2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

SPLC: Social Preference Learning for Crowd Robot Navigation

Zixuan Chen, Hao Fu*, Haiwen Hu, Shiquan Zheng

Abstract— **Offline reinforcement learning (RL) holds significant potential for crowd robot navigation in human-robot coexistence applications. However, the inherent complexity of pedestrian motion renders the design of effective reward functions for promoting socially compliant robot behaviors a persistent challenge. This paper proposes a Social Preference Learning for Crowd Robot Navigation (SPLC) algorithm to eliminate the need for detailed reward design. Its core innovation lies in the introduction of a social preference feedback mechanism to automatically generate preference data through principled preference evaluation criteria. By explicitly accounting for the intricacies of pedestrian dynamics, the pipeline mitigates the reward bias and facilitates the systematic quantification of broad social norms, thereby fostering socially compliant behaviors. Extensive experiments integrating SPLC with offline RL methods demonstrate consistent improvements over state-of-the-art baselines across standard performance metrics. Furthermore, real-world experiments on the TurtleBot4 further validate the effectiveness of SPLC in practical human–robot coexistence settings. Our code and video demos are available at https://github.com/sklus949/SPLC.**

I. INTRODUCTION

Mobile robots are becoming increasingly prevalent in various engineering applications, such as autonomous driving and logistics distribution, which often involve pedestrian-rich environments. A key enabling technology for these applications is crowd robot navigation, aiming to avoid potential collisions and reach the target in minimal time. However, the inherent unpredictability and often uncooperative dynamics of pedestrian behavior pose serious threats to the safety of human–robot interaction.

Despite these challenges, research on crowd robot navigation has achieved substantial progress. Existing solutions can be broadly classified into three categories. One category encompasses reactive approaches, such as the Social Force Model (SFM) [1], Reciprocal Velocity Obstacles (RVO) [2], and Optimal Reciprocal Collision Avoidance (ORCA) [3], which determine the robot's optimal actions based on physical and geometric interaction rules. Another line of work focuses on predicting pedestrians' future trajectories before planning the path [4] [5]. However, both approaches are prone to the robot freezing problem in highly dynamic crowd environments.

This work was supported in part by the National Natural Science Foundation of China under Grant 62303357 and Grant 62173262 and in part by the Hubei Provincial Natural Science Foundation of China under Grant 2023AFB109.

*Corresponding author: Hao Fu

Z. Chen, H. Fu, H. Hu, and S. Zheng are with the School of Computer Science and Technology, Wuhan University of Science and Technology and also with the Hubei Province Key Laboratory of Intelligent Information Processing and Real-time Industrial System, Wuhan 430081, China (e-mail: fuhao@wust.edu.cn).

Another approach is deep reinforcement learning (DRL) [6], which holds great potential for crowd robot navigation and can overcome the limitations of the above methods. Owing to its inherent interactive nature, DRL-based crowd robot navigation has achieved substantial progress. In this process, many deep learning techniques, such as long short-term memory networks (LSTM) [7], self-attention mechanisms [8], and Transformers [9], have played important roles in encoding interactive feature information. However, the deployment of online DRL in real-world crowd navigation remains hindered by substantial challenges, particularly the high safety risks and considerable costs associated with human-in-the-loop training. To mitigate these limitations, offline RL has attracted increasing attention as a viable alternative. Representative algorithms, including Implicit Q-Learning (IQL) [10], Conservative Q-Learning (CQL) [11], and TD3BC [12], have demonstrated significant potential in autonomous driving and robotics applications [13], [14]. Building on these foundations, recent efforts [15], [16] have begun to explore the integration of offline RL into crowd robot navigation, highlighting a promising direction for advancing both the safety and practicality of socially compliant robot systems.

Despite significant progress about the Offline RL-based crowd robot navigation, the design of its reward function remains a primary challenge. Existing approaches predominantly rely on manually handcrafted reward functions, exhibiting an inherent limitation. Specifically, the absence of a principled quantification of broad social norms in the reward functions constrains the robot's capacity to exhibit socially compliant behaviors. For instance, in the complex crowd scenario, a handcrafted reward function may successfully guide the robot around a single individual pedestrian but fail to anticipate the collective flow of a crowd or recognize socially attentive behaviors like following a group's implicit lanes, thereby producing unnatural, myopic, or socially undesirable behaviors.

Motivated by the above challenge, this paper proposes a **S**ocial **P**reference **L**earning for **C**rowd robot navigation (SPLC). This algorithm learns a reward function by leveraging trajectory preference labeling, avoiding manual design of complex rewards in the crowd robot navigation. The learned reward model is then employed in some offline RL methods, such as IQL, CQL, and TD3BC. The main contributions of this paper can be summarized as:

- The proposed SPLC integrates a social preference feedback mechanism with a preference transformer to model preference rewards, thereby ensuring that the robot exhibits socially compliant behaviors.

- This paper introduces a social preference feedback mechanism to automatically generate preference labels via preference evaluation criteria. The criteria mitigate the reward bias by explicitly incorporating the inherent unpredictability and often uncooperative dynamics of pedestrian motion, thereby contributing to the quantification of broad social norms.
- We evaluate our SPLC integrated with standard offline RL methods. Experimental results demonstrate that our approach significantly outperforms the baselines.

II. RELATED WORKS

A. Crowd Robot Navigation

Early research on crowd robot navigation primarily focused on decoupled models, treating pedestrians as autonomous agents with independent motion patterns or as dynamic obstacles [17], [18]. Some existing studies [19], [20] have identified two main limitations of these approaches: they neglect human–robot interaction modeling, which can cause robot "freezing," and they depend on human modeling, preventing autonomous adaptation to complex environments. With its increasing popularity, researchers have started shifting their focus toward learning-based approaches. Liu *et al.* [21] proposed decentralized structural-Recurrent Neural Network (DS-RNN), which was integrated with DRL to perform spatiotemporal reasoning in crowd navigation. Mun *et al.* [22] introduced an occlusion-aware DRL framework, integrating social occlusion inference with a variational autoencoder for safer navigation. Additionally, the integration of attention mechanisms with RL has gained traction in recent studies [23], [24], [25] to enhance mobile robot navigation strategies.

Solving the problem of crowd robot navigation using RL requires an appropriate reward function to guide the robot's movement. Initially, the reward function used sparse rewards, rewarding task completion while penalizing collisions or uncomfortable distances, with zero rewards in other situations [7], [8]. Due to the lack of reward signals, this approach often leads to inefficient learning. Later research introduced potential-based rewards [21], [22], which guided the robot towards the goal by calculating the goal distance. Although this addressed the issue of sparse rewards, over-reliance on the distance may result in unnatural or myopic behaviors. More recent efforts have explored carefully engineered, scenario-specific reward functions [23], [26], achieving improved navigation performance but at the cost of extensive manual design effort. In contrast, this work advances the field by employing the proposed SPLC algorithm to model the reward for crowd-robot navigation, thereby reducing the dependence on handcrafted designs and enabling more adaptive and socially compliant behaviors.

B. Preference-based reinforcement learning

Preference-based Reinforcement Learning (PbRL) is an RL approach that learns an implicit reward function from comparative feedback on different trajectories provided by humans or experts, rather than relying on manually designed explicit reward signals [27]. In this way, PbRL is able to capture human preferences and values, thereby guiding the agent to learn behaviors that better align with desired outcomes [28], [29]. As one of the core technologies driving ChatGPT 3, this approach has attracted extensive research interest and attention [30]. Several studies [31], [32], [33] have successfully employed online PbRL to learn reward functions, which enable agents to act in accordance with human preferences. However, online PbRL requires continuously querying humans for preference feedback during training, which incurs a substantial human feedback cost. To overcome this limitation, offline preference-based learning has garnered significant research interest. Shin *et al.* [34] applied PbRL to offline RL, establishing a series of benchmarks and algorithms for offline PbRL. Kim *et al.* [35] proposed a novel Transformer model that captures human preferences via non-Markovian reward-weighted modeling. Li *et al.* [36] proposed Scaling Preference, which allows humans to express the strength of preferences between trajectories, enabling more accurate reward learning from offline data. Nevertheless, even in offline RL, due to the complexity of real-world tasks and the inherent subjectivity of human annotators, humans often struggle to provide sufficient and accurate preference feedback. To address this issue, we introduce a social preference feedback mechanism that automatically generates preference labels based on preference evaluation criteria.

III. METHODOLOGY

A. Problem Formulation

The robot crowd navigation problem can be formulated as a sequential decision-making task. In this paper, we model it as a Partially Observable Markov Decision Process (POMDP), defined as a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, R, \Omega, \mathcal{O}, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{T}$ is the state transition probability, R is the reward function, Ω is the observation space, $\mathcal{O}$ is the observation distribution, and $\gamma \in (0, 1]$ is the discount factor.

1) *State space $\mathcal{S}$:* In a crowd robot navigation environment, the state space at each time step consists of observable and unobservable states of the agent (robot and humans). The observable states comprise the agent's position $\mathbf{p} = [p_x, p_y]$, velocity $\mathbf{v} = [v_x, v_y]$ and radius $\bar{r}$, while the unobservable states include the target position $\mathbf{p}_g = [g_x, g_y]$, preferred velocity v_{pref} and heading angle θ. To enhance the generality of the state representation, this paper adopts the robot-centric state representation [37]. The transformed states of the robot and pedestrian are:

$$\begin{aligned} s^r &= [d_g, v_{pref}, \bar{r}, v_x, v_y, \theta] \\ s^i &= [\tilde{p}_x^i, \tilde{p}_y^i, \tilde{v}_x^i, \tilde{v}_y^i, \bar{r}^i, d^i, \bar{r} + \bar{r}^i] \end{aligned} \tag{1}$$

where s^r and s^i represent the transformed states of the robot and the i-th pedestrian, respectively. $d_g = \|\mathbf{p}_g - \mathbf{p}\|$ is the distance from the robot to the target position, and $d^i = \|\mathbf{p} - \mathbf{p}_i\|$ is the distance from the

robot to pedestrian i. Then, the joint state can be represented as $s_t^{jn} = [s_t^r, s_t^1, s_t^2, \ldots]$ at the t-th time step.

2) *Action space $\mathcal{A}$:* In social scenarios, designing an appropriate action space helps robots generate smooth trajectories. Considering the kinematic specifications of real robots and their practical applications, this paper defines the robot's actions as:

$$a_t = (v_t, \omega_t) \tag{2}$$

where $v_t \in (0, 1)$ is the translational velocity and $\omega_t \in (-1, 1)$ is the rotational velocity.

3) *Optimization objective:* The objective of crowd robot navigation is to reach the target in the shortest possible time while avoiding potential collisions, thereby accomplishing the navigation task in an efficient and safe manner. Within the MDP framework, the objective is generally formulated via reward modeling, with the optimal policy expressed as:

$$\pi^* = \arg\max_{\pi} \sum_t \gamma^t r_t \tag{3}$$

where γ is a discount factor, $\pi : \mathcal{S} \to P(\mathcal{A})$ denotes the policy, and r_t is the reward at timestep t.

B. Social Preference Learning

The central challenge in achieving effective crowd robot navigation lies in the elaborate design of a reward function capable of guiding the learning process toward an optimal policy. In fact, handcrafted reward functions often struggle to quantify broad social norms. As a result, RL-based on such manually specified rewards frequently leads to suboptimal navigation policies, ultimately limiting the robot's ability to exhibit socially compliant behaviors. To address this issue, this paper proposes the SPLC algorithm to circumvent manual reward engineering by leveraging trajectory preference labeling. Its concrete overall framework is illustrated in Fig. 1.

PbRL typically learns a reward function from human-labeled preference data prior to the RL training phase. In this paradigm, a dataset annotated with human preference feedback is first collected and then used to guide the subsequent learning process. Nevertheless, this pipeline is associated with substantial annotation costs, as it requires extensive human labor. On the other hand, the inherent unpredictability and often uncooperative dynamics of pedestrian motion aggravate the reward bias, resulting from the inherent subjectivity of human annotators.

To overcome these limitations, this paper introduces a social preference feedback mechanism to automatically generate preference labels for each sampling instance, consisting of a pair of trajectory segments. The trajectory segment $\sigma = \{(s_1^{jn}, a_1), \ldots, (s_L^{jn}, a_L)\} \in$ is sampled from the offline dataset $\mathcal{D}_s = \{s_t^{jn}, a_t, s_{t+1}^{jn}\}_{t=1}^{M_s}$ with the maximum capacity M_s, is a time-indexed sequence of joint states and actions within a specified length L. A pair of randomly sampled trajectory segments is represented as σ^0 and σ^1,

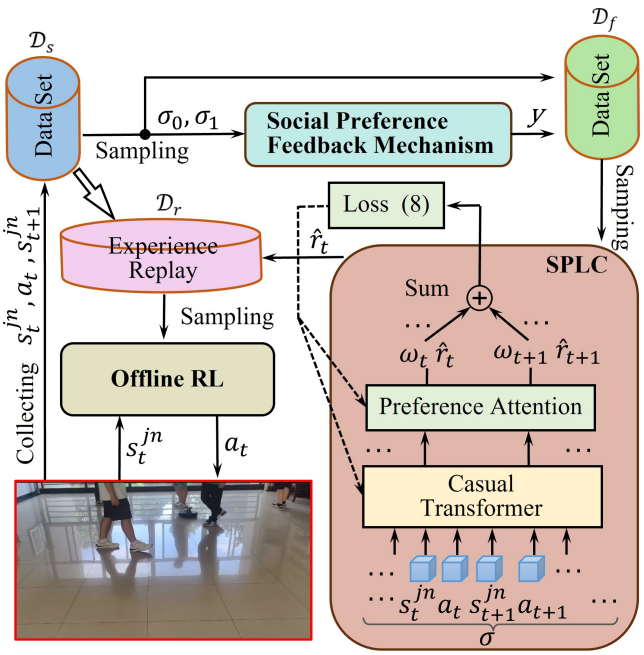


Fig. 1: The overall framework of our SPLC algorithm

respectively. The associated preference label y takes the form of a two-dimensional tuple, selected from the set $y \in \{(0.9, 0.1), (0.1, 0.9), (0.7-\alpha, 0.3+\alpha), (0.3-\alpha, 0.7+\alpha), (0.5-\alpha, 0.5+\alpha)\}$ with the preference risk α. To assign appropriate preference labels to σ^0 and σ^1, we define three preference evaluation criteria to determine preference labels by means of quantization of broad social norms. The criteria provide structured supervisory signals, and their integration with trajectory-level preference comparisons enables the implicit inference and capture of more subtle social behaviors.

Collision Occurrence: Indicates whether a collision between the robot and a pedestrian occurs within the trajectory segment. This serves as a direct measure of human–robot safety.

Goal Progress: Quantifies the robot's progress toward its goal, reflecting navigation efficiency. It is defined as

$$\eta = \|\mathbf{p}_1 - \mathbf{p}_g\| - \|\mathbf{p}_L - \mathbf{p}_g\|, \tag{4}$$

where $\mathbf{p}_1$ and $\mathbf{p}_L$ denote the start and end robot positions of the trajectory segment σ, respectively.

Risk Exposure: Serves as a supplement to *Goal Progress* and measures the frequency of risk events, where a risk is defined as the robot entering unsafe proximity to a pedestrian. This criterion is given by

$$\mu = \frac{1}{L} \Sigma_{t=1}^{L} \xi_t, \tag{5}$$

where ξ_t is a binary indicator of the danger zone [38] of a pedestrian at time step t.

Owing to the fundamental importance of these segment evaluation criteria in the crowd robot navigation, we impose a lexicographic priority order on the mechanism used to determine the preference label y from the 2-D tuple. The ordinal

hierarchy is specified as: "*Collision Occurrence*"←"*Goal Progress*".

According to this ordering, the evaluation of *Collision Occurrence* has the highest priority. A segment σ^0 without a collision is always preferred over another segment σ^1 with one, i.e., $\sigma^0 \succ \sigma^1$, where $\succ$ indicates a preference operator. In this case, the preference label is set to $y = (0.9, 0.1)$. Conversely, a segment σ^1 without a collision is always preferred over another segment σ^0 with one, i.e., $\sigma^0 \prec \sigma^1$. Accordingly, the preference label becomes $y = (0.1, 0.9)$. If both segments about *Collision Occurrence* are identical, this mechanism proceeds to the next criterion, i.e., *Goal Progress*.

The criterion of *Goal Progress* is determined by the values of η^0 and η^1 as defined in Equation (4). If $\eta^0 > \eta^1$, σ^0 is more preferable to σ^1, i.e., $\sigma^0 \succ \sigma^1$. Afterward, we can obtain the preference label $y = (0.7 - \alpha, 0.3 + \alpha)$. Conversely, if $\eta^0 < \eta^1$, the preference is reversed, resulting in $\sigma^0 \prec \sigma^1$ and $y = (0.3 - \alpha, 0.7 + \alpha)$. If both segments are identical with respect to *Goal Progress*, indicating no differential preference between the segments, a neutral label is assigned as $y = (0.5 - \alpha, 0.5 + \alpha)$.

Risk Exposure, as a supplement to *Goal Progress*, calculates μ^0 and μ^1 for each segment according to (5). Then, we can have the preference risk

$$\alpha = 0.1 \cdot \tanh\left(\frac{\mu^0 - \mu^1}{\mu_{\max} - \mu_{\min}}\right), \tag{6}$$

where $\mu_{\max}$ and $\mu_{\min}$ denote the maximum and minimum numbers of risk events within the segments in the dataset, respectively.

To obviate the need for designing a handcrafted reward-need, the trajectory segments and the preference labels are used to train a reward model, serving as the downstream offline RL robot navigation task. As a result, we store the segments (σ^0, σ^1), and the preference label y in the preference dataset $\mathcal{D}_f = \{\sigma_i^0, \sigma_i^1, y_i\}_{i=1}^{M_f}$, where M_f represents the size of the dataset.

Intuitively, socially compliant decisions inherently rely on its temporal information. Consequently, its temporal information has a role to play in designing reward functions. To this end, this paper employs the preference transformer to learn the reward function from the preference dataset $\mathcal{D}_f$. The trajectory segment σ is fed into the preference transformer to generate the weighted sum of non-Markovian rewards $\sum_\sigma w_{\psi,t}\hat{r}_{\psi,t}$ and the preference reward prediction sequences $[\hat{r}_{\psi,1}, \hat{r}_{\psi,2}, \ldots, \hat{r}_{\psi,L}]$, where ψ is the trained parameter and $w_{\psi,t}$ is the importance weight. According to the Bradley-Terry model [39], we can model a preference reward predictor as follows:

$$P_\psi(\sigma^1 \succ \sigma^0) = \frac{\exp(\sum_{\sigma^1} w_{\psi,t}\hat{r}_{\psi,t})}{\exp(\sum_{\sigma^1} w_{\psi,t}\hat{r}_{\psi,t}) + \exp(\sum_{\sigma^0} w_{\psi,t}\hat{r}_{\psi,t})}. \tag{7}$$

Subsequently, through the preference dataset $\mathcal{D}_f$, the reward function $\hat{r}_{\psi,t}$ is updated by minimizing binary cross-entropy loss:

$$\mathcal{L}(\psi) = -\mathbb{E}_{(\sigma^0,\sigma^1,y)\in\mathcal{D}_f}(y(0)\log P_\psi(\sigma^0 \succ \sigma^1) + y(1)\log P_\psi(\sigma^0 \prec \sigma^1)). \tag{8}$$

Remark 1: Our proposed SPLC algorithm models the preference reward through a novel integration of a social preference feedback mechanism and the preference transformer. The social preference feedback mechanism is able to automatically generate preference labels without manual labeling by annotators through the defined preference evaluation criteria. In addition, the preference evaluation criteria can eliminate the impact on the inherent unpredictability and often uncooperative dynamics of pedestrian motion, thereby mitigating the reward bias. Consequently, our SPLC circumvents the challenges of manual reward engineering, enabling offline RL agents to acquire socially compliant behaviors.

Combining with the offline dataset $\mathcal{D}_s$ and the reward prediction $\hat{r}_{\psi,t}$ yields transitions $(s_t^{jn}, a_t, s_{t+1}^{jn}, \hat{r}_{\psi,t})$. These transitions are collected into an experience replay buffer $\mathcal{D}_r$ to optimize robot navigation policies via offline RL algorithms, such as IQL, CQL, and TD3BC.

The detailed implementation procedure for data collection is shown in Algorithm 1.

IV. EXPERIMENTS

A. Experiment Setup

1) *Simulation environment:* This paper builds a crowd robot navigation simulation platform where a robot can navigate across diverse crowd scenarios. In the experimental setup, six pedestrians are randomly initialized within a circular area of 4 meters in radius. Each pedestrian's initial and target positions are symmetrically distributed along the circumference of the circle. Their motion behavior is governed by ORCA [3], ensuring collision-free navigation and obstacle avoidance. Unlike fixed-goal settings, pedestrians in this environment are assigned new random destinations immediately upon reaching their current targets, thereby maintaining a dynamic and continuously evolving interaction space.
2) *Generating Datasets:* Our datasets are built within this simulation environment. To evaluate the performance of the reward model in complex environments, the dataset used in this paper is of medium level. Details of the dataset, five metrics, i.e., "*Success*", "*Collision*", "*Timeout*", "*Time*" and "*Capacity*" are shown in Table I. They describe the success rate, collision rate, timeout rate, average navigation time of success trajectories and datasets capacity, respectively.

TABLE I: Specific performance metrics of datasets

Success	Collision	Timeout	Time	Capacity
76.2%	23.6%	0.2%	11.7 s	5×10^5

Algorithm 1 SPLC

Input Offline dataset $\mathcal{D}_s$, preference dataset size M_f, epochs N_p and N_e
Initialize Preference dataset $\mathcal{D}_f$, replay buffer $\mathcal{D}_r$, parameter ψ
while $|\mathcal{D}_f| \leq M_f$ **do** ▷ Social Preference Feedback Mechanism
  Sample a pair of segments $(\sigma^0, \sigma^1) \sim \mathcal{D}_s$
  if *Collision occurrence*(σ^0, σ^1) =(*False*, *True*) **then**
    $\sigma^0 \succ \sigma^1$ and $y = (0.9, 0.1)$
  else if *Collision occurrence*(σ^0, σ^1) =(*True*, *False*) **then**
    $\sigma^0 \prec \sigma^1$ and $y = (0.1, 0.9)$
  else
    Calculate η^0 and η^1 via (4)
    Compute α in (5) and (6)
    if $\eta^0 > \eta^1$ **then**
      $\sigma^0 \succ \sigma^1$ and $y = (0.7 - \alpha, 0.3 + \alpha)$
    else if $\eta^0 < \eta^1$ **then**
      $\sigma^0 \prec \sigma^1$ and $y = (0.3 - \alpha, 0.7 + \alpha)$
    else
      $y = (0.5 - \alpha, 0.5 + \alpha)$
    end if
  end if
  Append (σ^0, σ^1, y) to $\mathcal{D}_f$
end while
for $Step = 1$ *to* N_p **do** ▷ Learn Reward Model
  Sample a random minibatch $(\sigma^0, \sigma^1, y) \sim \mathcal{D}_f$
  Calculate $P_\psi(\sigma^1 \succ \sigma^0)$ in (7)
  Optimize $\hat{r}_{\psi,t}$ with respect to ψ via Loss (8)
end for
for $Step = 1$ *to* N_e **do** ▷ Training Policy
  Sample a random mini-batch transition tuples $(s_t^{jn}, a_t, s_{t+1}^{jn}) \sim \mathcal{D}_s$
  Compute reward $\hat{r}_{\psi,t}$ via preference transformer
  Append $(s_t^{jn}, a_t, s_{t+1}^{jn}, \hat{r}_{\psi,t})$ to $\mathcal{D}_r$
  Update navigation policy with offline RL algorithm (e.g., IQL, CQL, TD3BC)
end for

3) *Baseline:* To rigorously evaluate the effectiveness of our proposed algorithm, we adopt three representative offline RL methods, including IQL, CQL and TD3BC. Furthermore, to provide a comprehensive comparison of our reward function, we benchmark it against three state-of-the-art alternatives: a traditional handcrafted reward function (HR) [22], a reward function derived from human-labeled preference data (HPR) [25], and a robust preference-based reward under corrupted preference modeling (RPR) [40].
4) *Training Settings:* The parameters related to preference queries and reward model are summarized in Table II. In the offline RL section, we adopt the implementations of these algorithms from the CORL [41] library.

TABLE II: Hyperparameters for training reward models

Hyperparameter	Value
Max trajectory segments length	100
Number of queries	2000
Query length	15
Number of layers	1
Number of attention heads	4
Embedding dimension	256
Batch size	256
Dropout rate	0.1
Learning rate	1e−4
Optimizer	AdamW
Weight decay	1e−4
Total gradient steps	1e4

B. Quantitative Evaluation

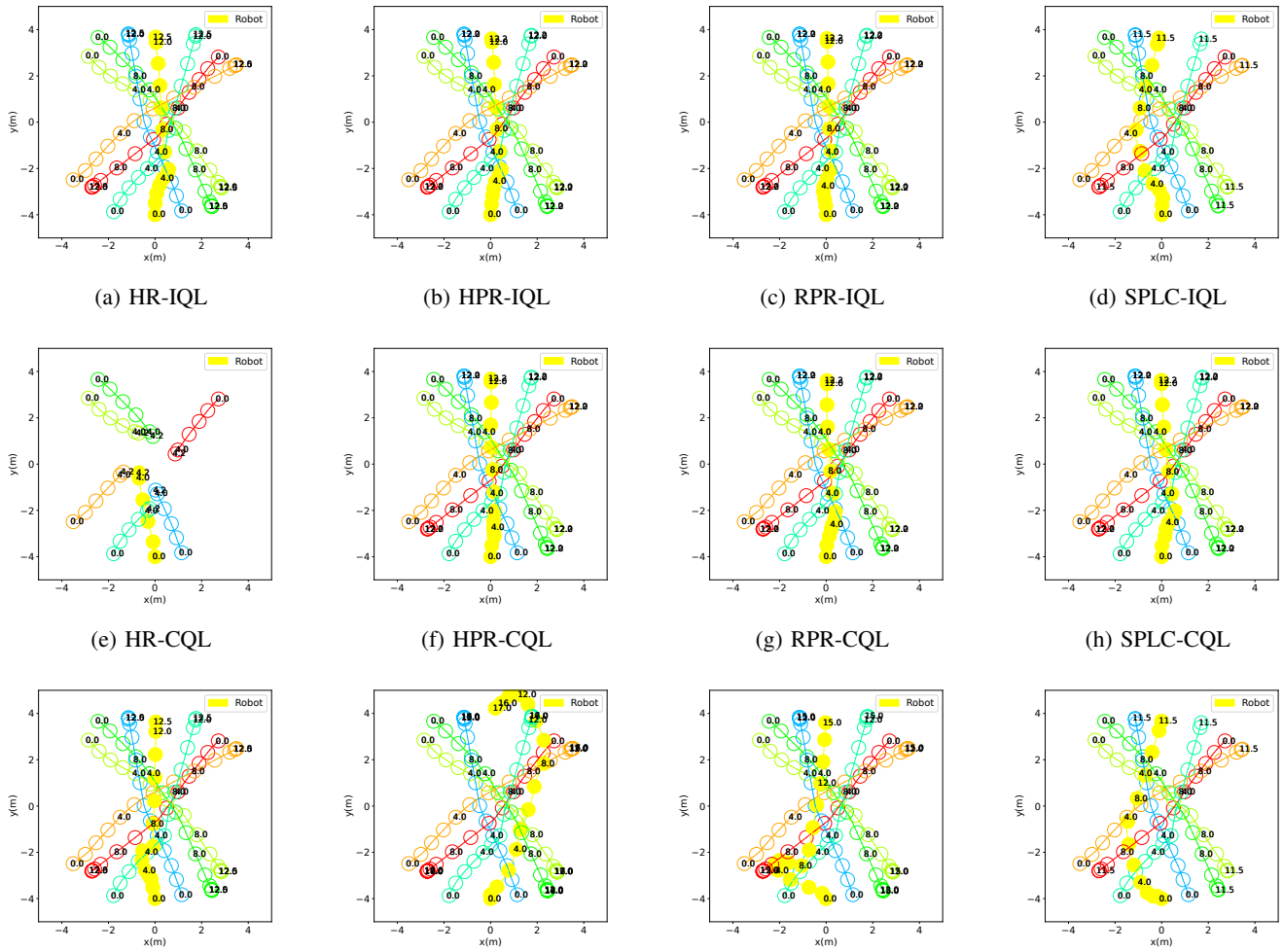
This section details a comprehensive evaluation comparison through qualitative analysis, spotlighting vital metrics in the context of the crowd robot navigation. These metrics involve "*Success*", "*Collision*", "*Timeout*" and "*Time*". In the qualitative evaluation, all methods are evaluated in 500 testing cases. Their comparison results are listed in Table III.

TABLE III: Quantitative results of all methods

Methods	Success	Collision	Timeout	Time
HR-IQL	93.40%	6.20%	0.40%	11.95 s
HPR-IQL	93.20%	6.80%	0.00%	11.10 s
RPR-IQL	92.20%	7.80%	0.00%	11.06 s
SPLC-IQL	94.60%	5.40%	0.00%	11.37 s
HR-CQL	36.40%	63.60%	0.00%	11.05 s
HPR-CQL	82.60%	17.40%	0.00%	10.86 s
RPR-CQL	86.20%	13.80%	0.00%	10.75 s
SPLC-CQL	95.40%	4.60%	0.00%	11.18 s
HR-TD3BC	82.40%	13.00%	4.60%	13.70 s
HPR-TD3BC	62.20%	7.00%	30.80%	16.74 s
RPR-TD3BC	79.00%	12.60%	8.40%	14.63 s
SPLC-TD3BC	90.60%	7.40%	2.00%	11.90 s

As shown in the first four rows of Table III, HR-IQL yields the longest navigation time, as its manual reward function fails to capture those subtle and unspoken social norms inherent in the crowd robot navigation and instead focuses only on local sparse information, thereby leading to myopic behaviors. Although HPR-IQL can implicitly infer desirable navigation intentions from human judgments, subjectivity of human annotators incurs the reward bias, ultimately resulting in a lower success rate. RPR-IQL shows similar performance, providing slightly more stable navigation without significantly changing success rate. In contrast, our SPLC-IQL achieves consistent improvements across all evaluation metrics, indicating that the social preference feedback mechanism provides a more faithful quantification of broad social norms and effectively mitigates reward bias, thereby enhancing navigation success rate and efficiency.

In the fifth to eighth rows of Table III, HR-CQL exhibits a very low success rate due to the inherent conservativeness of the algorithm combined with biases in handcrafted rewards. HPR-CQL improves both success rate and navigation time, but it still produces suboptimal behaviors due to insufficient



(a) HR-IQL (b) HPR-IQL (c) RPR-IQL (d) SPLC-IQL

(e) HR-CQL (f) HPR-CQL (g) RPR-CQL (h) SPLC-CQL

(i) HR-TD3BC (j) HPR-TD3BC (k) RPR-TD3BC (l) SPLC-TD3BC

Fig. 2: Comparison of robot trajectories using different methods in identical crowd robot navigation test scenarios. Yellow indicates the robot's trajectory, while other colors represent pedestrian trajectories.

reward accuracy. RPR-CQL demonstrates further improvements, yielding higher success rates and shorter navigation times. Notably, our SPLC-CQL increases the success rate to 95.40% while more effectively balancing efficiency and safety. The main reason is that the social preference feedback mechanism can effectively alleviate the reward bias, thereby making the CQL policy both more stable and effective.

In the last four rows of Table III, HR-TD3BC performs poorly in both success rate and navigation efficiency because handcrafted rewards fail to sufficiently amplify reward differences, keeping the policy in suboptimal behaviors. HPR-TD3BC performs even worse than handcrafted rewards, demonstrating that insufficient reward modeling cannot effectively guide the policy toward optimal behaviors. RPR-TD3BC improves over HPR-TD3BC by alleviating performance degradation, but remains inferior to handcrafted rewards. In contrast, our SPLC-TD3BC achieves the best performance across all metrics. This reveals that SPLC provides a more discriminative and reliable signal, effectively reducing reward bias and enabling the policy to overcome suboptimal behaviors and achieve superior navigation performance.

Overall, across the three offline RL methods, our SPLC algorithm better quantifies broad social norms and mitigates reward bias arising from the inherent unpredictability and often uncooperative dynamics of pedestrian motion.

C. Qualitative Evaluation

To qualitatively validate our algorithm, global trajectories from different methods are compared to visualize robot behaviors under a shared initial condition and target formation. Details are shown in Fig. 2.

As illustrated in Figs. 2a-2c, HR-IQL, HPR-IQL and RPR-IQL execute rightward turns at 4 seconds to avoid pedestrians, but subsequently enter densely crowded areas, resulting in slow subsequent movement. This reflects myopic and unnatural navigation caused by inaccurate rewards. In contrast, Fig. 2d shows that our SPLC-IQL proactively adjusts the path by selecting the opposite side, thereby steering

(a) start time (b) mid time (c) end time

Fig. 3: The three sub-figures sequentially show the robot's navigation process in the real-world experiment of our algorithm.

the robots toward a sparser and safer region and completing the navigation task more efficiently, which highlights that our SPLC better quantifies social norms and incorporates global awareness into the navigation policy, enabling the agent to anticipate the collective flow of the crowd and maintain appropriate social distancing while avoiding disruptive interactions.

Next, in Fig. 2e, the trajectory of HR-CQL collides at 4.2 seconds, as the robot trained with handcrafted rewards lacks sufficient obstacle avoidance capability. Fig. 2f and 2g show that HPR-CQL and RPR-CQL avoid direct collision but still approach pedestrians too closely at 4 seconds, with a tendency to pass through rather than detour. In Fig. 2h, our SPLC-CQL instead performs a clear rightward bypass, ensuring safer navigation and more stable trajectory evolution. These comparisons indicate that our preference evaluation criteria fully account for the intricacies of pedestrian dynamics, providing CQL with more stable and task-aligned learning signals, and thereby enhancing both the safety and efficiency of navigation with more socially compliant yielding behavior.

Furthermore, Fig. 2i shows that HR-TD3BC completes the task but slows down significantly in dense crowds, exhibiting inefficient low-speed detours. Similarly, Fig. 2j and 2k indicate that HPR-TD3BC and RPR-TD3BC are overly conservative, prolonging navigation time and sacrificing efficiency. By comparison, Fig. 2l demonstrates that our SPLC-TD3BC anticipates dense pedestrian regions and executes early avoidance, maintaining a safe distance from pedestrians during motion, thereby completing the navigation task more safely and in a shorter duration. The main reason is that our principled preference evaluation criteria better quantify social norms in crowd robot navigation and mitigate reward bias arising from the inherent unpredictability and often uncooperative dynamics of pedestrian motion, resulting in more socially compliant navigation.

D. Real-world Experiments

This work was further validated through real-world experiments in addition to simulations. The experimental platform comprised a laptop with an R9-7940HX processor and an RTX 4060 GPU, integrated with a TurtleBot4. An RPLIDAR-A1 LiDAR combined with a pedestrian leg detection algorithm was employed to identify pedestrians and estimate their relative positions and velocities. The robot's state was tracked using its built-in chassis odometry after mapping the environment. During the experiments, the collected state data were transmitted to the laptop, where the deployed SPLC-IQL algorithm generated the control actions, which were then executed by the robot in the physical environment.

As illustrated in Fig. 3, the robot successfully reached the target while avoiding collisions with five pedestrians. In Fig. 3b, it can be observed that the robot exhibited clear obstacle-avoidance behaviors when pedestrians approached nearby. Full real-world demonstrations of the proposed algorithm are provided in the submitted multimedia materials. These results confirm that the algorithm can be effectively transferred from simulation to a physical robot, enabling it to accomplish crowd robot navigation tasks.

V. CONCLUSIONS

In this study, we have proposed the SPLC algorithm to address the reward function design issue in DRL-based crowd navigation. Our SPLC introduces a social preference feedback mechanism that automatically generates preference data via preference evaluation criteria to model the reward function, alleviating reward biases in the crowd navigation. By integrating SPLC with several offline RL algorithms, we demonstrated through extensive experiments that the proposed SPLC algorithm achieves superior performance in terms of both average navigation success rate and navigation efficiency. Finally, experiments on the TurtleBot4 show that our SPLC algorithm successfully transfers from simulation to real-world robots. In future work, we will conduct more systematic real-world experiments to further examine sim-to-real transfer, while incorporating real-world crowd navigation datasets to improve the robustness and practical applicability of the proposed SPLC framework.

REFERENCES

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

[2] 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.

[3] 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.

[4] H. Kretzschmar, M. Spies, C. Sprunk, and W. Burgard, "Socially compliant mobile robot navigation via inverse reinforcement learning," *The International Journal of Robotics Research*, vol. 35, no. 11, pp. 1289–1307, 2016.

[5] P. Trautman, J. Ma, R. M. Murray, and A. Krause, "Robot navigation in dense human crowds: the case for cooperation," in *2013 IEEE international conference on robotics and automation*. IEEE, 2013, pp. 2153–2160.

[6] J. Choi, G. Lee, and C. Lee, "Reinforcement learning-based dynamic obstacle avoidance and integration of path planning," *Intelligent Service Robotics*, vol. 14, pp. 663–677, 2021.

[7] M. Everett, Y. F. Chen, and J. P. How, "Motion planning among dynamic, decision-making agents with deep reinforcement learning," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 3052–3059.

[8] 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.

[9] S. Liu, H. Xia, F. C. Pouria, K. Hong, N. Chakraborty, and K. R. Driggs-Campbell, "HEIGHT: Heterogeneous interaction graph transformer for robot navigation in crowded and constrained environments," *CoRR*, vol. abs/2411.12150, 2024.

[10] I. Kostrikov, A. Nair, and S. Levine, "Offline reinforcement learning with implicit q-learning," in *Deep RL Workshop NeurIPS 2021*, 2021. [Online]. Available: https://openreview.net/forum?id=EblVBDNalKu

[11] A. Kumar, A. Zhou, G. Tucker, and S. Levine, "Conservative Q-learning for offline reinforcement learning," *Advances in neural information processing systems*, vol. 33, pp. 1179–1191, 2020.

[12] S. Fujimoto and S. S. Gu, "A minimalist approach to offline reinforcement learning," *Advances in neural information processing systems*, vol. 34, pp. 20 132–20 145, 2021.

[13] B. R. Kiran, I. Sobh, V. Talpaert, P. Mannion, A. A. Al Sallab, S. Yogamani, and P. Pérez, "Deep reinforcement learning for autonomous driving: A survey," *IEEE transactions on intelligent transportation systems*, vol. 23, no. 6, pp. 4909–4926, 2021.

[14] J. Li, C. Tang, M. Tomizuka, and W. Zhan, "Hierarchical planning through goal-conditioned offline reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10 216–10 223, 2022.

[15] J. Wu, Y. Wang, H. Asama, Q. An, and A. Yamashita, "Risk-sensitive mobile robot navigation in crowded environment via offline reinforcement learning," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 7456–7462.

[16] K. Weerakoon, A. J. Sathyamoorthy, M. Elnoor, and D. Manocha, "VAPOR: Legged robot navigation in unstructured outdoor environments using offline reinforcement learning," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 10 344–10 350.

[17] S. Mitsch, K. Ghorbal, and A. Platzer, "On provably safe obstacle avoidance for autonomous robotic ground vehicles," in *Robotics: Science and Systems IX, Technische Universität Berlin, Berlin, Germany, June 24-June 28, 2013*, 2013.

[18] B. Ichter, J. Harrison, and M. Pavone, "Learning sampling distributions for robot motion planning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 7087–7094.

[19] A. Biswas, A. Wang, G. Silvera, A. Steinfeld, and H. Admoni, "Socnavbench: A grounded simulation testing framework for evaluating social navigation," *ACM Transactions on Human-Robot Interaction (THRI)*, vol. 11, no. 3, pp. 1–24, 2022.

[20] Z. Xie and P. Dames, "DRL-VO: Learning to navigate through crowded dynamic scenes using velocity obstacles," *IEEE Transactions on Robotics*, vol. 39, no. 4, pp. 2700–2719, 2023.

[21] S. Liu, P. Chang, W. Liang, N. Chakraborty, and K. Driggs-Campbell, "Decentralized structural-RNN for robot crowd navigation with deep reinforcement learning," in *2021 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2021, pp. 3517–3524.

[22] Y.-J. Mun, M. Itkina, S. Liu, and K. Driggs-Campbell, "Occlusion-aware crowd navigation using people as sensors," *arXiv preprint arXiv:2210.00552*, 2022.

[23] Z. Zhou, Z. Zeng, L. Lang, W. Yao, H. Lu, Z. Zheng, and Z. Zhou, "Navigating robots in dynamic environment with deep reinforcement learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 25 201–25 211, 2022.

[24] 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.

[25] W. Wang, R. Wang, L. Mao, and B.-C. Min, "NaviSTAR: Socially aware robot navigation with hybrid spatio-temporal graph transformer and preference learning," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 11 348–11 355.

[26] H. Jiang, N. Bhujel, Z. Lin, K.-W. Wan, J. Li, S. Jayavelu, and X. Jiang, "Learning relation in crowd using gated graph convolutional networks for DRL-based robot navigation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 6, pp. 5085–5095, 2023.

[27] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei, "Deep reinforcement learning from human preferences," *Advances in neural information processing systems*, vol. 30, 2017.

[28] Y. Abdelkareem, S. Shehata, and F. Karray, "Advances in preference-based reinforcement learning: A review," in *2022 IEEE international conference on systems, man, and cybernetics (SMC)*. IEEE, 2022, pp. 2527–2532.

[29] T. Kaufmann, P. Weng, V. Bengs, and E. Hüllermeier, "A survey of reinforcement learning from human feedback," *Transactions on Machine Learning Research*, 2024.

[30] H. Du, S. Teng, H. Chen, J. Ma, X. Wang, C. Gou, B. Li, S. Ma, Q. Miao, X. Na *et al.*, "Chat with chatGPT on intelligent vehicles: An IEEE TIV perspective," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 3, pp. 2020–2026, 2023.

[31] A. Pacchiano, A. Saha, and J. Lee, "Dueling rl: reinforcement learning with trajectory preferences," *arXiv preprint arXiv:2111.04850*, 2021.

[32] X. Wang, K. Lee, K. Hakhamaneshi, P. Abbeel, and M. Laskin, "Skill preferences: Learning to extract and execute robotic skills from human feedback," in *Conference on robot learning*. PMLR, 2022, pp. 1259–1268.

[33] Y. Cao, B. Ivanovic, C. Xiao, and M. Pavone, "Reinforcement learning with human feedback for realistic traffic simulation," in *2024 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2024, pp. 14 428–14 434.

[34] D. Shin, A. D. Dragan, and D. S. Brown, "Benchmarks and algorithms for offline preference-based reward learning," *arXiv preprint arXiv:2301.01392*, 2023.

[35] C. Kim, J. Park, J. Shin, H. Lee, P. Abbeel, and K. Lee, "Preference transformer: Modeling human preferences using transformers for RL," *arXiv preprint arXiv:2303.00957*, 2023.

[36] J. Li, B. Luo, X. Xu, and T. Huang, "Offline reward shaping with scaling human preference feedback for deep reinforcement learning," *Neural Networks*, vol. 181, p. 106848, 2025.

[37] 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.

[38] H. Fu, Q. Wang, and H. He, "Path-following navigation in crowds with deep reinforcement learning," *IEEE Internet of Things Journal*, vol. 11, no. 11, pp. 20 236–20 245, 2024.

[39] R. A. Bradley and M. E. Terry, "Rank analysis of incomplete block designs: I. the method of paired comparisons," *Biometrika*, vol. 39, no. 3/4, pp. 324–345, 1952. [Online]. Available: http://www.jstor.org/stable/2334029

[40] J. Heo, Y. J. Lee, J. Kim, M. G. Kwak, Y. J. Park, and S. B. Kim, "Mixing corrupted preferences for robust and feedback-efficient preference-based reinforcement learning," *Knowledge-Based Systems*, vol. 309, p. 112824, 2025.

[41] D. Tarasov, A. Nikulin, D. Akimov, V. Kurenkov, and S. Kolesnikov, "CORL: Research-oriented deep offline reinforcement learning library," *Advances in Neural Information Processing Systems*, vol. 36, pp. 30 997–31 020, 2023.