2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) | 979-8-3503-7770-5/24/$31.00 ©2024 IEEE | DOI: 10.1109/IROS58592.2024.10802269

SocialNav-FTI: Field-Theory-Inspired Social-aware Navigation Framework based on Human Behavior and Social Norms

Siyi Lu[1], Ping Zhong*[1], Shuqi Ye[1], Bolei Chen[1], Yu Sheng[1], Run Liu[2]

***Abstract*— Social navigation is a key consideration for integrating robots into human environments. Concurrently, it imposes heightened requisites: tasks must not only be executed succesfully without collisions, but also adhere to principles encompassing comprehensibility, courtesy, social compliance, comprehension, foresight, and scenario compliance. In this paper, we present the incorporation of social norms as a guiding framework for robot navigation within social contexts. We adopt field theory to provide a formal elucidation of the social norms, using Physical-Informed Neural Network (PINN) to predict pedestrian movement under the influence of social norms, respectively, and using Reinforcement Learning (RL) for navigation. We use supervised learning to train the pedestrian velocity field prediction model and reinforcement learning to train the navigation policy. We conduct three parts of experiments: (1) analyzing the spatiotemporal characteristics of the velocity field in the walking pedestrians dataset; (2) evaluating the accuracy of the vector field prediction in the pedestrian dataset; (3) using Gazebo simulation and the PEDSIM library to evaluate the improvement of navigation performance under constraints of social norms. Experiments have confirmed that the pedestrian motion data set indeed satisfies the Gaussian divergence theorem and can be described by the concept of field. The performance of navigation strategies incorporating social rules has been improved to a certain extent.**

I. Introduction

Social navigation is a multifaceted challenge that requires input from various disciplines, including human-computer interaction, psychology, and sociology [1]. The primary purpose of social navigation is to allow robots to move through crowds and reach their destination safely and efficiently, while avoiding collisions. According to the principles of social robot navigation [2], socially navigating robot should be defined as one that respects the principles of safety, comfort, comprehensibility, courtesy, social compliance, comprehension, foresight, and scenario compliance. This is the key to the socialization of robots and the fact that people do not reject robots appearing in various life scenes. In brief, we expect social robots to be able to recognize social cues, norms, and expectations, to have the understanding to interpret them correctly, and to have the capabilities to respond appropriately.

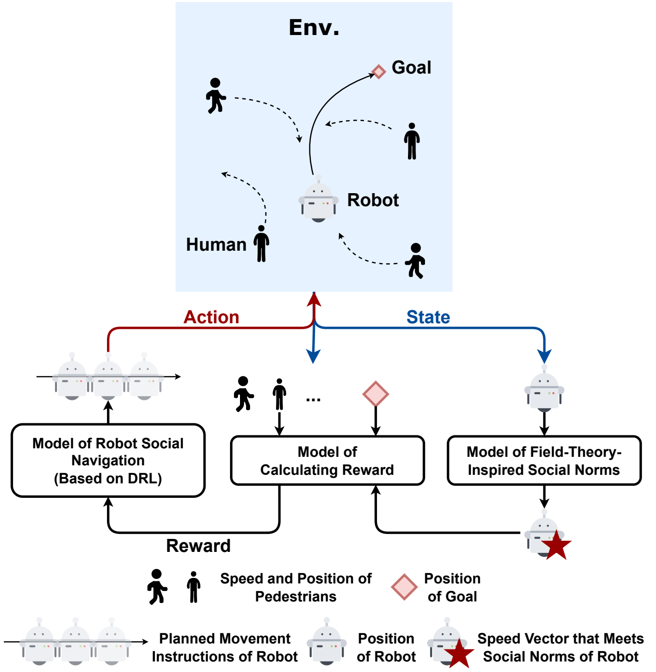
Env.
Goal
Robot
Human
Action
State
Model of Robot Social Navigation (Based on DRL)
Model of Calculating Reward
Model of Field-Theory-Inspired Social Norms
Reward
Speed and Position of Pedestrians
Position of Goal
Planned Movement Instructions of Robot
Position of Robot
Speed Vector that Meets Social Norms of Robot

Fig. 1. Illustration of proposed human-robot-environment relational model.

Existing work on modeling and extracting social signals is often based on the construction of a relational graph between robots and pedestrians surrounding it [3]. In a relational graph, nodes represent robots or pedestrians, and edges represent the relationships between them. By analyzing this relational graph, a robot can infer the intentions of surrounding pedestrians, and make corresponding decisions, such as avoiding pedestrians, cooperating with pedestrians, or competing with pedestrians. However, this modeling and extraction method has its own limitations. First, social navigation based only on relational graph may result in the robot having a limited understanding of human behavior, and thus making incorrect judgments about the social relationships between pedestrians. In addition, robots may have limited understanding of social norms and may make decisions that are inconsistent with social norms. This is because the relational graph is a human-robot relational model that only considers human behavior, but social navigation need a human-robot-environment relational model that takes into account human behavior and social norms.

Rules and conventions may allow the robot to better interact with pedestrians [2]. Many of the social abilities of human beings often come from following social conventions.

[1] Siyi Lu, Ping Zhong, Shuqi Ye, Bolei Chen, Yu Sheng are with School of Computer Science and Engineering, Central South University, Changsha, 410083, China. `llllsy@csu.edu.cn`, `ping.zhong@csu.edu.cn`, `suky-ye@csu.edn.cn`, `boleichen@csu.edu.cn`, `shengyu@csu.edu.cn`

[2] Run Liu is with Research Center of Ubiquitous Sensor Networks, University of Chinese Academy of Sciences, Beijing 100049, China. `liurun22@mails.ucas.ac.cn`

*Ping Zhong is the corresponding author

[4], [5], [6]. For example, the optimal preference for driving on the left or right is a typical social rule. Identifying and following local rules for driving on the left or right can greatly reduce the occurrence of traffic conflicts [7]. Therefore, the lack of semantic information about the environment will inevitably reduce the robot's understanding of human behavior and social norms, thus falling into a local optimal solution.

Different countries and regions have different cultures, and it is difficult to use a set of social norms to constrain socially navigating robot in all scenarios. We must use a more general framework to represent social norms. Intuitively, human behavior follows social norms, and social norms also affect human behavior. We want to indirectly obtain the guidance of social norms on the movement of socially navigating robot from the group behavior constrained by social norms. Inspired by [8], it was discovered that human behavior can be represented by fields and satisfies the Gaussian divergence theorem. This shows that human behavior under the influence of social norms does have rules, and the rules can be described by field theory.

In this paper, we study robot social navigation based on field theory, using Physical-Informed Neural Network (PINN) to distill the salient features of human behavior and social norms. At the same time, a velocity vector representing social norms is predicted at the robot's location, which is used to constrain the movement of the socially navigating robot. Finally, we leverage Reinforcement Learning (RL) to train the navigation policy. We respectively proved the feasibility in theory and practice of using pedestrian velocity vector field to indirectly represent social norms and use it as a reward function for reinforcement learning to constrain robot navigation strategies. At the same time, it also quantitatively proves the improvement effect of our method on navigation. The illustration of proposed human-robot-environment relational model is shown in Fig. 1. The code of our approach is publicly accessible at https://github.com/SiyiLoo/SocialNav-FTI

II. Related Work

Social navigation essentially allows robots to navigate using social signals. Properly extracting and utilizing social signals can improve robot navigation efficiency and socialization. Hence, current research on social navigation can be categorized based on the difference in the extraction and utilization of social signals.

Social navigation with no social signals. Early work designed rule-based navigation strategies based on ideal human cooperative behavior. RVO [9] and ORCA [10] solve for collision avoidance for multi-robot scenario under reciprocal assumptions. This approach is effective only when pedestrians are able to fully cooperate, which is often not possible. The ROS [11] navigation framework is another classic model-based approach. [12] takes human safety into the path planning algorithm and incorporates costmaps based on the ROS navigation framework to address human-aware navigation problems. All of these model-based methods all need to establish a pedestrian kinematics model, so as to use optimization methods to constrain the robot's motion to achieve obstacle avoidance. The existing pedestrian kinematics model cannot accurately depict the real pedestrian environment, so the navigation effect is poor.

Social navigation with coarse-grained social signals. Some studies apply supervised learning to social navigation. Trained with expert demonstrations, the supervised model is able to learn a complex mapping of target locations to the robot's required steering commands [13], [14], [15], [16]. The end-to-end navigation method based on supervised learning learns the connection between sensor input and navigation target, further improving the navigation success rate. However, pedestrian behavior is influenced by various intricate social cues and interactions, which pose a significant challenge for end-to-end learning methods to accurately capture and incorporate in order to improve robot navigation performance. It can also be seen from experiments that navigation methods based on supervised learning are difficult to generalize to new scenes and large-scale crowds.

Social navigation with social intent signals. With the development of trajectory prediction research [17], robots have begun to actively seek cooperation with pedestrians in social navigation, planning their own movements by inferring pedestrian intentions and predicting pedestrians' future trajectories. **'Understanding Pedestrians - Anticipate and match pedestrian behavior"** is an important guiding principle in social navigation [2]. In Social-LSTM [18], a recurrent network was used to simulate the movement of each pedestrian, and then they used a pooling mechanism to aggregate the recurrent output and predict subsequent trajectories. However, some studies show that relying only on the pedestrian's future movement trajectory to plan its own movement will cause the robot to freeze [19], [20].

Social navigation with social relation signals. In order to further improve the social awareness and navigation performance of robots, [6], [21] use deep reinforcement learning to train social navigation strategies. With the development of graph neural networks and attention mechanisms [22], [23], [24] began to mine social relationship signals in social navigation scenarios, using self-attention mechanisms to capture interactive information that will indirectly affect the robot. [25] proposed approach leverages graph convolutional networks for navigation. In the method proposed by [3], the relationship model is further expanded, and GCN is used not only to capture the robot's attention, but also to capture the attention between people. At the same time, the relationship model is combined with the pedestrian motion prediction model.

Social navigation with social norm signals. The semantic information of navigation scenes is also an important social signal. The semantic information of navigation scenes usually acts on pedestrian groups in the form of social norms. Recent research [8] shows that human migration from a macro perspective satisfies the Gaussian divergence theory and can be well described by a vector field model. This article will extract social norm signals based on the vector field model and improve the reward function in reinforcement learning based on social norms to improve navigation performance.

This article aims to make full use of social norm signals based on existing social relationship signals to guide robots to conduct social navigation in a more ideal way.

III. Approach

Building on prior work in robotics [26], [27], [28], [29], [30], we employ a hierarchical approach for social robot navigation, consisting of Global Planning and Local Planning components. Following [31], we use pure pursuit algorithm [32] to select points on the complete path as sub-goal to feed to our DRL-based Local Planner. In this section, we detail our DRL-based Local Planner and a novel reward function that takes into account social norms.

A. Problem Formulation

The focus of our work is the design of the local planner, which generates the motion control instructions for the robot. The objective is to enable the robot to quickly reach its target while avoiding collisions with surrounding pedestrians and considering social norms to ensure a comfortable walking experience for the pedestrians.

This local planning problem can be modeled as a Partially Observed Markov Decision Process (POMDP), represented by the tuple (S, A, P, R, Ω, O), which captures the robot's restricted field of view. Here, observation space $(\Omega = [o, h, g])$ consists of three component: Obstacles distance (o), human kinematics (h) and sub-goal position (g), action space $(A = [v_x, \omega_z])$ consists of linear velocity (v_x) and rotational velocity (ω_z). After obtaining partial observations (Ω^t) from the sensor, the robot uses the partial observations (Ω^t) as state space (s^t) to the control policy (π_θ). The control policy mapping state space (s^t) to action space (a^t) that the agent has to respond. Reward function R will be described in detail in Sec. III-B.

To solve the POMDP, we employ deep reinforcement learning (DRL) techniques. The general goal of DRL is to find an optimal neural network-based policy π^* that maximizes the value function $v^{\pi}(s)$ for each state:

$$v^{\pi^*}(s^t) = \sum_{i=0}^{\infty} \gamma^{i \cdot v_{pref}} R^{t+i}(s^{t+i}, \pi^*(s^{t+i})) \tag{1}$$

where $\gamma \in (0,1)$ is the discount factor, $R^{t+i}(s_{t+i}, \pi^*(s_{t+i}))$ is the reward at time $t+i$, preferred velocity v_{pref} is the normalized term in the discount factor.

B. Reward Function

The reward function is an important component of reinforcement learning. According to the principles of social robot navigation [2], in addition to ensuring pedestrian safety and quickly reaching the goal, robots should also adhere to social norms in public places. Social norms not only apply to humans, but also enable robots to better interact with humans. Many social skills are a matter of following routines rather than optimizing performance [4], [5], [6].

In this paper, we use the speed vector to represent the tendency of social norms, and use the similarity between the speed vector and the robot's actual speed to represent the degree to which the robot follows social norms, and reflect it in the reward function. Therefore, We define the reward function as:

$$r^t = r_g^t + r_c^t + r_\omega^t + r_n^t. \tag{2}$$

Same as [31], r_g^t incentivizes the robot to navigate towards the desired goal, r_c^t penalizes unsafe behavior that may lead to collisions, r_ω^t punishes large turns, making the robot's trajectory smooth. r_n^t punishes robot for violating social norms. The formal description of r_g^t, r_c^t, and r_ω^t is as follows:

$$r_g^t = \begin{cases} r_{\text{goal}} & \text{if } \left\|p_g^t\right\| < g_m \\ -r_{\text{goal}} & \text{else if } t \geq t_{\max} \\ r_{\text{path}}\left(\left\|p_g^{t-1}\right\| - \left\|p_g^t\right\|\right) & \text{otherwise} \end{cases} \tag{3}$$

$$r_c^t = \begin{cases} r_{\text{collision}} & \text{if } \left\|p_o^t\right\| \leq d_r \\ r_{\text{obstacle}}\left(d_m - \left\|p_o^t\right\|\right) & \text{else if } \left\|p_o^t\right\| \leq d_m \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

$$r_w^t = \begin{cases} r_{\text{rotation}}\left|\omega_z^t\right| & \text{if } \left|\omega_z^t\right| > \omega_m \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

The influence of social norms on movement can be reflected in the motion patterns at a macro level [33]. We introduce a spatial representation, **field**, which encapsulates the statistical dynamic behavior in the environment. The speed vector in the speed vector field represents the speed with higher probability among the speeds of pedestrians passing this location. Social norms are also a high-probability common manifestation in pedestrian movements. Then the velocity vector reflects social rules to a certain extent. The formal description of r_n^t is as follows.

$$r_n^t = r_{\text{field}}\left(\left\|v_f^t - v_z^t\right\|\right). \tag{6}$$

We use v_f^t to represent the velocity vector of the robot's position in the velocity vector field. And use the similarity between the robot's speed v_z^t and v_f^t as a penalty term of the reward function,it constrains the robot to behave in the opposite direction of most pedestrians. We will go into more detail about how v_f^t is calculated in Sec. III-C.

C. PINN-based Crowd Velocity Field Prediction

We aim to predict the velocity field components by leveraging the knowledge of the velocity field in some number of locations in the crowded environment. Through the pedestrian velocity vector field, we can obtain the velocity vector at any location, regardless of whether there are pedestrians at this location. According to the characteristics of pedestrian scenes commonly used in social navigation, we can know that pedestrians in the environment do not suddenly disappear or appear, so the velocity vector field divergence $J = 0$, and the partial differential equation of the velocity vector field is:

$$\nabla \cdot B = 0. \tag{7}$$

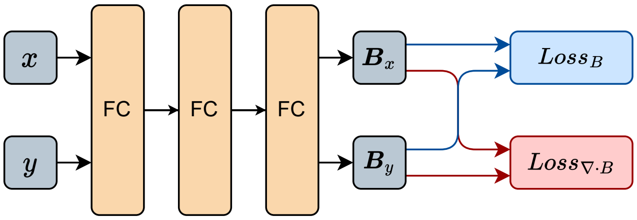

Fig. 2. PINN-based crowd velocity field prediction network. The input is the location information of pedestrians, and the output is the x-axis component and the y-axis component of the crowd velocity field. In addition to the prediction loss, the loss function also has a loss function about physical information.

According to the characteristic of crowd velocity field and sufficient number of crowd velocity data in the environment, we utilize PINNs to predict crowd velocity field [34]. With PINNs, prior physical knowledge about the partial differential equations of the pedestrian velocity vector field comes into play with the deep neural network, and the pedestrian velocity dataset and mathematical model are smoothly combined.

The partial derivatives (Eq. (7)) of the equation can be calculated through the automatic differentiation tools provided in TensorFlow [35] . The neural network we train for pedestrian velocity field prediction within the approximate area will have the structure shown in Fig. 2. The activation function of each hidden layer is hyperbolic tangent. Then, the network is trained with a loss function that combines the ground truth and the divergence constraint.

$$\text{Loss} \ = \text{Loss}_B + \lambda \, \text{Loss}_{\nabla \cdot B}, \tag{8}$$

where

$$\text{Loss}_B := \frac{1}{N_B} \sum_{i=1}^{N_B} \left\| B\left(r_B^i\right) - B_s\left(r_B^i\right) \right\|^2, \tag{9}$$

$$\text{Loss} \nabla \cdot B := \frac{1}{N_f} \sum_{i=1}^{N_f} \left| \nabla \cdot B\left(r_d^i\right) \right|^2, \tag{10}$$

where the parameter λ is a hyperparameter that affects network performance. N_B is the number of people. The variables r_d^i and r_B^i represent the position of the crowd and the collocation points within the environment, respectively. N_f denotes the number of collocation points in the domain and B_s is the vector of pedestrian velocity field measured at r_d^i. The collocation points r_i^d , are sampled from the environment and fixed throughout the training process, which are used to verify whether the constraints of the mathematical model of the pedestrian velocity vector field are met.

D. Deep Reinforcement Learning for Crowd Navigation

Our deep reinforcement learning framework follows [31] and adopts the Actor-Critic architecture, specific details are shown in Fig. 3. The input of deep reinforcement learning comes from the sensor equipment mounted on the robot and the target task, including camera, lidar, and goal point. We process raw sensor data and target tasks separately to obtain partial observation (O^t). First, input the raw data obtained by the camera into the YOLO framework to obtain the kinematic information of pedestrians (h^t). Secondly, lidar information is used to construct a lidar map that stores obstacle distance information (o^t). Finally, the target position is split to obtain the intermediate target point (g^t). The feature extractor module takes the partial observation O^t as input and extracts a set of high-level features that capture salient characteristics of the observed state. These high-level features are then fed into the critic module, which generates an estimate of the state value function. Concurrently, the actor module utilizes the same high-level features to produce the steering action a^t, which represents the robot's decision for the current time step.

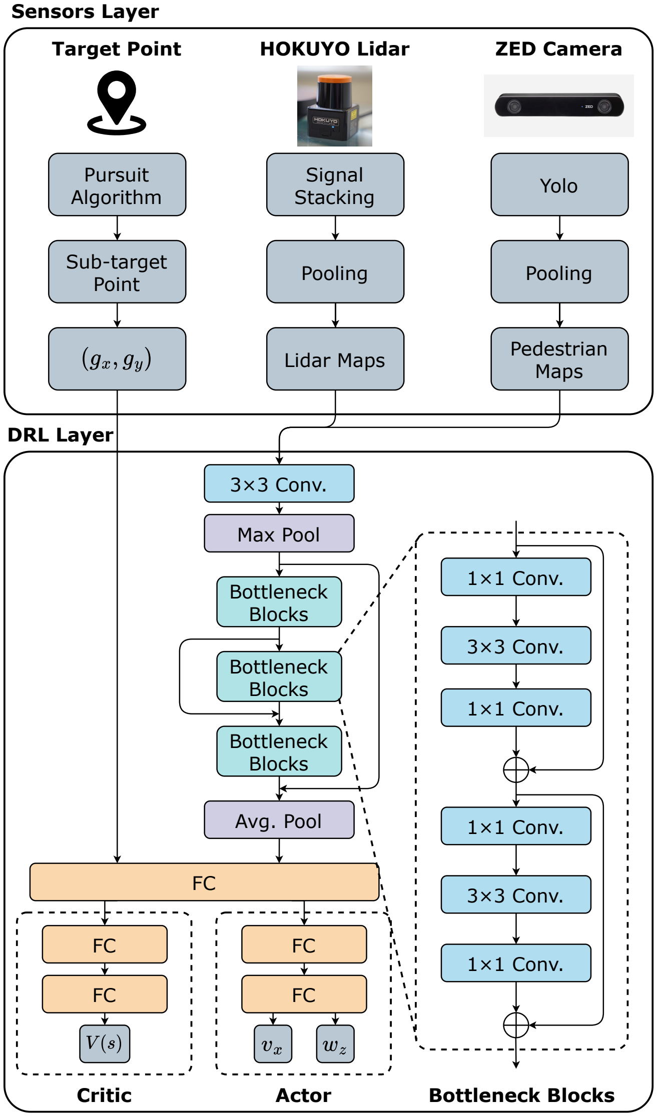

Fig. 3. Overall system architecture of SocialNav-FTI. The inputs to the navigation framework are camera data, radar data, and target locations, and pedestrian, obstacle, and target features are generated based on these raw inputs. And use the neural network to extract high-level features representing the environment and tasks, and combine it with the Actor-Critic framework to implement reinforcement learning.

Our feature extractor network is based on the CNN backbone architecture proposed in [36]. It takes the input observations h^t and o^t, which contain relevant environmental information, and extracts salient spatial and temporal features that capture the characteristics of the environment. These high-level environmental features are then fused with the goal information using a single fully connected layer. Each convolutional layer in the deep reinforcement learning framework is followed by a batch normalization layer to accelerate the convergence of the training process and a rectified linear unit (ReLU) activation layer to enhance the nonlinear modeling capabilities of the neural networks. Finally, both the critic and the actor modules are composed of two fully connected layers.

E. Joint Value Estimation and Crowd Velocity Field Prediction

The pedestrian speed vector field is closely related to the time area. Pedestrians in the same area may follow different social norms at different times of the day. Therefore, the pedestrian speed vector field will change over time. We need to update the pedestrian speed vector field in real time during the navigation process and train the pedestrian speed vector field prediction network in real time. Fortunately, it has been verified through experiments that the velocity vector field changes slowly with time and does not require high real-time performance. We can appropriately reduce the training frequency of the pedestrian velocity vector field prediction network, which will greatly reduce the complexity of the entire algorithm.

The pseudocode for the joint DRL value estimation learning and crowd velocity field prediction learning scheme is in Algorithm 1. We use supervised learning to train the pedestrian velocity field prediction model and reinforcement learning to train the navigation policy. According to the pedestrian velocity field characteristics observed on the walking pedestrian data set: at the same location, the pedestrian velocity field changes slowly over time, and there is a certain similarity between these pedestrian velocity fields. Therefore, we can first pre-train the PINN model using the walking pedestrian data set, and then fine-tune the pre-trained model in the simulation environment to update the pedestrian velocity field prediction model in real time. We utilize the pedestrian velocity field prediction model from the previous time period as a pre-trained model for the pedestrian velocity field prediction model in the subsequent period, and fine-tune it on this basis. This alleviates the need for computing resources and minimizes the amount of calculations at each step in navigation.

Algorithm 1: Joint training for f_{pre} and f_{val}

```
Initialize $f_{pre}$ with walking pedestrian dataset;
Initialize target value network $\hat{f}_{val} \leftarrow f_{val}$;
Initialize the replay memory $E \leftarrow D$;
for episode = 1, N do
    Initialize a sequence $S^0$ randomly; if not reach goal then
        Update $f_{pre}$ by minimizing $L_1 = ||f_{pre}(p_i) - vi||$;
        $a_t \leftarrow$ argmax$_{a_t \in A} \hat{R}\left(S^t, a^t, \hat{S}^{t+1}\right) + \gamma^{\Delta t \cdot v_{\text{pref}}} V^d\left(\hat{S}^{t+1}\right)$ where $\hat{R}$consist of$f_{pre}\left(S^t\right)$;
        Execute $a_t$ and obtain $r_t$ and $S^t + \delta t$;
        Store tuple $\left(S^t, a^t, r^t, S^{t+\Delta t}\right)$ in E;
        Update $f_{val}$ by minimizing $L_2 = ||f_{val}\left(S_i\right) - y_i||$;
    Update target value network $\hat{f}_{val} \leftarrow f_{val}$;
return
```

IV. Experimental Results

To evaluate the performance of pedestrian velocity field prediction and the impact of social norms on navigation, we completed three parts of experiments: (1) analyzing the spatiotemporal characteristics of the velocity field in the walking pedestrians dataset [37]; (2) evaluating the accuracy of the predicted pedestrian velocity field using the walking pedestrians dataset. [37]; (3) using Gazebo simulation [38] and the PEDSIM library [39] to evaluate the improvement of navigation performance under social norm constraints.

A. Simulation Setup

For the simulated tests in navigation experiment, We have selected four common scenarios, namely Lobby world, Autolab world, Cumberland world, and Freiburg world, so that we can easily compare with other navigation methods. The corresponding environment map is same as [31]. We use Turtlebot 2 as a socially navigating robot model, equipped with cameras and lidar. The socially-aware robot's maximum velocity is limited to an average of $0.5m/s$, corresponding to typical human walking speeds. PEDSIM is the most commonly used pedestrian plug-in for simulating pedestrian environments. It is based on the microscopic social forces model [40] to simulate pedestrian-pedestrian and pedestrian-environment interactions. In addition, the number of pedestrians in our control environment is 25. We evaluate the performance of the proposed method against several state-of-the-art baselines, including a CNN-based policy[36], other deep reinforcement learning (DRL) based policies(DRL [31], A1-RC and A1-RD [41]), and a traditional motion planning algorithm(DWA planner [42]).

B. Verify the Gaussian Divergence Theorem at the Micro Scale

Gaussian divergence theorem is the most basic theorem in field theory. They are the basis for building more complex metrics, theorems, etc. It is important to prove that vectors obtained from walking pedestrians datasets satisfy this Gaussian divergence theorem. The formula of Gaussian divergence theorem is as follows:

$$\phi_S = \oint d\ell \overrightarrow{\mathbf{n}} \overrightarrow{\mathbf{V}} = \int dS \nabla \overrightarrow{\mathbf{V}} = \phi_W \tag{11}$$

Where ϕ_S is the surface integral, $\overrightarrow{\mathbf{n}}$ is the unit vector at each point, $d\ell$ is the infinitesimal of length. $\overrightarrow{\mathbf{V}}$ is the average

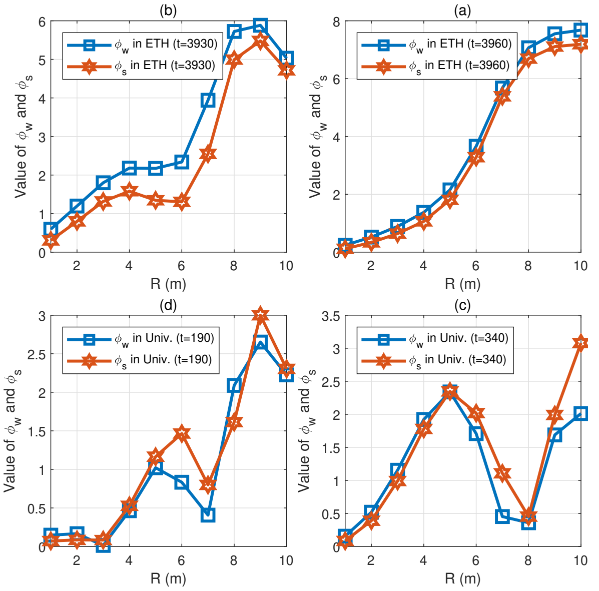

Fig. 4. Verification of Gauss's theorem. The red dashed line represents the divergence of pedestrian movement fields in different datasets under different grid cell sizes, while the blue dashed line represents the vorticity of pedestrian movement fields in different datasets under different grid cell sizes.

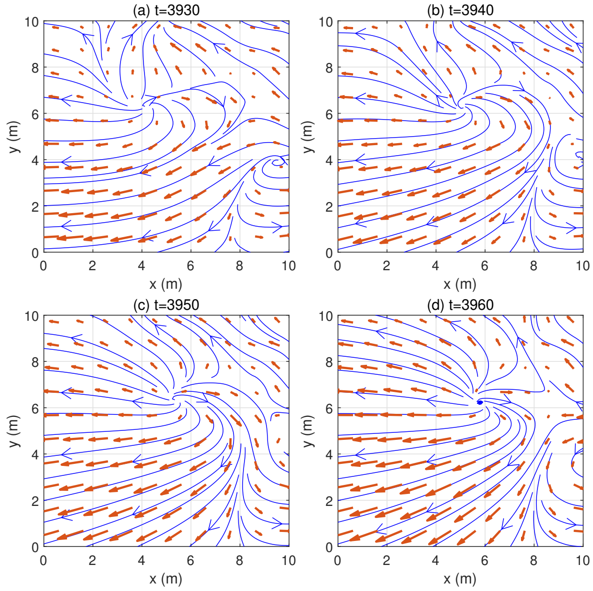

Fig. 5. Visualization of crowd field in continuous time. The red arrow represents the vector of the crowd field, and the blue line represents the streamline of the crowd field. It can be seen that the crowd field changes slowly over time.

velocity vector. ϕ_W is the volume integral of the divergence of $\overrightarrow{\mathbf{V}}$. dS is the infinitesimal of area.

It has been verified in [8] that the mobility of the crowd on the macro scale (km) satisfies the Gaussian divergence theorem. We need to verify that the mobility of the crowd still satisfies the Gaussian divergence theorem at the micro scale (m) walking pedestrians, in order to illustrate the rationality of our use of field models to model walking pedestrians. This is the basis for us to use field theory and PINN. We evaluated real pedestrian movement data from [37], and selected pedestrian movement data at different times and locations (ETH and Univ.). Verify whether the crowd field satisfies the Gaussian divergence theorem by calculating ϕ_w and ϕ_s of the crowd field in square areas with different side lengths R. The results are shown in Fig. 4. It can be seen that as the side length R of the square area changes, the ϕ_w and ϕ_s of the crowd field also change and are consistent with $\phi_w \approx \phi_s$. Excluding the deviations caused by numerical calculations, it can be concluded that **the mobility of pedestrians at the micro scale (m) satisfies the Gaussian divergence theorem, so field theory and PINN can be used to model the crowd field.**

C. Spatio-temporal Characteristics of Crowd Field

The same area often presents different functional attributes at different times. The functional attributes of this area will cause the pedestrian speed vector field to show completely different characteristics. Therefore, the pedestrian speed vector field prediction network needs to be continuously trained and updated over time to provide accurate prediction speed. Too frequent training will greatly increase the complexity of the model. We need to analyze how quickly the pedestrian speed vector field changes over time, and analyze the feasibility of using the pedestrian speed vector prediction module to provide data for the reward function.

Fig. 5 shows that **the crowd field is time-varying and changes slowly**. This requires the field-theory-inspired social norms model to learn the latest crowd field in a timely manner. Fortunately, subsequent experiments proved that the time required for the Field-theory-inspired social norms model to learn the latest crowd field is less than the time for the crowd field to make a significant difference. Therefore, the field-theory-inspired social norms model can be updated to the latest crowd field in time based on the perceived data.

D. Quantitative Evaluations for Crowd Field Prediction

We used data from [37] to conduct a quantitative evaluation of crowd field prediction accuracy. The crowd field prediction network based on PINN we established is a neural network with fully connected structure, the learning rate during training is 0.01, and the batch size is 64. The visualization comparison between the pedestrian motion vector predicted by the crowd field and the real pedestrian motion vector is shown in Fig. 6.

It can be seen from Fig. 6 that the pedestrian motion vector predicted by the crowd field is very close to the real pedestrian motion vector, indicating that the crowd field prediction network can effectively capture the field characteristics of pedestrian motion. Numerically, after training for 3000 rounds based on the pre-trained model, the MSE error between the predicted crowd vector field and the true value is 0.065.

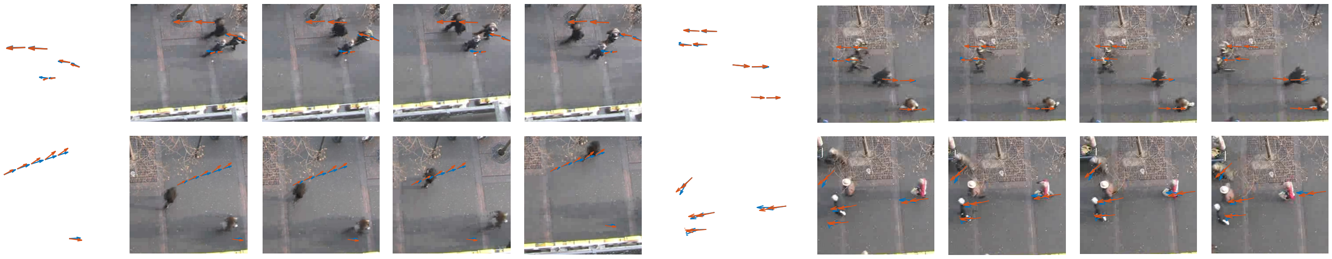

Fig. 6. Crowd field prediction result. The red arrow represents the predict pedestrian speed vector of current location, and the blue arrow represents the true pedestrian speed vector. The higher the coincidence between the blue arrow and the red arrow, the more accurate the prediction result is.

TABLE I

Evaluation of Navigation Performance in Four Distinct Environments with 25 Pedestrians

Environment	Method	SR[1](↑)	AT[2](↓)	AL[3](↓)	AS[4](↑)	Environment	Method	SR[1](↑)	AT[2](↓)	AL[3](↓)	AS[4](↑)
Lobby World 25 pedestrains	DWA	0.82	13.49	**5.12**	0.38	Autolab World 25 pedestrains	DWA	0.88	13.50	**5.47**	0.41
	CNN	0.80	19.31	6.16	0.32		CNN	0.78	21.59	6.61	0.31
	A1-RD	**0.90**	17.87	6.07	0.34		A1-RD	0.88	18.92	6.54	0.35
	A1-RC	0.88	14.18	6.26	0.44		A1-RC	0.77	15.63	7.35	0.43
	DRL	0.81	13.73	6.24	0.45		DRL	0.75	15.66	7.03	0.45
	Ours	0.85	**12.57**	5.87	**0.47**		Ours	**0.89**	**12.35**	6.03	**0.49**
Cumberland World 25 pedestrains	DWA	0.78	15.16	5.53	0.37	Freiburg World 25 pedestrains	DWA	**0.83**	13.90	5.75	0.41
	CNN	0.63	27.80	7.73	0.28		CNN	0.48	30.38	8.30	0.27
	A1-RD	**0.91**	17.47	6.63	0.38		A1-RD	0.81	17.95	6.61	0.37
	A1-RC	0.81	16.31	6.93	0.43		A1-RC	0.74	17.06	7.35	0.43
	DRL	0.66	14.28	6.38	0.45		DRL	0.53	15.68	7.06	0.45
	Ours	0.80	**11.93**	**5.44**	**0.46**		Ours	0.79	**11.96**	**5.73**	**0.48**

[1] Success Rate. [2] Average Time. [3] Average Length. [4] Average Speed.

E. *Quantitative Evaluations for Crowded Navigation*

We test our social navigation policies and selected four of the most commonly used evaluation indicators in social navigation for testing, which are **Success Rate (SR)**, **Average Time (AT)**, **Average Length (AL)**, **Average Speed (AS)**. Table I summarizes the results of our evaluation. As can be seen, our approach primarily serves as an additional component in the reward function of reinforcement learning. For the baseline reinforcement learning method, the four key performance indicators mentioned above demonstrate improvements compared to the alternative approaches. This fully proves that the constraints of social norms can not only improve pedestrian comfort at the social level, but also improve the efficiency of task completion in social navigation tasks.

We analyze each indicator in detail. Compared with other non-reinforcement learning, the navigation strategy based on reinforcement learning has no advantage in success rate. However, in the Autolab scenario, the reinforcement learning method that adds social norm constraints achieved the highest success rate, proving the improvement of reinforcement learning brought by mining social signals, and also confirming the huge potential of reinforcement learning navigation strategies.

In terms of average time, average length, and average speed, which are indicators that reflect the quality of a navigation task, our reinforcement learning method with social norm constraints has almost achieved the best performance. Social norms encode semantic environmental information and prevalent social rules to a degree. Experiments have proven that robots can find better navigation strategies with the help of information about social norms.

V. Conclusion

Our article applies field theory to the modeling of pedestrian motion in social navigation for the first time, and introduces the concept of social rules to constrain the reinforcement learning strategy for controlling robot navigation. Experiments have verified that the pedestrian motion dataset aligns with the Gaussian divergence theorem and can be characterized as a field-based representation. Moreover, the changes in the pedestrian field are continuous in the time dimension and change very slowly, which is suitable for providing social rule information during the navigation process. Experiments have shown that the performance of navigation strategies incorporating social rule constraints has been improved to a certain extent.

VI. Acknowledgment.

This work received partial support from the National Natural Science Foundation of China (62172443), the Natural Science Foundation of Hunan Province (2022JJ30760), and the Natural Science Foundation of Changsha (kq2202107, kq2202108). We are grateful for resources from the High Performance Computing Center of Central South University.

References

[1] C. Mavrogiannis, F. Baldini, A. Wang, D. Zhao, P. Trautman, A. Steinfeld, and J. Oh, "Core challenges of social robot navigation: A survey," *ACM Transactions on Human-Robot Interaction*, vol. 12, no. 3, pp. 1–39, 2023.

[2] A. Francis, C. Pérez-d'Arpino, C. Li, F. Xia, A. Alahi, R. Alami, A. Bera, A. Biswas, J. Biswas, R. Chandra *et al.*, "Principles and guidelines for evaluating social robot navigation algorithms," *arXiv preprint arXiv:2306.16740*, 2023.

[3] C. Chen, S. Hu, P. Nikdel, G. Mori, and M. Savva, "Relational graph learning for crowd navigation," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 10 007–10 013.

[4] T. Randhavane, A. Bera, E. Kubin, A. Wang, K. Gray, and D. Manocha, "Pedestrian dominance modeling for socially-aware robot navigation," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 5621–5628.

[5] E. Cheung, A. Bera, E. Kubin, K. Gray, and D. Manocha, "Identifying driver behaviors using trajectory features for vehicle navigation. in 2018 ieee," in *RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3445–3452.

[6] Y. F. Chen, M. Everett, M. Liu, and J. P. How, "Socially aware motion planning with deep reinforcement learning," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 1343–1350.

[7] R. Mirsky, X. Xiao, J. Hart, and P. Stone, "Prevention and resolution of conflicts in social navigation–a survey," *arXiv preprint arXiv:2106.12113*, 2021.

[8] M. Mazzoli, A. Molas, A. Bassolas, M. Lenormand, P. Colet, and J. J. Ramasco, "Field theory for recurrent mobility," *Nature communications*, vol. 10, no. 1, p. 3895, 2019.

[9] J. Van den Berg, M. Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," in *2008 IEEE international conference on robotics and automation*. Ieee, 2008, pp. 1928–1935.

[10] J. Van Den Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *Robotics Research: The 14th International Symposium ISRR*. Springer, 2011, pp. 3–19.

[11] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A. Y. Ng *et al.*, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.

[12] P. T. Singamaneni, A. Favier, and R. Alami, "Human-aware navigation planner for diverse human-robot interaction contexts," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 5817–5824.

[13] M. Pfeiffer, M. Schaeuble, J. Nieto, R. Siegwart, and C. Cadena, "From perception to decision: A data-driven approach to end-to-end motion planning for autonomous ground robots," in *2017 ieee international conference on robotics and automation (icra)*. IEEE, 2017, pp. 1527–1533.

[14] L. Tai, S. Li, and M. Liu, "A deep-network solution towards model-less obstacle avoidance," in *2016 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2016, pp. 2759–2764.

[15] A. Loquercio, A. I. Maqueda, C. R. Del-Blanco, and D. Scaramuzza, "Dronet: Learning to fly by driving," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1088–1095, 2018.

[16] G. Kahn, P. Abbeel, and S. Levine, "Badgr: An autonomous self-supervised learning-based navigation system," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1312–1319, 2021.

[17] A. Mohamed, K. Qian, M. Elhoseiny, and C. Claudel, "Social-stgcnn: A social spatio-temporal graph convolutional neural network for human trajectory prediction," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 14 424–14 432.

[18] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social lstm: Human trajectory prediction in crowded spaces," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 961–971.

[19] G. S. Aoude, B. D. Luders, J. M. Joseph, N. Roy, and J. P. How, "Probabilistically safe motion planning to avoid dynamic obstacles with uncertain motion patterns," *Autonomous Robots*, vol. 35, pp. 51–76, 2013.

[20] M. Bennewitz, W. Burgard, G. Cielniak, and S. Thrun, "Learning motion patterns of people for compliant robot motion," *The International Journal of Robotics Research*, vol. 24, no. 1, pp. 31–48, 2005.

[21] Y. F. Chen, M. Liu, M. Everett, and J. P. How, "Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning," in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 285–292.

[22] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner *et al.*, "Relational inductive biases, deep learning, and graph networks. arxiv 2018," *arXiv preprint arXiv:1806.01261*, 2018.

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[24] C. Chen, Y. Liu, S. Kreiss, and A. Alahi, "Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning," in *2019 international conference on robotics and automation (ICRA)*. IEEE, 2019, pp. 6015–6022.

[25] Y. Chen, C. Liu, B. E. Shi, and M. Liu, "Robot navigation in crowds by graph convolutional networks with attention learned from human gaze," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2754–2761, 2020.

[26] W. Gao, D. Hsu, W. S. Lee, S. Shen, and K. Subramanian, "Intention-net: Integrating planning and deep learning for goal-directed autonomous navigation," in *Conference on robot learning*. PMLR, 2017, pp. 185–194.

[27] Z. Shiller, Y.-R. Gwo *et al.*, "Dynamic motion planning of autonomous vehicles," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 2, pp. 241–249, 1991.

[28] J.-P. Laumond, P. E. Jacobs, M. Taix, and R. M. Murray, "A motion planner for nonholonomic mobile robots," *IEEE Transactions on robotics and automation*, vol. 10, no. 5, pp. 577–593, 1994.

[29] J. Z. Kolter, M. P. Rodgers, and A. Y. Ng, "A control architecture for quadruped locomotion over rough terrain," in *2008 IEEE International Conference on Robotics and Automation*. IEEE, 2008, pp. 811–818.

[30] X. B. Peng, G. Berseth, K. Yin, and M. Van De Panne, "Deeploco: Dynamic locomotion skills using hierarchical deep reinforcement learning," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, pp. 1–13, 2017.

[31] Z. Xie and P. Dames, "Drl-vo: Learning to navigate through crowded dynamic scenes using velocity obstacles," *IEEE Transactions on Robotics*, 2023.

[32] R. C. Coulter *et al.*, *Implementation of the pure pursuit path tracking algorithm*. Carnegie Mellon University, The Robotics Institute, 1992.

[33] T. Kucner, J. Saarinen, M. Magnusson, and A. J. Lilienthal, "Conditional transition maps: Learning motion patterns in dynamic environments," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 1196–1201.

[34] U. H. Coskun, B. Sel, and B. Plaster, "Magnetic field mapping of inaccessible regions using physics-informed neural networks," *Scientific Reports*, vol. 12, no. 1, p. 12858, 2022.

[35] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous systems," 2015.

[36] Z. Xie, P. Xin, and P. Dames, "Towards safe navigation through crowded dynamic environments," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 4934–4940.

[37] S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool, "You'll never walk alone: Modeling social behavior for multi-target tracking," in *2009 IEEE 12th international conference on computer vision*. IEEE, 2009, pp. 261–268.

[38] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ international conference on intelligent robots and systems (IROS)(IEEE Cat. No. 04CH37566)*, vol. 3. IEEE, 2004, pp. 2149–2154.

[39] T. Kretz, "Pedestrian traffic: simulation and experiments," Ph.D. dissertation, Duisburg, Essen, Univ., Diss., 2007, 2007.

[40] D. Helbing and P. Molnar, "Social force model for pedestrian dynamics," *Physical review E*, vol. 51, no. 5, p. 4282, 1995.

[41] R. Guldenring, M. Görner, N. Hendrich, N. J. Jacobsen, and J. Zhang, "Learning local planners for human-aware navigation in indoor environments," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 6053–6060.

[42] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.