_t^i$ for $i \in [1, n]$ and $t \in [1, t_h]$, we can obtain $H \in \mathbb{R}^{n \times t_h \times d_{\text{model}}}$, which is the input of the first spatial interaction-aware Transformer layer.

Unlike the standard Transformer encoder layer, which only fits in modeling the temporal dependency, the proposed spatial interaction-aware Transformer (SIT) layer can capture and integrate temporal dependencies of trajectories and spatial interactions among vehicles. As shown in Figure 2, compared to the standard Transformer layer, our SIT also contains a Spatial Graph Multi-Head Attention Network, which is used to capture the spatial interactions between closing vehicles based on the obtained adjacency matrices $A$. The following content describes how an SIT layer models temporal dependencies of trajectories and spatial interactions among vehicles using the Temporal Multi-Head Attention module and the Spatial Graph Multi-Head Attention Network.
4.2.3. Temporal Multi-Head Attention Module

Similar to the standard Transformer encoder layer, SIT uses a masked multi-head attention module to capture each vehicle’s trajectory’s temporal dependency independently. This masked attention module prevents steps from attending subsequent steps. Given the input $H \in \mathbb{R}^{n \times t_h \times d_{model}}$, the attention module first compute the query matrices $\{Q^i\}_{i=1}^n$, key matrices $\{K^i\}_{i=1}^n$ and the value matrices $\{V^i\}_{i=1}^n$. For $i$-th vehicle, we calculate

$$Q^i = f_Q(\{h^t_{i} \}_{t=1}^{t_h}), \quad K^i = f_K(\{h^t_{i} \}_{t=1}^{t_h}), \quad V^i = f_V(\{h^t_{i} \}_{t=1}^{t_h}) \quad (6)$$

where $f_Q$, $f_K$ and $f_V$ are the corresponding query, key and value functions shared by vehicles $i \in [1, n]$; and $h^t_{i} = H[i, t], \{h^t_{i} \}_{t=1}^{t_h} \in \mathbb{R}^{t_h \times d_{model}}$. For the trajectory of vehicle $i$, as shown in Figure 3a, we define the message passing from time step $s$ to $t$ as

$$m^{s\rightarrow t} = (q^t_{i})^T k^s_{i} \quad (7)$$

Then, we compute the masked attention for vehicle $i$ at time step $t$ as follows:

$$\text{Attention}(t) = \frac{\text{Softmax}(\{m^{s\rightarrow t}_{j=1}^{t_h-1} \}_{i=1}^n)_{j=s}^{t-1}}{\sqrt{d_k}} \quad (8)$$

where $\{m^{s\rightarrow t}_{j=1}^{t_h-1} \}_{i=1}^n$ shows the current step can only access its previous steps. Similarly, we can obtain the masked multi-head attention ($k$ heads) for vehicle $i$ for time step $t$:

$$T^i_{f} = f_O(\text{head}_1; \text{head}_2; ...; \text{head}_k) \quad (9)$$

where $f_O$ is a fully connected layer that merges the $k$ heads’ information. After calculating the multi-head attention $T^i_{f}$ for each vehicle $i \in [1, n]$ and each time step $t \in [1, t_h]$, we
obtain $T \in \mathbb{R}^{n \times t \times d_{\text{model}}}$, which contains the extracted temporal information from the historical trajectories.

### Figure 3. (a) The temporal message-passing: the hidden representation of vehicle $i$ at time step $t$, i.e., $h^i_t$, can only access its previous steps’ hidden states $\{h^i_1, ..., h^i_{t-1}\}$; (b) The spatial message-passing is used in the Spatial Graph Multi-head Attention Network, which only allows the message-passing to happen between neighboring vehicles at each step.

#### 4.2.4. Spatial Graph Multi-Head Attention Network

Based on the obtained $T \in \mathbb{R}^{n \times t \times d_{\text{model}}}$ and adjacency matrices $A$, a spatial graph multi-head attention network is applied to extract the spatial interactions among the observed vehicles.

The self-attention mechanism can be regarded as message passing on an undirected fully connected graph. For a time step $t$, we can get $n$ vehicles’ features $\{h^i_t\}_{i=1}^n \in \mathbb{R}^{n \times d_{\text{model}}}$ from $T$ and represent its corresponding query vector, key vector and value vector, respectively, as $q^i_t = f_Q(h^i_t)$, $k^i_t = f_K(h^i_t)$ and $v^i_t = f_V(h^i_t)$. Similar to Section 4.2.3, we calculate

$$Q_t = f_Q(\{h^i_t\}_{i=1}^n), \quad K_t = f_K(\{h^i_t\}_{i=1}^n), \quad V_t = f_V(\{h^i_t\}_{i=1}^n)$$

(11)

and define the message passing from vehicle $j$ to $i$ in the fully connected graph as

$$m^{j \rightarrow i} = (q^j_t)^T k^i_t$$

(12)

then the attention at time step $t$ can be calculated as

$$\text{Attention}(Q_t, K_t, V_t) = \frac{\text{Softmax}(\{m^{j \rightarrow i}\}_{j=1}^n)_{[v^i_t]}_{[v^j_t]}}{\sqrt{d_k}}$$

(13)

However, it is inefficient to regard the spatial interactions among vehicles as a fully connected graph. Therefore, we use the adjacency matrices $A$ to replace the fully connected graph above, which ensures the message passing from vehicle $j$ to $i$ at a time step $t$ only when the current distance of these two vehicles is shorter than a threshold $T_{\text{close}}$ and the two vehicles are on the neighboring lanes, as shown in Figure 3b. Then we can rewrite the attention calculation of vehicle $i$ at time step $t$:

$$\text{Attention}(i) = \frac{\text{Softmax}(\{m^{j \rightarrow i}\}_{j \in N(i)}))_{[v^i_t]}_{[v^j_t]}}{\sqrt{d_k}}$$

(14)

where $N(i) = \{j | A[i,j] = 1, j \in [1,n]\}$ represents a neighbor set of vehicle $i$. Similarly, we can obtain the multi-head attention ($k$ heads) of vehicle $i$ for time step $t$:
\[ S_i^t = f_O(\text{head}_1; \text{head}_2; \ldots; \text{head}_k) \]  \hspace{1cm} (15)

where \( \text{head}_j = \text{Attention}_j(i) \) \hspace{1cm} (16)

where \( f_O \) is a fully connected layer that merges the \( k \) heads’ information. After calculating the multi-head attention \( S_i^t \) for each vehicle \( i \in [1, n] \) and each time step \( t \in [1, t_h] \), we obtain \( S \in \mathbb{R}^{n \times t_h \times d_{\text{model}}} \), which contains the extracted interaction information among the observed vehicles. We stack multiple SIT layers to capture more complex and abstract temporal and spatial information.

### 4.3. Trajectory Prediction Module

We apply a GRU-based encoder-decoder module to predict the future trajectories of all observed vehicles. The outputs of the last SIT layer will be fed into the GRU encoder. At the first decoding step, both the hidden feature of the encoder and the velocities of all objects at the last observed time step are fed into the decoder to predict vehicles’ velocities. For the following decoding steps, the decoder takes both the hidden feature of itself and the predicted velocities of all objects at the previous time step as inputs to make the prediction.

However, such decoding processes ignore the potential interactions among the future trajectories of observed vehicles. To model those potential interactions, for each decoding step, our decoder will access the previous step’s hidden features of vehicles and use a multi-head self-attention module to guide the message-passing among those vehicles. Then, the decoder takes the interacted hidden features instead of the origin hidden features as input to make the final prediction, as shown in Figure 1.

### 4.4. Implementation Details

Following Li et al. [8], we process a traffic scene within ±90 feet and all vehicles in this scene will be observed and predicted in the future. When constructing the adjacency matrices \( A \), we set \( T_{\text{close}} = 50 \) feet. In spatial interaction-aware Transformer layers, we let \( d_{\text{model}} = 128 \); the number of head of multi-head attention modules is 4; and the number of SIT layers is 2.

In the GRU-based encoder-decoder module, both the encoder and decoder are a two-layer GRU. We set the number of hidden units of GRUs equals to 60 and apply a \( \tanh \) activation function to rescale the output of decoder to range of \((-1, 1)\).

Our code is implemented using PyTorch Library [34], we train our model as a regression task. The overall loss can be calculated as:

\[ \text{Loss} = \frac{1}{t_f} \sum_{t=1}^{t_f} \left\| Y_i^{\text{pred}} - Y_i^{\text{gold}} \right\|^2 \]  \hspace{1cm} (17)

where \( t_f \) is the number of time step to be predicted in the future, \( Y_i^{\text{pred}} \) and \( Y_i^{\text{gold}} \) are predicted positions and ground truth at time step \( t \), respectively. We train the model using Adam [35] optimizer with \( \eta = 0.001 \), \( \beta_1 = 0.9 \), and \( \beta_2 = 0.999 \). The learning rate is 0.0001.

We set \( \text{batch\_size} = 32 \) during training. We apply Teacher Forcing in training to accelerate convergence.

### 5. Experimental Evaluation

#### 5.1. Experimental Setting

This section presents the evaluation of the proposed model. For a fair comparison with other methods, our model was trained and evaluated on two publicly available datasets. We perform the experiments on a desktop running Ubuntu 18.04 with 2.50GHZ Intel Xeon E5-2678 CPU, 32 GB Memory, and an NVIDIA 1080Ti Graphics Card.

#### 5.1.1. Dataset

The proposed model was trained and evaluated using the public NGSIM US-101 and I-80 datasets. Both datasets were captured at 10 Hz over 45 min and split into three periods.
of 15 min. These periods represent mild, moderate, and congested traffic conditions. These two datasets consist of vehicles’ trajectories on real freeway traffic. Each vehicle’s trajectory was divided into segments of 8 s, where the first 3 s are used as observed track history and the remaining 5 s are the prediction horizon. Following Deo et al. [2], the trajectory data were down-sampled for 10 Hz to 5 Hz, i.e., five frames per second. The two datasets above are merged into one dataset, which is randomly shuffled and divided into the training set, validation set, and test set at a ratio of 7:1:2. The following experimental evaluations are conducted on the test set. The code for data preprocessing and dataset segmentation can be downloaded at GitHub (https://github.com/nachiket92/conv-social-pooling, accessed on 10 October 2021).

5.1.2. Evaluation Metrics

We use the same evaluation metrics as other methods [2,18] and report our evaluation results in terms of the root of the mean squared error (RMSE) of the predicted future trajectories for each time step within the 5 seconds prediction horizon. The RMSE at time step $t$ can be calculated as follows:

$$\text{RMSE}_t = \sqrt{\frac{1}{m} \sum_{i=1}^{m} (Y_{t, i}^{\text{pred}} - Y_{t, i}^{\text{gold}})^2}$$  \hspace{1cm} (18)

where $m$ is the number of vehicles in the test dataset, $Y_{t, i}^{\text{pred}}$ and $Y_{t, i}^{\text{gold}}$ are predicted positions and ground truth at time step $t$, respectively.

5.2. Ablation Study

5.2.1. Ablation Experiments on Neighboring Thresholds

As mentioned in Section 4.1.2, we introduce two thresholds to construct the neighboring graph: the neighboring distance threshold $T_{\text{close}}$ and the lane difference limit $T_{\text{lane}}$.

In this subsection, we conduct two experiments to present the impacts of different $T_{\text{close}}$ and various $T_{\text{lane}}$ on our model SIT-ID. The range of $T_{\text{lane}}$ we apply in our ablation experiments is $[0, 2]$, while the $T_{\text{close}}$ values we select are 0, 30, 50, 70, and 90 feet. As shown in Figure 4a, when we fix $T_{\text{lane}} = 1$, $T_{\text{close}} = 50$ performs better than other neighboring distance thresholds. From Figure 4b, we can see that the optimal lane difference limit is 1 if $T_{\text{close}} = 50$. Therefore, considering too many neighboring vehicles or not considering neighboring vehicles at all will degrade model performance. Based on these observations, in this paper, we set $T_{\text{close}} = 50$ feet and $T_{\text{lane}} = 1$ as our default setting unless specified otherwise.

Figure 4. (a) Comparison among various $T_{\text{close}}$ values when $T_{\text{lane}} = 1$; (b) Comparison among various $T_{\text{lane}}$ values when $T_{\text{close}} = 50$ feet.
5.2.2. Ablation Experiments on the Proposed Model

In this subsection, we perform three ablation experiments on the proposed model SIT-ID. First, we compare the proposed SIT layers and the standard Transformer (ST) layer to verify whether our Spatial Graph Multi-head Attention Network can improve precision by capturing the spatial interaction. ST-GD and SIT-GD both use a standard GRU encoder-decoder module to make predictions. The ST layer used here can only capture the temporal dependency of each vehicle’s historical trajectory. As shown in Table 1, the SIT-GD model performs better than the ST-GD model in terms of RMSE values, especially in long-term future predictions. The SIT layers reduced the RMSE5s value by 25.8% compared to the standard Transformer layers. This result shows that the proposed SIT layer can capture more useful information for trajectory prediction by using the Spatial Graph Multi-head Attention Network to model the interactions among neighboring vehicles, which verifies the importance of the spatial interactions among vehicles in trajectory prediction.

Table 1. Comparison of RMSE values for ablation studies of the proposed method. These values are converted into the meter unit. ST: the standard Transformer layer; WoE: without GRU encoder, only GRU decoder is used for decoding; GD: standard GRU encoder-decoder; ID: standard GRU encoder and spatial interaction-aware GRU decoder.

| Model   | RMSE1s | RMSE2s | RMSE3s | RMSE4s | RMSE5s | RMSE5s (Improvement on ST-GD) |
|---------|--------|--------|--------|--------|--------|-------------------------------|
| ST-GD   | 0.68   | 1.54   | 2.74   | 4.21   | 5.93   | 0 (↑ 0.0%)                   |
| SIT-WoE | 0.58   | 1.26   | 2.11   | 3.17   | 4.48   | 1.45 (↑ 24.4%)               |
| SIT-GD  | 0.58   | 1.26   | 2.09   | 3.13   | 4.40   | 1.53 (↑ 25.8%)               |
| SIT-ID  | 0.58   | 1.23   | 1.99   | 2.96   | 4.05   | 1.88 (↑ 31.7%)               |

Second, to check the effectiveness of the GRU encoder in our framework, we compare these two models: SIT-GD and SIT-WoE. SIT-WoE is the model without the GRU encoder, and its GRU decoder directly takes as input the hidden state of the last step of SIT layers. SIT-GD uses a standard GRU encoder-decoder to make predictions. As shown in Table 1, SIT-GD is slightly better than SIT-WoE, the RMSE5s values of the two models are 4.40 and 4.48, respectively. This result confirms the effectiveness of the GRU encoder. However, we think the GRU encoder can be removed if we can find a better way to utilize the hidden states of SIT layers, such as the adoption of attention mechanisms or pooling methods. We leave it for future study.

Third, to validate the effect of considering potential interactions among the observed vehicles’ future trajectories in decoding, we contrast the proposed interaction-aware GRU decoder and the standard GRU decoder. SIT-GD and SIT-ID both use two SIT layers to capture temporal and spatial dependencies, but the former uses the standard GRU encoder-decoder to make predictions, while the latter applies a standard GRU encoder and a spatial interaction-aware GRU decoder. As shown in Table 1, the latter improves the RMSE values of long-term future predictions (e.g., RMSE4s and RMSE5s) still further, which substantiates that considering the potential interactions among vehicles in decoding is also essential to trajectory prediction, especially long-term trajectory prediction.

To highlight the importance of modeling the spatial interaction, we report the results of these three models on congested traffic scenes. We think a traffic scene is congested when the number of its observed vehicles is equal or greater to 12. From Tables 1 and 2, we can see that the models considering the spatial interaction, i.e., SIT-GD and SIT-ID, widened the gap with ST-GD in congested traffic scenes, compared to non-congested traffic scenes. In congested traffic scenes, SIT-GD further widened the gap from 25.8% to 38.6%, while SIT-ID widened from 31.7% to 40.3%.
Table 2. Comparison of RMSE values for ablation studies on congested traffic scenes. These values are converted into the meter unit.

| Model   | RMSE$_{1s}$ | RMSE$_{2s}$ | RMSE$_{3s}$ | RMSE$_{4s}$ | RMSE$_{5s}$ | (Improvement on ST-GD) |
|---------|-------------|-------------|-------------|-------------|-------------|-----------------------|
| ST-GD   | 0.56        | 1.40        | 2.48        | 3.80        | 5.31        | 0 (↑ 0.0%)            |
| SIT-GD  | 0.48        | 1.02        | 1.62        | 2.53        | 3.26        | 2.05 (↑ 38.6%)       |
| SIT-ID  | 0.48        | 1.01        | 1.60        | 2.31        | 3.17        | 2.14 (↑ 40.3%)       |

5.3. Compared Models

We compare the proposed model to the following baselines:

- Constant velocity (CV) [2]: This method simply uses a constant-velocity Kalman filter to predict trajectories.
- Vanilla LSTM (V-LSTM) [2]: This approach does not consider interactions and uses a LSTM-based encoder-decoder structure to make predictions.
- LSTM with fully connected social pooling (S-LSTM) [13]: Different from V-LSTM, this work incorporates the historical trajectories of the target’s surrounding vehicles and uses a fully connected layer to fuse the encoded representations of the target vehicle and its surrounding vehicles in decoding.
- LSTM with convolutional social pooling (CS-LSTM) [2]: This method applies the convolutional social pooling layer to consider interactions among the target and its surrounding vehicles based on a spatial grid. The output is the unimodal trajectory distribution.
- CS-LSTM(M) [2]: Different from CS-LSTM, this model outputs the maneuver-based multimodal trajectory distribution. The mode with the highest probability is used for evaluating.
- Dynamic and static context-aware attention network (DSCAN) [18]: This method utilizes an attention mechanism to decide which surrounding vehicles are more important to the target vehicle and considers the environment information by using a constraint network.

5.4. Compared Results

Table 3 presents the RMSE values of the models compared. We perceive that CV and V-LSTM yield much higher RMSE values than other models. These two models only use the target vehicle’s track history, while other models utilize the surrounding vehicles’ motion information. This result demonstrates that considering inter-vehicle interactions is essential to trajectory prediction.

Table 3. Comparison of RMSE values with other baseline methods. These values are converted into the meter unit.

| Model   | RMSE$_{1s}$ | RMSE$_{2s}$ | RMSE$_{3s}$ | RMSE$_{4s}$ | RMSE$_{5s}$ |
|---------|-------------|-------------|-------------|-------------|-------------|
| CV [2]  | 0.73        | 1.78        | 3.13        | 4.78        | 6.68        |
| V-LSTM [2] | 0.68      | 1.65        | 2.91        | 4.46        | 6.27        |
| S-LSTM [13] | 0.65    | 1.31        | 2.16        | 3.25        | 4.55        |
| CS-LSTM [2] | 0.61      | 1.27        | 2.09        | 3.10        | 4.37        |
| CS-LSTM(M) [2] | 0.62    | 1.29        | 2.13        | 3.20        | 4.52        |
| DSCAN [18] | 0.58      | 1.26        | 2.03        | 2.98        | 4.13        |
| SIT-ID (ours) | 0.58   | 1.23        | 1.99        | 2.96        | 4.05        |

We note that CS-LSTM(M) leads to higher RMSE values than CS-LSTM. As mentioned in [2], this could be partly due to misclassified maneuvers.

We also note that our SIT-ID produces lower RMSE values compared to the S-LSTM, CS-LSTM, and DSCAN, especially for long-term predictions, e.g., RMSE$_{4s}$ and RMSE$_{5s}$. S-LSTM, CS-LSTM, and DSCAN do not consider the potential interactions in decoding. This
result shows that considering the potential interactions among vehicles in decoding also significantly impacts trajectory prediction, especially for long-term trajectory predictions.

5.5. Visualization of Prediction Results

We visualize a good and a bad prediction case selected from the test set, in Figure 5a and Figure 5b, respectively. After observing 3 seconds of history trajectories, our SIT-ID predicts the trajectories over 5 seconds in the future. We use different colors to distinguish different vehicles; the solid line represents the observed trajectories, while the markers “+” and “•” represent the ground truth in the future and the predicted results, respectively. The red colors correspond to the cars located in middle which is the target that CS-LSTM [2] and DSCAN [18] try to predict. The good case shows that our model can precisely predict the trajectories of all vehicles in an observed scene simultaneously. But, as can be seen from the bad case, our model performs poorly in case of an emergency lane change that happens right after the observation stage. We think this is mainly because the samples in the NGSIM dataset containing emergency lane changes are too few. Therefore, in the near future, we would like to evaluate our model on other datasets, e.g., the Apollo dataset [10], in which data is captured not only on a highway but also from urban areas.

Figure 5. Visualization of SIT-ID’s prediction results. (a) a well predicted example; (b) a poorly predicted example. Different colors represent different vehicles; the solid line represents the observed trajectories, while the markers “+” and “•” represent the ground truth in the future and the predicted results, respectively. The red colors correspond to the cars located in middle which is the target that CS-LSTM [2] and DSCAN [18] try to predict.

5.6. Attention Distribution Analysis

The Temporal Multi-Head Attention (TMHA) module and the Spatial Graph Multi-Head Attention Network (SGMA) are based on the attention mechanism. Attention in deep learning can be broadly interpreted as a vector of importance weights, which reflect one element how strongly it is correlated with other elements. Therefore, to further analyze the mechanism of our model, we visualize the attention distributions produced by the TMHA and SGMA of the last SIT layer of our model.

Figure 6 shows a sample of temporal attention distributions calculated by the TMHA module. We use $k$-head attention mechanisms in both the TMHA and SGMA and set $k = 4$, so there are four different distributions corresponding to different attention heads, respectively. Inspecting the attention distribution of head 2 in Figure 6, we note that for each time step, its attention is mainly distributed to the current and the previous few steps, and the farther away in time, the lower the attention weight. This mechanism is similar to
humans. When driving, a human driver anticipates the motion of a neighboring vehicle, usually based on the recent locations of this vehicle and does not consider its locations of a long time ago.

Figure 6. A sample of temporal multi-head attention distributions calculated by the last SIT layer of our model. The lighter color indicates the greater attention weight. We use masks to prevent steps from attending subsequent steps, so the attentions between a step and subsequent steps are masked to 0.

Figure 7 represents a sample of spatial attention distributions calculated by the SGMA. The values in the grid are the Euclidean distance between the corresponding vehicles in the foot unit. We note that the attention weights tend to be slightly symmetrical along the diagonal. Besides, these weights are linearly related to the Euclidean distances, i.e., a smaller distance usually has a more significant attention weight. This attention distribution is also similar to humans; given a time step, a human driver should pay more attention to vehicles closer to him.

The above analysis shows that the TMHA and SGMA used in our proposed SIT can effectively capture temporal dependencies of trajectories and spatial interactions of vehicles.
The main advantages of the proposed model are summarized as follows:

- Transformer (SIT) layers are utilized to obtain useful context information for trajectory predictions.
- To consider the future potential interactions, for each decoding step, the temporal dependencies of trajectories and spatial interactions among vehicles, respectively.
- The interactions of its surrounding vehicles. Considering the interactions among vehicles, both in the historical trajectory encoding and the future trajectory decoding stages, is essential to trajectory prediction. Thus, this paper proposes a spatial interaction-aware Transformer-based model. In the encoding stage, the proposed Spatial Interaction-aware Transformer (SIT) layers are utilized to obtain useful context information for trajectory prediction. The SIT layer contains two key modules: the Temporal Multi-Head Attention module and the Spatial Graph Multi-Head Attention Network, which are applied to capture temporal dependencies of trajectories and spatial interactions among vehicles, respectively.

In the decoding stage, a GRU-based encoder-decoder module is applied to make the final predictions. To consider the future potential interactions, for each decoding step, the decoder first access the last states of all observed vehicles and control the message-passing among them based on the multi-head attention mechanism, then make a prediction for each vehicle.

The proposed model was evaluated using the public NGSIM US-101 and I-80 datasets. The main advantages of the proposed model are summarized as follows:

- The proposed SIT-based model can predict the trajectory more accurately than other baselines, especially for long-term prediction and in highly interactive situations. Because it considers interactions among vehicles both in the encoding and the decoding stages.

6. Conclusions

In highly dynamic traffic scenes, the vehicle’s subsequent movements are affected by the interactions of its surrounding vehicles. Considering the interactions among vehicles, both in the historical trajectory encoding and the future trajectory decoding stages, is essential to trajectory prediction. Thus, this paper proposes a spatial interaction-aware Transformer-based model. In the encoding stage, the proposed Spatial Interaction-aware Transformer (SIT) layers are utilized to obtain useful context information for trajectory prediction. The SIT layer contains two key modules: the Temporal Multi-Head Attention module and the Spatial Graph Multi-Head Attention Network, which are applied to capture temporal dependencies of trajectories and spatial interactions among vehicles, respectively.

In the decoding stage, a GRU-based encoder-decoder module is applied to make the final predictions. To consider the future potential interactions, for each decoding step, the decoder first access the last states of all observed vehicles and control the message-passing among them based on the multi-head attention mechanism, then make a prediction for each vehicle.

The proposed model was evaluated using the public NGSIM US-101 and I-80 datasets. The main advantages of the proposed model are summarized as follows:

- The proposed SIT-based model can predict the trajectory more accurately than other baselines, especially for long-term prediction and in highly interactive situations. Because it considers interactions among vehicles both in the encoding and the decoding stages.
• The proposed SIT layers can effectively capture and integrate temporal dependencies of trajectories and spatial interactions among vehicles when encoding. In the ablation study, the SIT layers reduced the RMSE$_5$ value by 25.8% compared to the standard Transformer layers.

Due to the datasets used in the work consisting of only highway sections, which are more simple than typical traffic scenes, e.g., urban traffic scenes, our results have certain limitations in generalization. Considerably more work will need to be done to adapt to complex environments and incorporate traffic information, such as lane types and traffic lights.

**Author Contributions:** Conceptualization, Jing Xia and Xiaolong Li; methodology, Jing Xia and Xiaolong Li; formal analysis, Xiaolong Li and Jing Xia; investigation, Jing Xia; writing—original draft preparation, Jing Xia and Xiaolong Li; writing—review and editing, Xiaoyong Chen, Yongbin Tan and Jing Chen; visualization, Jing Xia; supervision, Xiaolong Li and Xiaoyong Chen. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was supported in part by the National Key R&D Program of China (Grant No. 2017YFB0503700) and the Open Research Fund Program of LIESMARS (Grant No. 2001).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Publicly available datasets were analyzed in this study. This data can be found here: https://ops.fhwa.dot.gov/trafficanalysistools/ngsim.htm, accessed on 25 September 2021.

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

**References**

1. Mozaffari, S.; Al-Jarrah, O.Y.; Dianati, M.; Jennings, P.; Mouzakitis, A. Deep Learning-Based Vehicle Behaviour Prediction for Autonomous Driving Applications: A Review. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 33–47. [CrossRef]

2. Deo, N.; Trivedi, M.M. Convolutional Social Pooling for Vehicle Trajectory Prediction. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Salt Lake City, UT, USA, 18–22 June 2018; IEEE: Salt Lake City, UT, USA, 2018; pp. 1549–15498. [CrossRef]

3. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. *arXiv* **2017**, arXiv:abs/1706.03762.

4. Yu, C.; Ma, X.; Ren, J.; Zhao, H.; Yi, S. Spatio-Temporal Graph Transformer Networks for Pedestrian Trajectory Prediction. *Computer Vision—ECCV* 2020, *Springer International Publishing: Cham*, Switzerland, 2020; Volume 12357, pp. 507–523. [CrossRef]

5. Pang, Y.; Zhao, X.; Hu, J.; Yan, H.; Liu, Y. Bayesian Spatio-Temporal Graph Transformer Network (B-Star) for Multi-Aircraft Trajectory Prediction. *arXiv* 2020, arXiv:2003.08111.

6. Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *arXiv* **2014**, arXiv:1412.3555.

7. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]

8. Li, X.; Ying, X.; Chuah, M.C. GRIP: Graph-Based Interaction-Aware Trajectory Prediction. In Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, 27–30 October 2019; pp. 3960–3966. [CrossRef]

9. Giulari, F.; Hasan, I.; Cristani, M.; Galasso, F. Transformer Networks for Trajectory Forecasting. *arXiv* **2020**, arXiv:2003.08111.

10. Ma, Y.; Zhu, X.; Zhang, S.; Yang, R.; Wang, W.; Manocha, D. TrafficPredict: Trajectory Prediction for Heterogeneous Traffic-Agents. *arXiv* **2019**, arXiv:1811.02146. [CrossRef]

11. Chandra, R.; Bhattacharya, U.; Bera, A.; Manocha, D. TraPHic: Trajectory Prediction in Dense and Heterogeneous Traffic Using Weighted Interactions. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; IEEE: Long Beach, CA, USA, 2019; pp. 8475–8484. [CrossRef]

12. Deo, N.; Trivedi, M.M. Multi-Modal Trajectory Prediction of Surrounding Vehicles with Maneuver Based LSTMs. In Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, China, 26–30 June 2018; pp. 1179–1184. [CrossRef]

13. Alahi, A.; Goel, K.; Ramanathan, V.; Robicquet, A.; Fei-Fei, L.; Savarese, S. Social LSTM: Human Trajectory Prediction in Crowded Spaces. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; IEEE: Las Vegas, NV, USA, 2016; pp. 961–971. [CrossRef]

14. Zhang, P.; Ouyang, W.; Zhang, P.; Xue, J.; Zheng, N. SR-LSTM: State Refinement for LSTM towards Pedestrian Trajectory Prediction. *arXiv* **2019**, arXiv:1903.02793.
15. Gupta, A.; Johnson, J.; Fei-Fei, L.; Savarese, S.; Alahi, A. Social GAN: Socially Acceptable Trajectories with Generative Adversarial Networks. arXiv 2018, arXiv:1803.10892.
16. Hasan, I.; Setti, F.; Tsesmelis, T.; Del Bue, A.; Galasso, F.; Cristani, M. MX-LSTM: Mixing Tracklets and Vislets to Jointly Forecast Trajectories and Head Poses. arXiv 2018, arXiv:1805.00652.
17. Lee, N.; Choi, W.; Vernaza, P.; Choy, C.B.; Torr, P.H.S.; Chandraker, M. DESIRE: Distant Future Prediction in Dynamic Scenes with Interacting Agents. arXiv 2017, arXiv:1704.04394.
18. Yu, J.; Zhou, M.; Wang, X.; Pu, G.; Cheng, C.; Chen, B. A Dynamic and Static Context-Aware Attention Network for Trajectory Prediction. ISPRS Int. J. Geo-Inf. 2021, 10, 336. [CrossRef]
19. Yang, T.; Nan, Z.; Zhang, H.; Chen, S.; Zheng, N. Traffic Agent Trajectory Prediction Using Social Convolution and Attention Mechanism. arXiv 2020, arXiv:2007.02515.
20. Li, X.; Ying, X.; Chuah, M.C. GRIP++: Enhanced Graph-Based Interaction-Aware Trajectory Prediction for Autonomous Driving. arXiv 2020, arXiv:1907.07792.
21. Yu, B.; Yin, H.; Zhu, Z. Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting. arXiv 2018, arXiv:1709.04875.
22. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. arXiv 2019, arXiv:1810.04805.
23. Li, X.; Feng, J.; Meng, Y.; Han, Q.; Wu, F.; Li, J. A Unified MRC Framework for Named Entity Recognition. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online. 5–10 July 2020; pp. 5849–5859. [CrossRef]
24. Yamada, I.; Asai, A.; Shindo, H.; Takeda, H.; Matsumoto, Y. LUKE: Deep Contextualized Entity Representations with Entity-Aware Self-Attention. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online. 5–10 July 2020; pp. 6442–6454. [CrossRef]
25. Gu, J.; Bradbury, J.; Xiong, C.; Li, V.O.K.; Socher, R. Non-Autoregressive Neural Machine Translation. arXiv 2018, arXiv:1711.02281.
26. Meng, Y.; Zhang, Y.; Huang, J.; Xiong, C.; Ji, H.; Zhang, C.; Han, J. Text Classification Using Label Names Only: A Language Model Self-Training Approach. arXiv 2020, arXiv:2010.07245.
27. Althoff, M.; Mergel, A. Comparison of Markov Chain Abstraction and Monte Carlo Simulation for the Safety Assessment of Autonomous Cars. IEEE Trans. Intell. Transp. Syst. 2011, 12, 1237–1247. [CrossRef]
28. Hillenbrand, J.; Spieker, A.M.; Kroschel, K. A Multilevel Collision Mitigation Approach—Its Situation Assessment, Decision Making, and Performance Tradeoffs. IEEE Trans. Intell. Transp. Syst. 2006, 7, 528–540. [CrossRef]
29. Polychronopoulos, A.; Tsogas, M.; Amditis, A.J.; Andreone, L. Sensor Fusion for Predicting Vehicles’ Path for Collision Avoidance Systems. IEEE Trans. Intell. Transp. Syst. 2007, 8, 549–562. [CrossRef]
30. Messaoud, K.; Yahiaoui, I.; Verroust-Blondet, A.; Nashashibi, F. Attention Based Vehicle Trajectory Prediction. IEEE Trans. Intell. Veh. 2021, 6, 175–185. [CrossRef]
31. Kim, H.; Kim, D.; Kim, G.; Cho, J.; Huh, K. Multi-Head Attention Based Probabilistic Vehicle Trajectory Prediction. arXiv 2020, arXiv:2004.03842.
32. Peng, Y.; Zhang, G.; Shi, J.; Xu, B.; Zheng, L. SRAI-LSTM: A Social Relation Attention-based Interaction-aware LSTM for human trajectory prediction. Neurocomputing 2021. [CrossRef]
33. Yuan, Y.; Weng, X.; Ou, Y.; Kitani, K. AgentFormer: Agent-Aware Transformers for Socio-Temporal Multi-Agent Forecasting. arXiv 2021, arXiv:2103.14023.
34. Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; Lerer, A. Automatic Differentiation in PyTorch. Available online: https://openreview.net/forum?id=BJsrmiCZ (accessed on 30 October 2021).
35. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. arXiv 2017, arXiv:1412.6980.