IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 12 September 2022, accepted 2 October 2022, date of publication 10 October 2022, date of current version 11 November 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3213691

RESEARCH ARTICLE

Space-Time Tree Search for Long-Term Trajectory Prediction

TINGYONG WU[1], (Member, IEEE), PEIZHI LEI[1], (Student Member, IEEE), FUQIANG LI[2], AND JIENAN CHEN[1], (Senior Member, IEEE)

[1]National Key Laboratory of Science and Technology on Communications, University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, China
[2]Key Laboratory of Technology on Datalink, China Electronics Technology Group Corporation (CETC), 20th Institute, Xi'an, Shanxi 710068, China

Corresponding author: Fuqiang Li (lfqnet@qq.com)

This work was supported in part by the China Electronics Technology Group Corporation (CETC) Key Laboratory of Data Link Technology under Grant CLDL-20202415, and in part by the Sichuan Science and Technology Program under Grant 2021YFG0127.

ABSTRACT The pedestrian trajectory prediction is critical for autonomous driving, intelligent navigation, and abnormal behavior detection. With the booming of artificial intelligence (AI), many researchers have employed deep learning technologies to solve the pedestrian trajectory prediction problem and obtained relatively better performance in the short-term trajectory prediction. However, long-term trajectory prediction is still challenging to achieve high prediction accuracy. In this work, we propose a space-time tree search (STTS) method for long-term pedestrian trajectory prediction. Compared with existing methods only considering the problem from the space dimension, the proposed method formulates the trajectory prediction problem as a joint space-time tree search process by mapping the environment to a grid map. Since the human's trajectory is relative to space and time dimensions, the trajectory prediction accuracy can be improved by the two dimensions. Then, a space-time reward trained neural network is employed to extract the pedestrian's intent with both the scene image and the historical trajectory as input and outputs the prior search probabilities. Finally, the tree search can obtain the optimal predicted trajectory according to the prior probabilities, significantly improving the tree search efficiency. After testing, our proposed method can perform better than existing methods.

INDEX TERMS Long-term trajectory prediction, space-time reward, tree search, neural network.

I. INTRODUCTION

The goal of pedestrian trajectory prediction is to predict human trajectories by analyzing the historical trajectories. The trajectory prediction has been widely applied in autonomous driving [1], [2], [3], intelligent navigation [4], [5], [6], maritime Internet of Things [7], [8], abnormal behavior detection et. al [9], [10], [11], [12]. For example, in autonomous driving, the surrounding vehicles' or pedestrians' future trajectory predictions can help the autonomous driving controller avoid potential risks. Besides, in maritime Internet of Things, trajectory prediction can improve smart traffic service for ships.

The pedestrian's trajectory can be influenced by multiple factors: the surrounding environment, the pedestrian's intent, and so on. Many researchers have optimized their methods from these factors. However, in a crowded environment, the surrounding environment will constantly change as the surrounding pedestrians walk. It's hard to model the surrounding environment for the trajectory prediction. Besides, in the long-term trajectory prediction, the pedestrian's long-term intent is hard to be extracted, which makes the prediction error relatively large.

The existing pedestrian trajectory prediction methods can be categorized into two types: the traditional model-based methods and deep learning methods. The traditional model-based methods mainly rely on manually designed human behavior models. They need hand-crafted settings for the pedestrian's walking properties. Hence, many traditional

The associate editor coordinating the review of this manuscript and approving it for publication was Jason Gu.

model-based methods for trajectory prediction use the social force [13], [14], [15], [16], which considers the pedestrian behaviour as the co-operation of several forces. Furthermore, some bayesian model-based methods have also been proposed for trajectory prediction [17], [18], [19], [20]. However, the traditional model-based methods generally can't work well in the crowded scenes because the complex environmental information is challenging to be fuzed into one model manually. In the last few years, with the development of deep learning, the long short-term memory (LSTM) has been widely applied to the trajectory prediction [21], [22], [23], [24], which can learn the human intent by human past trajectories instead of manually designing the human behavior model. Moreover, paper [25] proposes a model combining sequence prediction and generative adversarial networks, which can predict more reliable trajectories.

The methods mentioned above suffer relatively low accuracy in the long-term trajectory prediction. Due to the only consideration of the pedestrian's historical trajectory, existing methods pay little attention to the space information. Existing deep learning methods only considering minimizing the space distances between the predicted trajectory and the ground truth trajectory. Thus, they ignore the significant scene information for long-term trajectory prediction. However, for the long-term trajectory prediction, the pedestrian's long-term intent should be extracted by both the surrounding environment and the historical trajectory. As the environment contains high spacial information of the complex surroundings. Different from the existing methods, our proposed method can minimize both the space and time distance to generate a more reliable trajectory prediction.

In this paper, we propose a space-time tree search (STTS) method to generate the future trajectory in the scenario of long-term trajectory prediction. Different from existing methods only considering the problem from the space dimension, we convert the trajectory prediction problem to a joint space-time tree search process. Hence, we can improve the trajectory prediction accuracy from the two dimensions. Then, a space-time reward trained neural network is employed to extract the pedestrian's intent with both the scene image and the historical trajectory as input and outputs the prior search probabilities. Finally, the tree search is employed to obtain the optimal predicted trajectory by the outputted prior probabilities, which can significantly promote the tree search efficiency. The contributions of our proposed method are summarized as follows.

A. THE PEDESTRIAN's INTENT PREDICTION

The pedestrian's intent is of great importance in the long-term trajectory prediction. Different from existing methods only considering the scene information or the historical trajectory, we employ a neural network to learn the pedestrian's intent with both the scene image and the historical trajectory as input. Then, the neural network will output the prior search probabilities to guide the tree search process and can be trained by the space and time reward.

B. THE TREE SEARCH PROCESS

The environment is converted to a grid map by the superpixel segmentation network [26] and the scene parsing network [27]. Meanwhile, the historical trajectory is employed to improve the performance of converting. After converting the environment to a grid map, the trajectory prediction problem is transformed into a grid search game. Hence, we can use the tree search to find the optimal trajectory prediction according to the prior probabilities outputted by a space-time trained neural network. The prior probabilities outputted by the neural network can guide the tree search to explore the grid point with high prior probability. Hence, the search efficiency can be significantly improved.

C. THE SPACE AND TIME REWARD

The space and time reward is calculated at each predicted grid point in the grid map. Especially, the space reward is relative to the minimum distance between the currently predicted grid point and all the grid points in the ground truth trajectory. The time reward is relative to the distance between the currently predicted grid point and the grid point in the ground truth trajectory at the same time. Hence, the space and time reward can make the predicted trajectory more similar to the real trajectory.

After testing, in the datasets: UCLA, hyang, and coupa, the performance of our proposed method improves 21.22%, 25.21%, and 18.39%, respectively, compared with the ''dark matter'' method [28]. The rest of the paper is organized as follows. We first introduce the related works of trajectory prediction in Section II. Then, in Section III, the space time tree search process will be illustrated detailedly. What's more, the experimental results will be shown in Section IV. Finally, we conclude our work in Section V.

II. RELATED WORK

A. TRADITIONAL MODEL-BASED METHODS FOR TRAJECTORY PREDICTION

Social force is firstly proposed in paper [13], which considers the pedestrian behavior as the co-operation of three main forces: acceleration towards the desired velocity of motion, repulsive force and attractive force. Later, Mehran et al. [15] propose that the pedestrian's interaction force can be estimated based on the social force model. A discrete choice model of pedestrian behavior is proposed in [29], which uses a dynamic and individual-based spatial discretization to represent the physical space. Furthermore, the Bayesian model [19], [20] has also been proposed to model activities and interactions in crowded and complicated scenes for trajectory prediction. Recently, Xie et al. [28] present a method for predicting human intents and trajectories in surveillance videos of public spaces by using ''dark matter'', which can only be observed as an attractive or repulsive ''field'' in the public space.

Besides, some agent-based methods [30], [31] have also been proposed for the trajectory prediction. Paper [32]

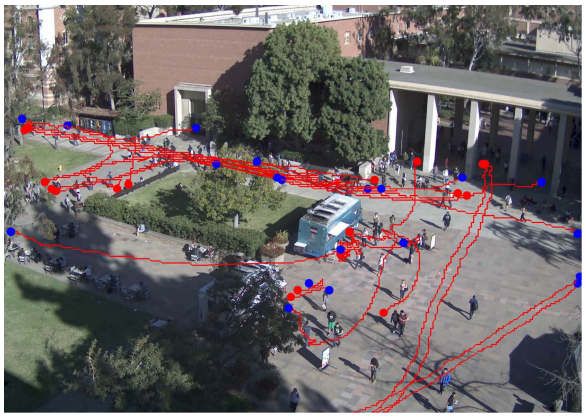

FIGURE 1. The pedestrians' trajectories. The red circle denotes the start point and the blue circle denotes the end point(destination).

proposed Linear Trajectory Avoidance (LTA) model for the short-term trajectory prediction, inspired by models developed for crowd simulation. A novel model for pedestrian behavior modeling is proposed in [33], which includes stationary crowd groups as a key component. It can also be employed to investigate pedestrian behaviors by inferring interactions between stationary crowd groups and pedestrians. Another method proposed in [34] models the joint distribution over future trajectories of all interacting agents in the crowd by a real human trajectory data trained interaction model, which can infer the velocity of each agent. Then, the proposed method can infer the goal of the agent from its past trajectory and use the learned model to predict its future trajectory.

The main disadvantage of traditional model-based methods is that the output accuracy depends very much on manual feature engineering, such as walking directions, walking velocity, interactive action among persons, etc. When we face a crowded environment, these features are very complex to model. Moreover, when we tackle with long-term trajectory prediction tasks, predicting the motions of human targets is extremely challenging.

B. DEEP LEARNING METHODS FOR TRAJECTORY PREDICTION

With the development of AI, deep learning has been widely applied in our daily life [35], [36], [37], [38], [39], [40]. Especially, recurrent neural network (RNN) is widely used to analyze the structure of the time-series data. One popular variant of RNN is the long short-term memory (LSTM) model [41]. The LSTM has shown excellent performance for various tasks such as speech recognition [42], [43], image captioning [44], [45], language translation [46], [47] and so on.

The LSTM has also been applied to analyze the pedestrian trajectory prediction task because the pedestrian trajectory can also be considered time-series data. Compared with traditional model-based methods, deep learning based methods can learn human intent by historical trajectory data without manually designing a human behavior model. Alahi et al. [21] proposed one Social-LSTM model to generate the future trajectory by analyzing the past trajectory, which doesn't incorporate the scene information and will be unreasonable to generate trajectory. A DESIRE (Deep Stochastic Inverse optimal control RNN Encoder-decoder) model is proposed in [48] for trajectory prediction, which is realized by a single end-to-end trainable network. It can effectively predict future trajectory in multiple scenes by using an RNN scene context fusion module to jointly capture past motion histories, the semantic scene context and interactions among multiple agents. Xue et al. [23] use the SS-LSTM network to model the scene information and the past movement trajectories of pedestrians, which generates better results for future trajectory. Zhang et al. [24] propose a data-driven state refinement module for the LSTM network (SR-LSTM), which can activate the utilization of the current intention of neighbors. Besides, a model combining sequence prediction and generative adversarial networks is proposed in [25] for predicting more reliable trajectories. However, the LSTM-based methods do not characterize the relationship between the past trajectories and the current intents of a human at a specific area of a scene, that the error will be accumulated in the long term trajectory prediction.

III. SPACE-TIME TREE SEARCH

In this work, we first convert the scene image into a grid map by superpixel segmentation, scene parsing network, and walkable region correction. Then, we divide the scene image into many small grids, label the grids to a walkable or non-walkable region and correct the errors in the labeling process, respectively. After that, we use neural networks to extract human behavior and surrounding environment information. Authors of [49] proposed a model which utilizes LSTM to extract the multiple time-scale information contained in the convolutional layers in CNN. The model performed well on hotspots prediction tasks. Motivated by [49], we employ an LSTM to output the prior search probabilities with the extracted information from CNN as input. Finally, the tree search will generate the optimal trajectory according to the prior probabilities. A feature of the proposed method is considering the trajectory problem from space and time dimensions. Hence, we use a space-time reward to train the neural networks in the training process.

A. PROBLEM FORMULATION

Assume the coordinate of pedestrian at time t is (x_t, y_t), we can transform the coordinate to a grid coordinate g^t_{mn} by the superpixel network. Then, the trajectory prediction problem is converted into a sequence grid coordinates prediction problem from time $t+1$ to T with the historical trajectory $\{g^1_{mn}, g^2_{mn}, \ldots, g^t_{mn}\}$, where T is the terminal time. Here, the predicted trajectory is denoted as $D_{pre} = \{g^{t+1}_{mn}, g^{t+2}_{mn}, \ldots, g^T_{mn}\}$ and the corresponding ground truth trajectory is denote $U = \{(g^{t+1}_{mn})_U, (g^{t+2}_{mn})_U, \ldots, (g^T_{mn})_U\}$.

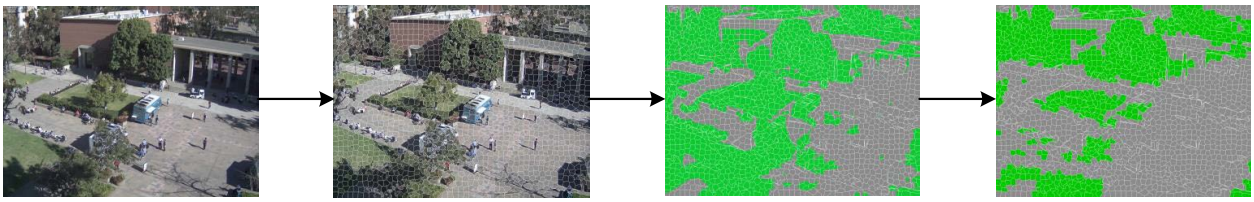

FIGURE 2. (a) The original image. (b) The superpixel image. (c) The superpixel segmented image. The gray pixel is walkable. The green pixel is non-walkable. (d) The corrected superpixel segmented image. The gray pixel is walkable. The green pixel is non-walkable.

Hence, the optimization objective is to minimize the distance between the predicted trajectory and the ground truth trajectory from both the space and time dimensions as

$$Dis = \sum_{i=t+1}^{T} \left\| g^i_{mn} - (g^i_{mn})_U \right\|_2 \tag{1}$$

The part pedestrians' trajectories are shown in Fig. 1, where the red circles are the start points, and the blue circles are the endpoints.

B. GRID-LIKE MAP

We first convert the scenario image to a grid map by superpixel segmentation network, scene parsing network and walkable region correction shown in Fig. 2. We use the superpixel segmentation network to divide the scenario image into grids of different shapes. Then, the scene parsing network is employed to label the grids into the walkable or non-walkable region. Finally, we use the historical trajectories to correct the errors of walkable and non-walkable labels. The details of the three processes are summarized as follows.

1) SUPERPIXEL SEGMENTATION NETWORK

The superpixel segmentation network is based on the method of [26], which can generate a segmented image with grids of different shapes. It is realized by the k-means clustering algorithm, where the distance measure is denoted as

$$D = \sqrt{d_c^2 + \left(\frac{d_s}{S}\right)^2 m^2}. \tag{2}$$

The color similarity is measured by d_c, and the spatial proximity is denoted as d_s. The grid interval is S, and m weighs the relative importance between color similarity and spatial proximity. In this paper, we employ the superpixel segmentation network to divide the scene image into many small grids shown in Fig. 2(b). Hence, each trajectory point (x_t, y_t) of the pedestrian can be corresponding to one grid location g^t_{mn}.

2) SCENE PARSING NETWORK

In the process of pedestrian trajectory prediction, the pedestrian will walk in the walkable region. Hence, we should obtain the walkable region of the scene image. In this paper, the Pyramid Scene Parsing Network (PSPNet) [27] is employed to obtain the category label of each pixel in the scene image. Generally, the output of the PSPNet has hundreds of different category labels. However, in this proposed scheme, we only need two categories: walkable and non-walkable. Therefore, we classify roads and trails into a walkable category. The tree and building are classified into a non-walkable category shown in Fig. 2(c). The gray segment is the walkable region. The superpixel segmentation network is based on the method of [26], which can generate a segmented image with grids of different shapes. Then, the trajectory information can be mapped to discrete grid locations. The segmented image is shown in Fig. 2(c). Finally, we calculate the number of walkable and non-walkable pixels. If the number of walkable pixels is bigger than the number of non-walkable pixels, the label of this grid $g_{m,n} \cdot C$ is 1 (walkable). Otherwise, the label is 0 (non-walkable).

3) WALKABLE REGION CORRECTION

The problems of shadow and angle will result in the errors of walkable and non-walkable labels. Hence, we use historical trajectories to correct the errors. We will calculate the frequency of the grids in the non-walkable area people entered according to the humans' historical trajectories. If the frequency of the grid is larger than a threshold τ, the label of the grid will be corrected to 1(walkable). The correction performance is shown in Fig. 2(d).

Finally, we calculate the number of walkable and no-walkable pixel. If the number of walkable pixel C_W in each gird is bigger than the number of non-walkable pixel C_N, the label of this grid $l_{g_{mn}}$ is set to 1 (walkable). Otherwise, the label $l_{g_{mn}}$ is set to 0 (non-walkable). The process can be represented by

$$l_{g_{mn}} = \begin{cases} 1, & if\ C_W \geq C_N \\ 0, & if\ C_W < C_N \end{cases} \tag{3}$$

Besides, our whole grid map is denoted as $\mathcal{I}^{M\times N}$. The grid location g^t_{mn} is the valid position of $\mathcal{I}^{M\times N}$ as $g^t_{mn} \in \mathcal{I}^{M\times N}$.

Each pedestrian has following attributions, the current and historic trajectory as

$$H_t = \{g^1_{mn}, g^2_{mn}, \ldots, g^t_{mn}\}, \tag{4}$$

where t is the time index. The conditional probability of next moving is $p(g^{t+1}_{mn}|g^t{}_{mn})$, where g^{t+1}_{mn} is the adjacent grid of $g^t_{m,n}$. The grid location of pedestrian's destination is denoted

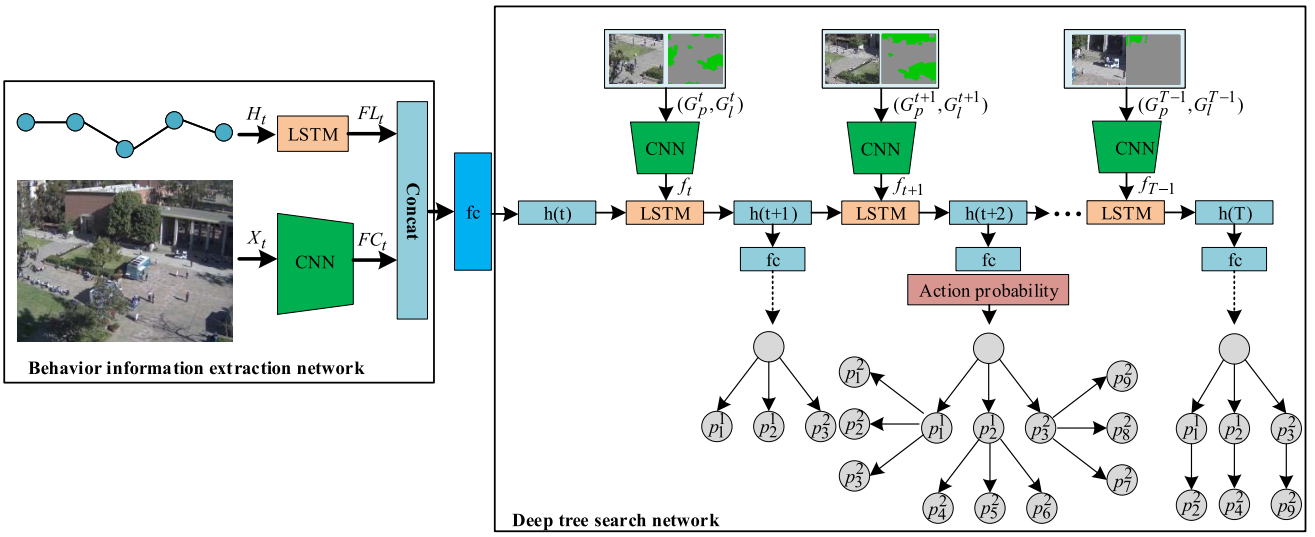

FIGURE 3. The whole framework of deep tree search-based pedestrian trajectory prediction. The framework mainly contains a human behavior module and a deep tree search network.

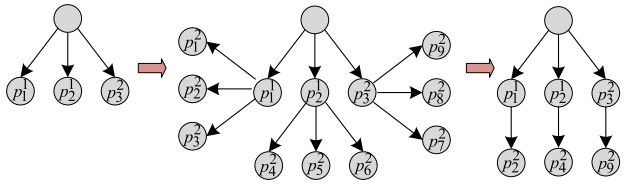

FIGURE 4. The tree search process.

as $g_{m,n}^T$, where T is the search terminal time. The prediction of possible trajectory is denoted as

$$D_{pre} = \{g_{mn}^{t+1}, g_{mn}^{t+2}, \ldots, g_{mn}^T\} \tag{5}$$

C. THE WHOLE FRAMEWORK OF PROPOSED METHOD

In the real environment, the pedestrians will plan their trajectories according to the destination. We propose the space-time tree search model to generate a longer trajectory shown in Fig. 3. The space-time tree search model contains the human behavior module and the deep tree search network. The inputs of the behavior information extraction network are the historical trajectory H_t and the scene image X_t, which are processed by the LSTM and CNN, respectively. Then, the extracted features FL_t and FC_t will be concatenated as the initial state of the LSTM in the deep tree search network. Finally, the deep tree search network can generate the trajectory from (x_{t+1}, y_{t+1}) to (x_T, y_T). In the deep tree search network, the neural network can output the prior probabilities for the tree search. Hence, the tree search efficiency can be greatly promoted. In the next subsection, we will introduce the tree search process detailedly.

D. THE TREE SEARCH PROCESS

At time t, the position of pedestrians (x_t, y_t) is the start position of our search algorithm. The possible destination position is denoted as (x_T, y_T). Meanwhile, we use a grid point to represent each position. The trajectory point (x_t, y_t) is corresponding to the grid location g_{mn}^t. Then the future trajectory route can be predicted by predicting the future grid points. Each grid point g_{mn}^t has eight adjacent grid points g_{mn}^{t+1}. Hence, we define the transition probability of each grid point at the next time as

$$p\left(g_{mn}^{t+1}|g_{mn}^t\right) = \frac{\left\|g_{mn}^t - g_{mn}^T\right\|^2}{\left\|g_{mn}^{t+1} - g_{mn}^T\right\|^2} * p_l\left(g_{mn}^{t+1}|g_{mn}^t\right) \tag{6}$$

where $p_l(g_{mn}^{t+1}|g_{mn}^t)$ represents the transition probability from grid point g_{mn}^t to grid point g_{mn}^{t+1} and is an element of the probability vector $\vec{p}_l(g_{mn}^t)$ with a size 1×8 outputted by the neural network. Besides, the pedestrians are more likely to walk in the direction closer to the destination. Hence, in the first term in the equation (6), the numerator represents the Euclidean Distance from the current grid point to the

Algorithm 1 The Space-Time Tree Search

1: Calculate the top-K prior probability vector as $\vec{P} = \{P_1, P_2, \ldots, P_K\}$ according to equation (6)
2: **while** not reach the destination **do**
3: Initialize an empty list L_p
4: **for** $i = 1, 2, ..K$ **do**
5: The neural network output the probability vector $\vec{p}_l(g_{mn}^t)$ of grid point corresponding to P_i
6: Calculate $\vec{p}(g_{mn}^t)$ according to equation (6)
7: The top K probability values of $P_i * \vec{p}(g_{mn}^t)$ insert the list L_p
8: Expand the search tree with the top K probabilities in L_p
9: $\vec{P} \leftarrow$ the top K probabilities in L_p

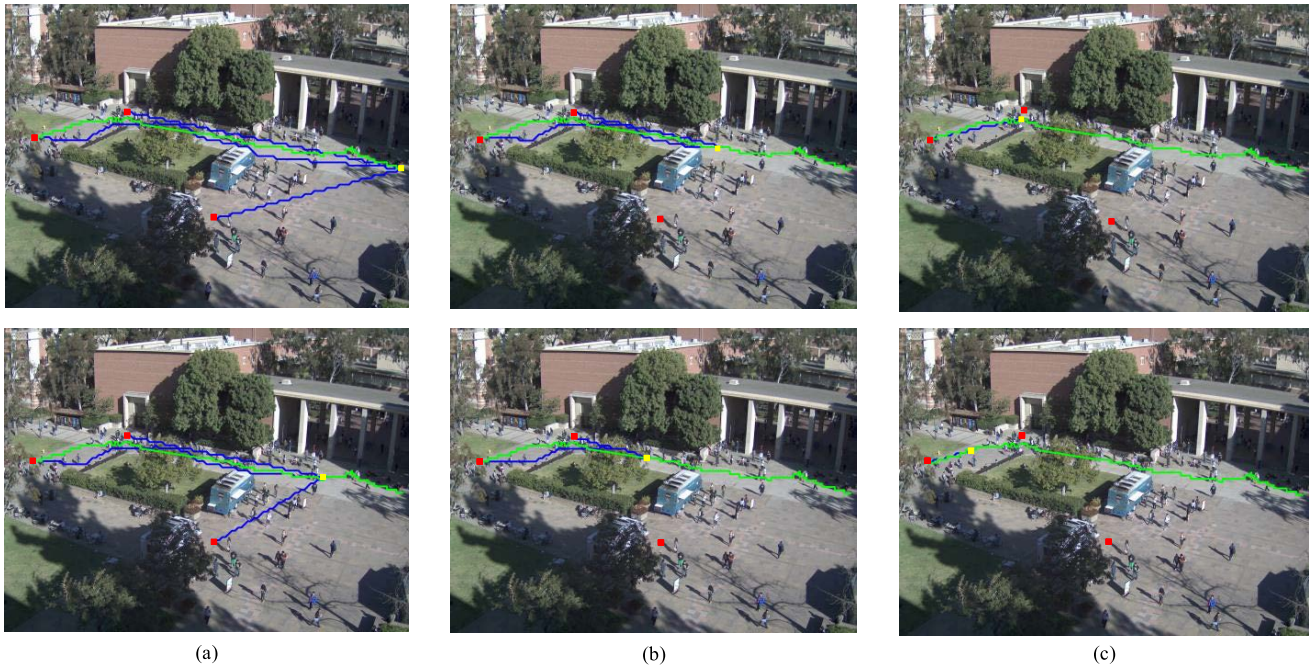

FIGURE 5. The predicted multiple possible trajectories at different start points. As the pedestrian walks, the number of predicted possible trajectories will decrease to 1.

destination grid point. And the denominator represents the Euclidean Distance from the next possible grid point to the destination grid point. Then, suppose the next possible grid point is closer to the destination grid point. In that case, its transition probability will be relatively higher. Finally, for the eight transition probabilities, we choose the grid points corresponding to the largest K probabilities as the possible moving positions at the next time. On the other hand, although g_{mn}^{t} has eight adjacent grid points $g_{m,n}^{t+1}$, we get from the previous section that K possible grid points must meet the following conditions: $l_{g_{mn}^{t+1}} = 1$, which means the grid point must be walkable.

As shown in Fig. 4, we take $K = 3$ as an example and get the corresponding probability values, which are denoted as p_1^1, p_2^1, p_3^1. Then at time $t+1$, we need to predict the probability values of three new grid points according to equation (6). Similarly, each grid point selects three adjacent grid points with the largest probability value, as shown in the figure, we get nine probability values $p_1^2, \ldots, p_9^2$. For boosting the search speed, we calculate the joint probability distribution of the obtained probability values, which are $P_1 * p_1^2, P_1 * p_2^2, \ldots, P_2 * p_4^2, \ldots, P_3 * p_9^2$, where $P_1 = p_1^1, P_2 = p_2^1, P_3 = p_3^1$. Then we select the three grid points corresponding to the maximum three joint probabilities of the nine as the final three estimated trajectory points at time $t+1$. We finally select p_2^2, p_4^2, p_9^2. Then the prior probabilities can be updated to $P_1 = p_1^1 * p_2^2, P_2 = p_2^1 * p_4^2, P_3 = p_3^1 * p_9^2$. Repeat the above steps until you reach the destination grid point. The whole process is summarized as Algorithm 1.

Algorithm 2 Training the Neural Networks

Input: The trajectory set Ω, the scene image X
Output: The trained parameters of the networks θ

Initialize $\theta_{hbm}^{L}, \theta_{hbm}^{C}, \theta_{h}, \theta_{enc}^{C}, \theta_{gen}^{L}, \theta_{p}$ randomly
while within max predicting steps **do**
  Sample a trajectory $U(t)$ from trajectory set Ω
  $FL_t = LSTM_{hbm}\left(H_t; \theta_{hbm}^{L}\right)$
  $FC_t = f_{hbm}^{C}\left(X_t; \theta_{hbm}^{C}\right)$
  $F_t = concat(FL_t, FC_t)$
  $h_t = f_1\left(F_t; \theta_h\right)$
  $f_t = f_{enc}^{C}\left(G_p^t, G_l^t; \theta_{enc}^{C}\right)$
  $p_l\left(g_{mn}^{t}\right) = f_2\left(LSTM_{gen}\left(f_t, h_t; \theta_{gen}^{L}\right); \theta_p\right)$
  Expand the search tree according to $p_l\left(g_{mn}^{t}\right)$
  Repeat step 4-10 until reaching the destination
  Obtain the prediction trajectory D_{pre}
  Calculate the space-time reward R
  Update the parameters of the networks by
  $\theta \leftarrow \theta + \alpha \nabla_\theta \log p_l\left(g_{mn}^{t}\right) R$

E. DESTINATION SELECTION

In the previous subsection, we need to calculate the search probability $p\left(g_{mn}^{t+1} | g_{mn}^{t}\right)$, which is relative to the trajectory's destination. The destination selection contains three steps. Firstly, we will analyze the start points and end points of the historical trajectories. For each start point P_s, we will build a memory M_{P_s} for its possible end points as

$$M_{P_s} = \{(P_e^i, O_e^i)\} \tag{7}$$

where P_e^i is one of the P_s's corresponding end points and O_e^i denotes the occurrence frequency of P_e. Then, when predicting a trajectory, we will obtain all the possible end points corresponding to the start point in the range, which is a circle of radius R centered at the start point. Then, we will select N possible end points, which have the top-N occurrence frequencies as the possible destinations. Finally, we will generate N possible trajectories by the tree search in the start point P_s. The tree search process generates the N possible trajectories independently according to the corresponding destinations. Besides, as the pedestrian walks, the N possible trajectories will be gradually corrected. Some possible trajectories will be abandoned until one trajectory is left because their end points are in the opposite direction of the pedestrian walking. As shown in Fig. 5, from top to bottom and from left to right, at the initial start point, the number of predicted possible trajectories is 3. Then, as the trajectory is searched, the number of predicted possible trajectories will decrease to 1.

F. NETWORK TRAINING

1) REWARD DEFINITION

The reward contains two parts: space reward and time reward. The space reward is denoted as the difference between predicted and true trajectory in walking space. The time reward represents the difference between predicted and true trajectory in walking time.

Consequently, the normalized game reward in terms of space dimension is given by

$$R^s = \frac{\sum\limits_{t \to T} \|U(t+1) - U(t)\|^2}{\sum\limits_{t \to T} F_s\left(D_{pre}(t) - U\right)} \tag{8}$$

where $U(t)$ is grid location of the ground truth trajectory at time t. The denominator of equation (8) is the difference of prediction between ground truth trajectory in l2-norm and the numerator is the distance of ground truth trajectory. Considering the different walking speed among each agent, the strict trajectory synchronization is not required. We employ a function F_s to calculate the trajectory distance by find the minimal distance from ground truth trajectory U as

$$F_s\left(D_{pre}(t) - U\right) = \min_{o \to T} \left\|D_{pre}(t) - U(o)\right\|^2 \tag{9}$$

Hence, we can introduce the time-reward as

$$R^t = \frac{\sum\limits_{t \to T} \|U(t+1) - U(t)\|^2}{\sum\limits_{t \to T} \left\|D_{pre}(t) - U(t)\right\|^2} \tag{10}$$

Consequently, the space-time reward is given by

$$R = \beta R^s + (1-\beta)R^t \quad (0 < \beta < 1) \tag{11}$$

where β is a regulator factor to balance the reward of space and time.

TABLE 1. The network structure of CNN.

CNN		
layer name	output size	layers
conv1	112×112	7×7, 64, stride 2
conv2_x	55×55	3×3 max pool, stride 2 $\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$
conv3_x	27×27	2×2 max pool, stride 2 $\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 3$
conv4_x	13×13	2×2 max pool, stride 2 $\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 3$
conv5_x	6×6	2×2 max pool, stride 2 $\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$
fc	512	6×6 max pool, stride 1

2) TRAINING DETAILS

We need to consider the influence of historical trajectory information and scene information simultaneously when selecting the probability of grid points the next time. Hence, in this section, we mainly discuss the specific structure of the neural network. Firstly, we use a neural network to extract the whole scene and historical track information. Then, we use the extracted feature vector as the initial hidden state of the LSTM network. Meanwhile, we map the output of the scene LSTM unit to the selection probability of current grid point $p_l(g_{mn}^{t+1}|g_{mn}^t)$ by the fully connected layer. The CNN is based on Resnet [50], which contains 6 parts. The size of input image is $244 \times 244 \times 3$. The first part is a convolution layer with a 7×7 kernel. The 2-5 parts all consist of a max-pooling layer and three residual blocks. The sixth part is a 6×6 max-pooling layer. The final output is a vector with 512 dimensions. The total amount of parameters in CNN is 48142.

Scene information and historical trajectory contain the basic information needed for trajectory prediction. At time t, the pedestrian's trajectory is H_t. In order to obtain the continuous trajectory information, we use a LSTM in the human beaviour module(HBM) to extract observed information before time t. Then the hidden state FL_t contained in the trajectory information of the pedestrian at time t can be expressed as

$$FL_t = LSTM_{hbm}\left(H_t; \theta_{hbm}^L\right) \tag{12}$$

where $LSTM_{hbm}$ represents the encoded LSTM network for pedestrian trajectory extraction. The extracted hidden state of pedestrian's trajectory is denoted as FL_t. θ_{hbm}^L is weight matrix of LSTM. Meanwhile, for extracting the hidden information of the scene information, we train a HBM CNN network to get the feature vector of the scene image FC_t as

$$FC_t = f_{hbm}^C(X_t; \theta_{hbm}^C), \tag{13}$$

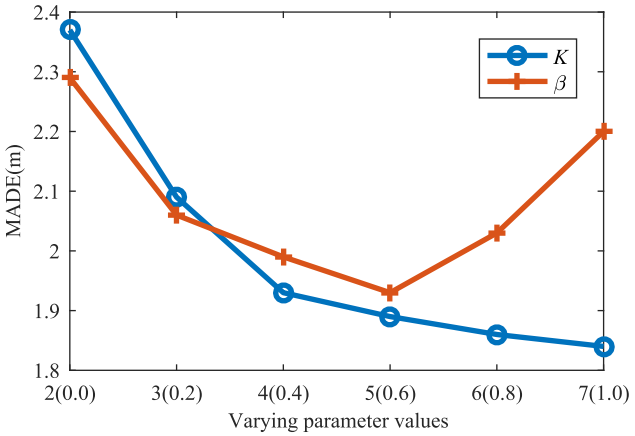

FIGURE 6. The MADEs of our proposed method under varying parameter values in the dataset: UCLA. The blue line is under parameter K, which is set to {2, 3, 4, 5, 6, 7}. The red line is under parameter β, which is set to {0, 0.2, 0.4, 0.6, 0.8, 1.0}.

where θ^C_{hbm} is the corresponding weight matrix and X_t is the scene image. After getting the historical trajectory information FL_t and scene feature information FC_t at time t, we linearly concatenate the two features into F_t as

$$F_t = \{FL_t, FC_t\}. \tag{14}$$

where, FL_t is a $1 \times L$ vector and FC_t is a $1 \times C$ vector, thus F_t is a $1 \times (L + C)$ vector. Then, we use a fully connected layer to get the initial hidden state $h_t = f_1(F_t; \theta_h)$ for the selection probability generation LSTM($LSTM_{gen}$). Through the linear concatenation process and nonlinear activation of deep learning models, we can obtain a hidden vector containing both trajectory and space features in a scene.

After getting the hidden unit, we need to decode the h_0 to get the predicted trajectory point. At time t, we obtain the local grid image information G^t_p centered on g^t_{mn} and the corresponding label image information G^t_l. Then we extract a feature vector value f_t through an encoder CNN network as

$$f_t = f^C_{enc}(G^t_p, G^t_l; \theta^C_{enc}), \tag{15}$$

where θ^C_{enc} is the weight of CNN. Then we will take f_t as the input of $LSTM_{gen}$ and h_t as the hidden unit input at the current time. The hidden state value h_{t+1} of the next time is obtained by $LSTM_{gen}$. In order to obtain the state selection probability at time $t + 1$, we use a fully connected network to map the hidden state h_{t+1} to action select probability $p_l(g^t_{mn})$. The weight matrix of $LSTM_{gen}$ is denoted as θ^L_{gen}. Finally, the prior probability provided by the neural network in the current state can be obtained as

$$\vec{p}_l(g^t_{mn}) = f_2\left(LSTM_{gen}\left(f_t, h_t; \theta^L_{gen}\right); \theta_p\right) \tag{16}$$

Through the search algorithm, we can select the optimal trajectory prediction according to the prior probability. Then we can update the weights θ of the whole neural network as the policy gradient [51], which is given by

$$\theta \leftarrow \theta + \alpha \nabla_\theta \log p_l\left(g^{t+1}_{mn} | g^t_{mn}\right) R \tag{17}$$

TABLE 2. The performance comparison with serval existing method. Our proposed method has the best performance in all the three datasets.

Method	MADE(m)		
	UCLA	hyang	coupa
Social LSTM [21]	3.01	2.98	2.86
Dark Matter [28]	2.45	2.38	2.23
Social GAN [25]	2.86	2.65	2.82
SR-LSTM [24]	2.63	2.49	2.71
Our model	**1.93**	**1.78**	**1.82**

The training process is summarized in Algorithm 2. Firstly, the scene information and the historical trajectory are inputted into the neural network to get the prior search probabilities. The search tree will be expanded. This process will be executed repeatedly until we reach the destination. Then we calculate the reward according to the predicted and ground truth trajectory. Finally, the whole neural network can be updated according to the equation (18).

When the predicted grid point reaches or near the target point, we stop the search and assume that the final output of K prediction trajectories is $a_i \cdot D_k, k = 1, 2, .., K$. Hence, we can build the loss function as

$$L = \sum_{k=1}^{K} \left\| \sum_{t \to T} F_s \left(a_i \cdot D_k(t) - a_i \cdot U\right) \right\|^2 + \alpha \left\|\theta^C_{hbm}\right\| + \beta \left\|\theta^L_{hbm}\right\| + \gamma \left\|\theta^C_{enc}\right\| + \delta \left\|\theta^L_{gen}\right\| \tag{18}$$

where $\alpha, \beta, \gamma, \delta$ are the coefficients of regularization term. F_s is the distance difference between each predicted trajectory point and the nearest real trajectory point. $a_i \cdot U$ is the true trajectory.

IV. EXPERIMENT

In this section, we conduct experimental studies to evaluate the performance of the proposed method by comparing it with existing Social LSTM [21], Dark Matter [28], Social GAN and SR-LSTM [24] methods.

A. DATASETS AND METRICS

Our proposed model is evaluated on three public pedestrian trajectory datasets: UCLA campus [28], hyang [52] and coupa [53]. These three datasets have different scenes: courtyard, crossroad, and park. Besides, we also test our proposed model on our own collected campus dataset. Among the datasets, UCLA campus dataset contains the most amount of complex scenarios where the moving directions of people cross each other. Thus, the prediction tasks on UCLA campus dataset was the most challenging.

In our experiments, we use the Mean Average Displacement error (MADE) [32] in meters as the metric to evaluate the performance of trajectory prediction. The MADE is the Mean Euclidean distance between predicted points and ground truth at all predicted time steps. In our method, the MADE is calculated by the ground truth trajectory and its closest predicted possible trajectory.

FIGURE 7. The pedestrian trajectory prediction performance of our proposed model in the datasets: UCLA, hyang, and coupa. The green line is the ground truth trajectory and the blue line the prediction trajectory.

B. IMPLEMENTATION DETAILS

The network structure of CNN is shown in Table 1. The final output size of CNN is 512. The number of LSTM units is 500. The parameter β is set to 0.6, which is used to balance the proportion of the space reward and time reward. The parameter K is set to 4, which is the number of the expanded nodes at each level during the tree search process. The number of possible end points N is set to 5. The threshold τ of walkable region correction is set to 6. The range R of the start point is set to 5. During the training process, the mini-batch size is set to 32. The learning rate is set to 0.0001 and the training optimizer is Adam. These hyperparameters are all obtained by the trading off between the performance and computation complexity through trial in their possible values after grid searching in the given range.

C. EXPERIMENTAL RESULTS UNDER VARYING PARAMETER VALUES

The MADEs of our proposed method under varying parameter values in the dataset: UCLA are shown in Fig. 6. The parameter K is the number of the expanded nodes at each level during the tree search process. The parameter β is employed to control the proportion of the space reward and time reward.

The blue line is under parameter K and as the K increases, the MADE of our proposed method will get smaller

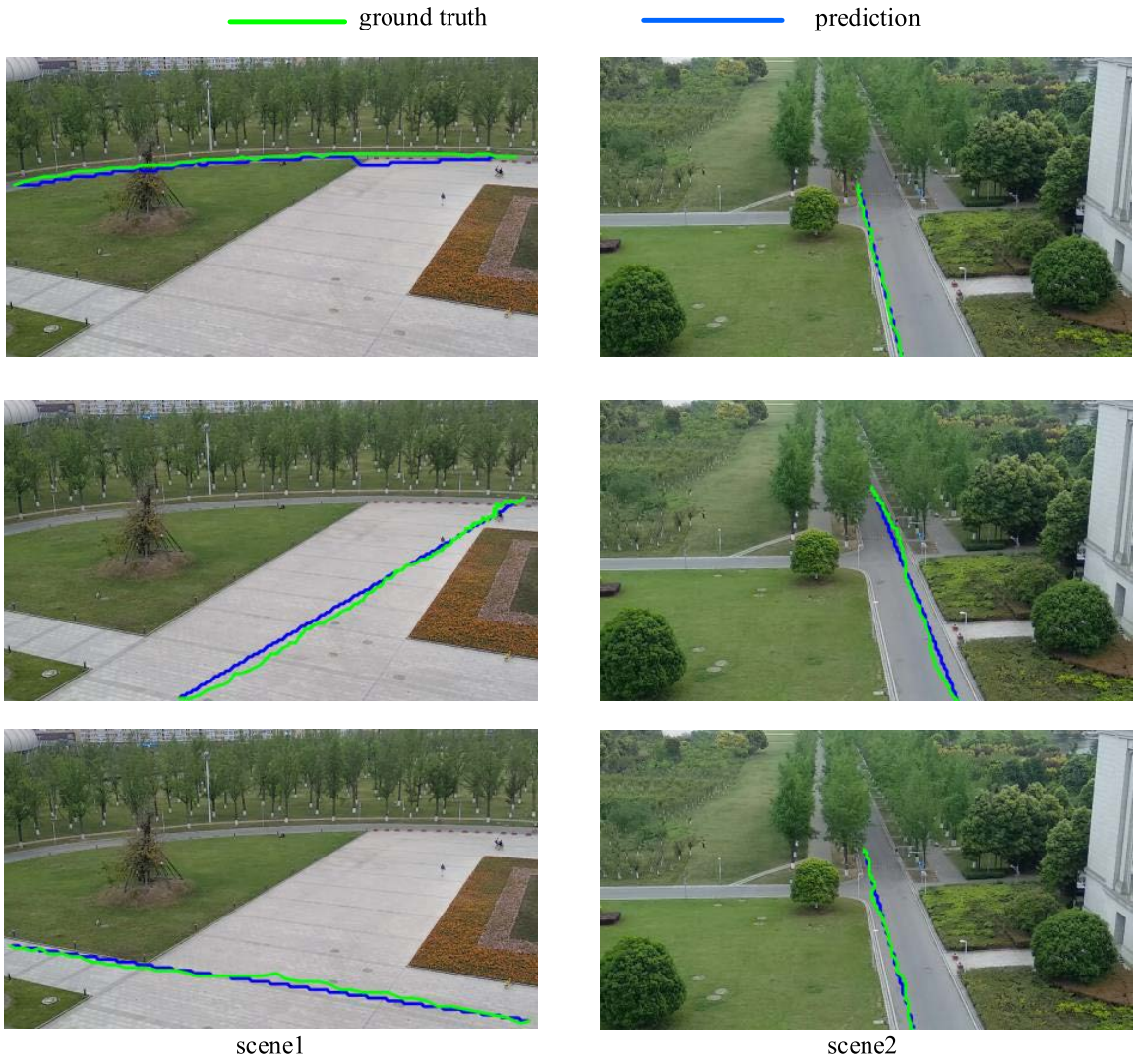

FIGURE 8. The pedestrian trajectory prediction performance of our proposed model in the datasets: our campus, which contains two different scenes. The green line is the ground truth trajectory, and the blue line is the prediction trajectory. The predicted trajectories are very close to the ground truth trajectories.

at $\beta = 0.6$. It means that in the tree search process, more expanded nodes at each level during the tree search process will improve the performance of the proposed method. However, the computation complexity will significantly increase. After balancing the performance and computation complexity, we select $K = 4$.

The red line is under parameter β and as the β increases, the MADE of our proposed method will first get smaller and then get bigger. Hence, we select $\beta = 0.6$, which gets the smallest MADE. It means space and time rewards are equally important for the long-term trajectory prediction. What's more, only considering the space reward or time reward will perform much worse.

D. COMPARISON WITH EXISTING METHODS

In this section, our model is compared with several recent existing works: (1) Social LSTM [21]: An LSTM model is proposed to learn general human movement and predict their future trajectories. (2) Dark Matter [28]: A model is proposed to infer "Dark Matter" and predict pedestrian intents and trajectories. (3) Social GAN [25]: A pedestrian trajectory prediction model is proposed by combining tools from sequence prediction and generative adversarial networks. (4) SR-LSTM [24]: A data-driven state refinement module for the LSTM network is proposed to activate the utilization of the current intention of neighbors.

The experimental results are shown in Table 2. The datasets and evaluation metrics are set to be consistent for all of the methods. The MADEs of our proposed model are the smallest in all the three datasets, which means that the predicted trajectory of our proposed model is closest to the real trajectory. We can also know that the method "Dark Matter" performs the second best. The reason is that our proposed model and the "Dark Matter" method both utilize the surrounding latent

environment information, which can correct the accumulated error during the long-term trajectory prediction. The ''dark matter'' method uses the ''dark matter'' to represent the surrounding latent environment information, which plays a key role in the trajectory prediction. In our model, we use the neural network to extract the surrounding latent environment information, which has an excellent instructive effect during the tree search for the trajectory prediction. Besides, our proposed method considers both the space and time dimensions, which make our proposed method perform better than the ''dark matter'' method. The trajectory prediction performance of our proposed model on UCLA campus dataset is shown in Fig. 7. The green line is the ground truth trajectory, and the blue line is the prediction trajectory. From the results, we can observe that even people move in various directions, our method can precisely predict the trajectory. In the three datasets, the predicted trajectories of our model are all close to the real trajectory. After calculating, in the datasets: UCLA, hyang and coupa, the performance of our proposed method improves 21.22%, 25.21%, and 18.39%, respectively, compared with the ''dark matter'' method [28].

Finally, We also tested our proposed model on our campus, which contains two different scenarios. Our proposed method can perform well in both scenarios shown in Fig. 8 where the green line is the ground truth trajectory, and the blue line is the prediction trajectory. It proves that our proposed method can be applied to the real environment and is adapted to a variety of scenarios.

V. CONCLUSION

In this paper, we develop a space-time tree search method for long-term pedestrian trajectory prediction. Different from existing methods only considering the problem from the space dimension, we transform the trajectory prediction problem into a joint space-time tree search process by mapping the environment to a grid map. Actually, the human trajectory contains the space and time dimensions. Hence, we can improve the trajectory prediction accuracy from the two dimensions. Then, we employ a space-time reward-trained neural network to boost the tree search to obtain the optimal predicted trajectory, where the neural network can output the prior search probabilities. After testing, in the datasets: UCLA, hyang, and coupa, the performance of our proposed method improves 21.22%, 25.21%, and 18.39%, respectively, compared with the ''dark matter'' method [28].

In future work, we will consider migrating our proposed model to the crowded scenario. In the crowded scenario, the behaviors of pedestrians can influence each other. Hence, the neural network should be able to extract the surrounding pedestrians' information. It may be a scheme to realize it using objection detection and tracking.

REFERENCES

[1] F. Altché and A. de La Fortelle, ''An LSTM network for highway trajectory prediction,'' in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2017, pp. 353–359.

[2] S. Li, F. Wu, S. Luo, Z. Fan, J. Chen, and S. Fu, ''Dynamic online trajectory planning for a UAV-enabled data collection system,'' *IEEE Trans. Veh. Technol.*, early access, Aug. 22, 2022, doi: 10.1109/TVT.2022.3200458.

[3] B. Kim, C. M. Kang, J. Kim, S. H. Lee, C. C. Chung, and J. W. Choi, ''Probabilistic vehicle trajectory prediction over occupancy grid map via recurrent neural network,'' in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2017, pp. 399–404.

[4] I. Kaminer, A. Pascoal, E. Hallberg, and C. Silvestre, ''Trajectory tracking for autonomous vehicles: An integrated approach to guidance and control,'' *J. Guid., Control, Dyn.*, vol. 21, no. 1, pp. 29–38, Jan. 1998.

[5] X. Li, S. Chen, Y. Zhou, J. Chen, and G. Feng, ''Intelligent service migration based on hidden state inference for mobile edge computing,'' *IEEE Trans. Cognit. Commun. Netw.*, vol. 8, no. 1, pp. 380–393, Mar. 2022.

[6] J. Chen, S. Chen, Q. Wang, B. Cao, G. Feng, and J. Hu, ''IRAF: A deep reinforcement learning approach for collaborative mobile edge computing IoT networks,'' *IEEE Internet Things J.*, vol. 6, no. 4, pp. 7011–7024, Aug. 2019.

[7] R. W. Liu, M. Liang, J. Nie, W. Y. B. Lim, Y. Zhang, and M. Guizani, ''Deep learning-powered vessel trajectory prediction for improving smart traffic services in maritime Internet of Things,'' *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 5, pp. 3080–3094, Sep. 2022.

[8] R. W. Liu, M. Liang, J. Nie, Y. Yuan, Z. Xiong, H. Yu, and N. Guizani, ''STMGCN: Mobile edge computing-empowered vessel trajectory prediction using spatio-temporal multigraph convolutional network,'' *IEEE Trans. Ind. Informat.*, vol. 18, no. 11, pp. 7977–7987, Nov. 2022.

[9] M. Luber, J. A. Stork, G. D. Tipaldi, and O. A. Kai, ''People tracking with human motion predictions from social forces,'' in *Proc. IEEE Int. Conf. Robot. Autom.*, Sep. 2010, pp. 2–4.

[10] D. Makris and T. Ellis, ''Learning semantic scene models from observing activity in visual surveillance,'' *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 35, no. 3, pp. 397–408, Jun. 2005.

[11] R. Mehran, A. Oyama, and M. Shah, ''Abnormal crowd behavior detection using social force model,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 20–25.

[12] T. Fernando, S. Denman, S. Sridharan, and C. Fookes, ''Soft + hardwired attention: An LSTM framework for human trajectory prediction and abnormal event detection,'' *Neural Netw.*, vol. 108, pp. 466–478, Dec. 2018.

[13] D. Helbing and P. Molnár, ''Social force model for pedestrian dynamics,'' *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 51, no. 5, p. 4282, May 1995.

[14] X. Yang, H. Dong, Q. Wang, Y. Chen, and X. Hu, ''Guided crowd dynamics via modified social force model,'' *Phys. A, Stat. Mech. Appl.*, vol. 411, pp. 63–73, Oct. 2014.

[15] R. Mehran, A. Oyama, and M. Shah, ''Abnormal crowd behavior detection using social force model,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 935–942.

[16] W. Zeng, P. Chen, H. Nakamura, and M. Iryo-Asano, ''Application of social force model to pedestrian behavior analysis at signalized crosswalk,'' *Transp. Res. C, Emerg. Technol.*, vol. 40, pp. 143–159, Mar. 2014.

[17] J. Wiest, M. Hoffken, U. Kresel, and K. Dietmayer, ''Probabilistic trajectory prediction with Gaussian mixture models,'' in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2012, pp. 141–146.

[18] G. Aoude, J. Joseph, N. Roy, and J. How, ''Mobile agent trajectory prediction using Bayesian nonparametric reachability trees,'' in *INFOTECH Aerospace 2011*. American Institute of Aeronautics and Astronautics, 2011, p. 1512.

[19] X. Wang, X. Ma, and W. E. L. Grimson, ''Unsupervised activity perception in crowded and complicated scenes using hierarchical Bayesian models,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 3, pp. 539–555, Mar. 2009.

[20] X. Wang, K. T. Ma, G.-W. Ng, and W. E. L. Grimson, ''Trajectory analysis and semantic region modeling using nonparametric hierarchical Bayesian models,'' *Int. J. Comput. Vis.*, vol. 95, no. 3, pp. 287–312, 2011.

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

[22] H. Su, J. Zhu, Y. Dong, and B. Zhang, ''Forecast the plausible paths in crowd scenes,'' in *Proc. 26th Int. Joint Conf. Artif. Intell.*, vol. 1, Aug. 2017, p. 2.

[23] H. Xue, D. Q. Huynh, and M. Reynolds, ''SS-LSTM: A hierarchical LSTM model for pedestrian trajectory prediction,'' in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2018, pp. 1186–1194.

[24] P. Zhang, W. Ouyang, P. Zhang, J. Xue, and N. Zheng, "SR-LSTM: State refinement for LSTM towards pedestrian trajectory prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12085–12094.
[25] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social GAN: Socially acceptable trajectories with generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2255–2264.
[26] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, Nov. 2011.
[27] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2881–2890.
[28] D. Xie, T. Shu, S. Todorovic, and S.-C. Zhu, "Learning and inferring 'dark matter' and predicting human intents and trajectories in videos," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 7, pp. 1639–1652, Jul. 2017.
[29] G. Antonini, M. Bierlaire, and M. Weber, "Discrete choice models of pedestrian walking behavior," *Transp. Res. B, Methodol.*, vol. 40, no. 8, pp. 667–687, 2006.
[30] E. Bonabeau, "Agent-based modeling: Methods and techniques for simulating human systems," *Proc. Nat. Acad. Sci. USA*, vol. 99, no. 3, pp. 7280–7287, May 2002.
[31] S. Kim, S. J. Guy, W. Liu, R. W. Lau, M. C. Lin, and D. Manocha, "Predicting pedestrian trajectories using velocity-space reasoning," in *Algorithmic Foundations of Robotics X*. Cham, Switzerland: Springer, 2013, pp. 609–623.
[32] S. Pellegrini, A. Ess, K. Schindler, and L. van Gool, "You'll never walk alone: Modeling social behavior for multi-target tracking," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep. 2009, pp. 261–268.
[33] S. Yi, H. Li, and X. Wang, "Understanding pedestrian behaviors from stationary crowd groups," in *Proc. IEEE Conf. CVPR*, Jun. 2015, pp. 3488–3496.
[34] A. Vemula, K. Muelling, and J. Oh, "Modeling cooperative navigation in dense human crowds," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 1685–1692.
[35] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.
[36] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
[37] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[38] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4700–4708.
[39] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.
[40] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, and Y. Chen, "Mastering the game of go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354–359, 2017.
[41] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[42] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 6645–6649.
[43] A. Graves, N. Jaitly, and A.-R. Mohamed, "Hybrid speech recognition with deep bidirectional LSTM," in *Proc. IEEE Workshop Autom. Speech Recognit. Understand.*, Dec. 2013, pp. 273–278.
[44] X. Jia, E. Gavves, B. Fernando, and T. Tuytelaars, "Guiding the long-short term memory model for image caption generation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2407–2415.
[45] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo, "Image captioning with semantic attention," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4651–4659.
[46] L. Gao, Z. Guo, H. Zhang, X. Xu, and H. T. Shen, "Video captioning with attention-based LSTM and semantic consistency," *IEEE Trans. Multimedia*, vol. 19, no. 9, pp. 2045–2055, Sep. 2017.
[47] D. Guo, W. Zhou, H. Li, and M. Wang, "Hierarchical LSTM for sign language translation," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 3–5.
[48] N. Lee, W. Choi, P. Vernaza, C. B. Choy, P. H. S. Torr, and M. Chandraker, "DESIRE: Distant future prediction in dynamic scenes with interacting agents," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2165–2174.
[49] S. D. Khan, L. Alarabi, and S. Basalamah, "Toward smart lockdown: A novel approach for COVID-19 hotspots prediction using a deep hybrid neural network," *Computers*, vol. 9, no. 4, p. 99, Dec. 2020.
[50] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[51] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2000, pp. 1057–1063.
[52] A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese, "Learning social etiquette: Human trajectory understanding in crowded scenes," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 549–565.
[53] A. Robicquet, A. Alahi, A. Sadeghian, B. Anenberg, J. Doherty, E. Wu, and S. Savarese, "Forecasting social navigation in crowded complex scenes," 2016, *arXiv:1601.00998*.

TINGYONG WU (Member, IEEE) received the B.E., M.S., and Ph.D. degrees in communication systems from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 1998, 2001, and 2007, respectively. He is currently an Associate Professor with the National Key Laboratory of Science and Technology on Communications, UESTC. His current research interests include signal processing in wireless communication, circuit-system design, and artificial intelligence for communication.

PEIZHI LEI (Student Member, IEEE) received the B.E. degree from the School of Computer Science and Engineering, University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2022, where he is currently pursuing the M.Sc. degree in communications with the National Key Laboratory of Science and Technology. His current research interests include computer vision, data augmentation, and active learning.

FUQIANG LI received the B.E. and M.Sc. degrees in communication systems from the University of Electronic Science and Technology of China (UESTC), Chengdu, China. He is currently working as a Senior Engineer with China Electronic Technology Group Corporation (CETC), 20th Institute, Xi'an, China. His current research interest includes communication networks.

JIENAN CHEN (Senior Member, IEEE) received the B.S. and Ph.D. degrees in communication systems from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2007 and 2014, respectively. He was worked with the School of Electrical and Computer Engineering, University of Minnesota, Minneapolis, as a Visiting Scholar, in 2012 and 2014, and then as a Postdoctoral Scholar with the University of North Texas. He is currently a Professor with the National Key Laboratory of Science and Technology on Communications, UESTC. His current research interests include VLSI circuit designs, low-power circuit designs, stochastic computation-based system designs, machine learning-based signal processing, artificial intelligence for networking, and circuit-system design. He served as a TPC Member for GLOBECOM and ICC and the Symposium Chair for GlobalSIP.